

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

RAFAEL DIVINO FERREIRA FEITOSA

**Classificação de cenas utilizando a
análise da aleatoriedade por
aproximação da complexidade de
Kolmogorov**

Goiânia
2020



UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO (TECA) PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a [Lei 9.610/98](#), o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo das Teses e Dissertações disponibilizado na BDTD/UFG é de responsabilidade exclusiva do autor. Ao encaminhar o produto final, o autor(a) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do material bibliográfico

Dissertação Tese

2. Nome completo do autor

Rafael Divino Ferreira Feitosa

3. Título do trabalho

Classificação de cenas utilizando a análise da aleatoriedade por aproximação da complexidade de Kolmogorov

4. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador)

Concorda com a liberação total do documento SIM NÃO¹

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante:

- a) consulta ao(à) autor(a) e ao(à) orientador(a);
- b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo da tese ou dissertação.

O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

Obs. Este termo deverá ser assinado no SEI pelo orientador e pelo autor.



Documento assinado eletronicamente por **Anderson Da Silva Soares, Professor do Magistério Superior**, em 02/06/2020, às 23:14, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **RAFAEL DIVINO FERREIRA FEITOSA, Discente**, em 03/06/2020, às 11:02, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1331347** e o código CRC **A0A21934**.

RAFAEL DIVINO FERREIRA FEITOSA

Classificação de cenas utilizando a análise da aleatoriedade por aproximação da complexidade de Kolmogorov

Tese apresentada ao Programa de Pós-Graduação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

Área de concentração: Ciência da Computação.

Orientador: Prof. Dr. Anderson da Silva Soares

Goiânia
2020

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Ferreira Feitosa, Rafael Divino

Classificação de cenas utilizando a análise da aleatoriedade por
aproximação da complexidade de Kolmogorov [manuscrito] / Rafael
Divino Ferreira Feitosa. - 2020.

CCIII, 203 f.: il.

Orientador: Prof. Dr. Anderson da Silva Soares.

Tese (Doutorado) - Universidade Federal de Goiás, Instituto de
Informática (INF), Programa de Pós-Graduação em Ciência da
Computação em rede (UFG/UFMS), Goiânia, 2020.

Bibliografia. Apêndice.

Inclui siglas, símbolos, tabelas, lista de figuras, lista de tabelas.

1. classificação de cenas. 2. compressão de dados. 3.
aleatoriedade. 4. complexidade de Kolmogorov. 5. complexidade
visual. I. da Silva Soares, Anderson, orient. II. Título.

CDU 004



UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

ATA DE DEFESA DE TESE

Ata Nº **03/2020** da sessão de Defesa de Tese de **Rafael Divino Ferreira Feitosa** que confere o título de Doutor em Ciência da Computação, na área de concentração em Ciência da Computação.

Aos treze dias do mês de maio de dois mil e vinte, a partir das dezenove horas e trinta minutos, via sistema de webconferência da RNP, realizou-se a sessão pública de Defesa de Tese intitulada “**Classificação de cenas utilizando a análise da aleatoriedade por aproximação da complexidade de Kolmogorov**”. Os trabalhos foram instalados pelo Orientador, Professor Doutor Anderson da Silva Soares (INF/UFG) com a participação dos demais membros da Banca Examinadora: Professor Doutor Alexandre Cláudio Botazzo Delbem (ICMC/USP), membro titular externo; Professor Doutor Gustavo Teodoro Laureano (INF/UFG), membro titular externo; Professor Doutor Fabrizzio Alphonsus Alves de Melo Nunes Soares (INF/UFG), membro titular interno; Professor Doutor Ronaldo Martins da Costa (INF/UFG), membro titular interno. A realização da banca ocorreu per meio de videoconferência, em atendimento à recomendação de suspensão das atividades presenciais na UFG dadas pelo Comitê UFG para o Gerenciamento da Crise COVID-19, bem como à recomendação de isolamento social da Organização Mundial de Saúde e do Ministério da Saúde para enfrentamento da emergência de saúde pública decorrente do novo coronavírus. Durante a arguição os membros da banca não fizeram sugestão de alteração do título do trabalho. A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Tese tendo sido o candidato **aprovado** pelos seus membros. Proclamados os resultados pelo Professor Doutor Anderson da Silva Soares, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora, aos treze dias do mês de maio de dois mil e vinte.

TÍTULO SUGERIDO PELA BANCA



Documento assinado eletronicamente por **Anderson Da Silva Soares, Professor do Magistério Superior**, em 13/05/2020, às 22:47, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Fabrizzio Alphonsus Alves De Melo Nunes Soares, Professor do Magistério Superior**, em 13/05/2020, às 22:49, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Gustavo Teodoro Laureano, Professor do Magistério Superior**, em 13/05/2020, às 22:49, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Ronaldo Martins Da Costa, Professor do Magistério Superior**, em 13/05/2020, às 22:49, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Alexandre Cláudio Botazzo Delbem, Usuário Externo**, em 13/05/2020, às 22:50, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **RAFAEL DIVINO FERREIRA FEITOSA, Discente**, em 13/05/2020, às 23:16, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1318678** e o código CRC **1B28D3F5**.

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador(a).

Rafael Divino Ferreira Feitosa

Doutor em Ciência da Computação pela Universidade Federal de Goiás (2020), especialista em Desenvolvimento de Sistemas para Web pela UniAnhanguera (2010) e graduado em Sistemas de Informação pela Universidade Estadual de Goiás (2008). Atualmente é professor do Departamento de Informática do Instituto Federal de Educação, Ciência e Tecnologia Goiano - Campus Ceres.

A Deus, causa primária de todas as coisas,
A Débora Nice, minha companheira *ao infinito e além...* 🐦,
A Fátima, meu sustento moral.

Agradecimentos

Primeiramente, a Deus, fonte de infinita sabedoria e misericórdia, que nos presenteia todos os dias com oportunidades de renovação e superação.

Agradeço ao projeto de P&D ANEEL-Copel Distribuição de número 2866-04842017 que ofereceu suporte financeiro a esta tese.

A minha amada esposa Débora Nice, eterna namorada, amiga e companheira, que tanto me apoia nos momentos difíceis e sempre celebra comigo as conquistas. Sem sua compreensão e renúncia tudo seria mais difícil. Com ela ao meu lado, tudo é mais fácil e leve.

À minha mãe Fátima, que sempre me enviou boas energias por meio de suas orações e me incentivou com palavras confortantes e de apoio.

Ao Professor Anderson, meu mentor e orientador, pela paciência, por partilhar seus conhecimentos e experiências e, sobretudo, pela confiança depositada em mim quando aceitou me orientar nessa gratificante jornada. Todos esses anos ao seu lado me inspiraram como docente e me motivaram a retribuir tudo o que recebi em forma de dedicação aos meus alunos, à ciência e à educação.

Aos Professores Gustavo Teodoro, Ronaldo Martins, Humberto Longo e Les Foulds pelos momentos que dedicaram em me auxiliar na pesquisa, sempre dispostos a colaborar.

Ao Instituto Federal de Educação, Ciência e Tecnologia Goiano que oportunizou a minha dedicação integral à pesquisa.

Enfim, agradeço a todos os meus professores e colegas que, de alguma forma, contribuíram tanto para a conclusão desse doutorado quanto para o meu crescimento como pessoa.

"Fazer ciência é comprimir uma grande quantidade de informações sobre um fenômeno (as observações) em uma descrição mínima (uma teoria)."

Gregory Chaitin (1974) [28]

Traduzido por Campani e Menezes (2009) [21]

Resumo

Ferreira Feitosa, Rafael Divino. **Classificação de cenas utilizando a análise da aleatoriedade por aproximação da complexidade de Kolmogorov**. Goiânia, 2020. 203p. Tese de Doutorado. Instituto de Informática, Universidade Federal de Goiás.

Em muitos problemas de reconhecimento de padrões, as características discriminantes são desconhecidas e/ou as fronteiras das classes não são bem definidas. Diversos trabalhos têm utilizado a compressão de dados para descoberta de conhecimento, sem a extração e seleção dessas características. O fundamento dessa abordagem é que dois objetos podem ser agrupados como semelhantes, se o conteúdo da informação de um deles explicar, de forma significativa, o conteúdo da informação do outro. Entretanto, as técnicas baseadas em compressão não são eficientes para imagens, pois desconsideram a semântica presente na correlação espacial dos dados bidimensionais. É proposto um classificador, que estima a complexidade visual de cenas, denominado *Reconhecimento de Padrões por Aleatoriedade* (RPA). O funcionamento do método é fundamentado em transformações de dados, que ampliam as características mais discriminantes e suprimem os detalhes. A principal contribuição do trabalho é o uso da aleatoriedade como medida de discriminação. A aproximação entre cenas e modelos treinados, baseada na distorção representacional, promove um processo de compressão com perda. Essa perda é associada a detalhes irrelevantes, quando a cena é reconstruída com a representação de sua própria classe, ou à degradação de informações, quando é reconstruída com representações díspares. Quanto mais informação preservada, maior a aleatoriedade da reconstrução. Do ponto de vista matemático, o funcionamento do método é explicado por duas principais medidas no plano U -dimensional de projeção das amostras: interseção e dispersão. Os resultados mostram acurácias de 0,6967, para um problema de 12 classes, e 0,9286 para 7 classes. Comparado com o k -NN e um *toolkit* de mineração de dados, o classificador proposto foi superior. O método é capaz de gerar modelos eficientes a partir de poucas amostras de treinamento. É invariante para reflexões verticais e horizontais e resistente a algumas transformações geométricas e processamentos de imagem.

Palavras-chave

classificação de cenas, compressão de dados, quantização vetorial, aleatoriedade, complexidade de Kolmogorov, entropia, complexidade visual

Abstract

Ferreira Feitosa, Rafael Divino. **Scene classification using the analysis of randomness by approximation of Kolmogorov's complexity**. Goiânia, 2020. 203p. PhD. Thesis. Instituto de Informática, Universidade Federal de Goiás.

In many pattern recognition problems, discriminant features are unknown and/or class boundaries are not well defined. Several studies have used data compression to discover knowledge, without features extraction and selection. The basic idea is two distinct objects can be grouped as similar, if the information content of one explains, in a significant way, the information content of the other. However, compression-based techniques are not efficient for images, as they disregard the semantics present in the spatial correlation of two-dimensional data. A classifier is proposed for estimates the visual complexity of scenes, namely *Pattern Recognition by Randomness* (PRR). The operation of the method is based on data transformations, which expand the most discriminating features and suppress details. The main contribution of the work is the use of randomness as a measure discrimination. The approximation between scenes and trained models, based on representational distortion, promotes a lossy compression process. This loss is associated with irrelevant details, when the scene is reconstructed with the representation of true class, or with the information degradation, when it is reconstructed with divergent representations. The more information preserved, the greater the randomness of the reconstruction. From the mathematical point of view, the method is explained by two main measures in the U -dimensional plane: intersection and dispersion. The results yielded accuracy of 0.6967, for a 12-class problem, and 0.9286 for 7 classes. Compared with k-NN and a data mining toolkit, the proposed classifier was superior. The method is capable of generating efficient models from few training samples. It is invariant for vertical and horizontal reflections and resistant to some geometric transformations and image processing.

Keywords

scene classification, data compression, vector quantization, randomness, Kolmogorov's complexity, entropy, visual complexity

Sumário

Listas de Siglas e Notação	14
Lista de Figuras	17
Lista de Tabelas	21
1 Introdução	23
1.1 Contexto	23
1.2 Problema	25
1.3 Hipóteses	27
1.4 Objetivos	28
1.5 Proposta da Solução	29
2 Reconhecimento de padrões e compressão de dados	31
2.1 Fundamentos de Reconhecimento de Padrões	31
2.2 Sistema Visual Atentivo Humano	34
2.3 Compressão de Dados	37
2.3.1 Redundância de Dados em Imagens	38
2.3.2 Conceitos de Teoria da Informação	40
2.3.3 Compressão e Reconstrução dos Dados	41
2.3.4 Métodos de Compressão de Imagens	43
2.3.5 Compressão por Quantização	45
2.3.6 Formatos de Imagens	50
2.4 Algoritmos de Classificação de Cenas	52
2.5 Medida para Avaliação de Sobreposição de Classes <i>R-value</i>	54
2.6 Complexidade de Kolmogorov	56
2.6.1 Bases da Complexidade de Kolmogorov	57
2.6.2 Complexidades $K(x)$ e $K(x y)$	59
2.6.3 Universalidade de K e Aleatoriedade	60
2.6.4 Aleatoriedade e Informação	61
2.7 Similaridade Baseada em Compressão de Dados	62
2.7.1 Distância da Informação	62
2.7.2 Distância de Compressão	63
2.7.3 Distância Normalizada da Informação	64
2.7.4 Distância Normalizada de Compressão	64

3	Medidas de similaridade de imagens baseadas em compressão	66
3.1	Compressão como Medida de Similaridade	66
3.2	Toolkits: CompLearn e DAMICORE	68
3.3	Medida Quantitativa de Agrupamentos G	69
3.4	Propriedades e Parâmetros da NCD	70
3.4.1	Algoritmos de Compressão	70
3.4.2	Formatos de Imagens	73
3.4.3	Transformações e Processamentos de Imagem	74
3.4.4	Linearização e Concatenação	76
3.5	Outras Medidas Baseadas em Compressão	78
3.5.1	Decomposição NID (α -NCD)	78
3.5.2	Distância Normalizada de Compressão Condicional (NCCD)	79
3.5.3	Método Campana-Keogh (CK-1)	80
3.5.4	Distância Normalizada de Compressão Modificada (MNCD)	81
3.5.5	Distância de Compressão Rápida (FCD)	82
3.5.6	Distância Normalizada de Compressão Condicional Modificada (NCCD')	83
3.5.7	Distância Baseada na Representação Esparsa	85
3.5.8	Compressão Normalizada Relativa (NRC)	86
3.6	Outras Aplicações de Medidas de Similaridade por Compressão	87
3.7	Contribuições e Lacunas de Pesquisa	89
4	Método de classificação de cenas utilizando compressão de dados	94
4.1	Preliminares	94
4.2	Base de Dados	95
4.3	Arquitetura do Método Proposto	96
4.4	Módulo de Treinamento	98
4.4.1	Prototipação	98
4.4.2	Quantização de Treinamento e Medidas de Aleatoriedade	100
	Entropia Algorítmica Aproximada por Compressão	102
	Entropia Média	104
4.4.3	Validação	106
4.5	Módulo de Classificação	109
4.5.1	Quantização de Teste e Decisão	109
4.5.2	Implementação e Código-fonte	110
5	Parametrização e desempenho do classificador de cenas	111
5.1	Experimentos	111
5.1.1	Recursos Computacionais	111
5.1.2	Parametrização do Método	112
	Particionamento dos Dados	112
	Convergência de Treinamento ϵ	112
	Tamanhos da Janela e do Dicionário	112
5.2	Resultados e Discussão	113
5.2.1	Particionamento dos Dados e Taxas de Convergência	113
5.2.2	Tamanho da Janela e do Dicionário	122
5.2.3	Avaliação de Desempenho	129
	Transformações e Processamentos de Imagens	140
	Comparativo com DAMICORE e k-NN	143

5.2.4	Escopo e Restrições do Método	146
6	Conclusões	156
6.1	Trabalhos futuros	159
	Referências Bibliográficas	161
A	Cenários dos Experimentos	175
B	Tabulação dos Dados	187

Listas de Siglas e Notação

Lista de Siglas

k -NN	k-Nearest Neighbors	NJ	Neighbor Joining
ANN	Artificial Neural Networks	NRC	Normalized Relative Compression
BMP	Bitmap Image File	PGM	Portable Gray Map
BoW	Bag-of-Words	PNG	Portable Network Graphics
CBIR	Content-Based Image Retrieval	PPM	Portable Pixel Map
DAMICORE	Data-Mining of Code Repositories	PPM	Prediction by Partial Matching
DCT	Discrete Cosine Transform	PRDC	Pattern Representation based on Data Compression
DPCM	Differential Pulse-Code Modulation	PSNR	Peak Signal-to-Noise Ratio
DWT	Discrete Wavelet Transform	RD	Representational Distortion
FCD	Fast Compression Distance	RLE	Run-Length Encoding
FCM	Finite-Context Model	SCPO	Self-Describing Context Based Pixel Ordering
GIF	Graphics Interchange Format	SIT	Structural Information Theory
JBIG	Joint Bi-level Image Experts Group	SNR	Signal-to-Noise Ratio
JPEG	Joint Photographic Experts Group	SSIM	Structural Similarity
K-SVD	K-Singular Value Decomposition	SVM	Support Vector Machines
LBG	Algoritmo Linde-Buzo-Gray	t-SNE	t-Distributed Stochastic Neighbor Embedding
LDA	Linear Discriminant Analysis	TIFF	Tagged Image File Format
LUT	LookUp Table	WinRAR	Windows Roshal Archive
LZMA	Lempel-Ziv-Markov Algorithm		
MNCD	Modified Normalized Compression Distance		
MSE	Mean Squared Error		
NCCD	Normalized Conditional Compression Distance		
NCCD'	Modified Normalized Conditional Compression Distance		
NCD	Normalized Compression Distance		
NID	Normalized Information Distance		

Notação

$(x \rho_x)$	Porção-residual de uma imagem no cálculo da α -NCD(x, y). Obtida pela diferença entre a imagem e sua porção-modelo ρ .
α -NCD(x, y)	Distância normalizada de compressão decomposta entre os objetos x e y
δ	Subconjunto de classes com seus respectivos parâmetros de treinamento, denominado, <i>cenário</i>
ε	Limite inferior da diferença entre as distorções das iterações para convergência do algoritmo LBG

η	Unidade de decréscimo da acurácia mínima (τ) na expansão da busca de cenários δ válidos	ω ϕ	Classe de objetos Conjunto de classes e seus respectivos parâmetros com acurácia mínima τ
Γ	Conjunto de tamanhos de janela		
$\hat{y}^{(\cdot)}$	Variável resposta predita para a amostra $\mathbf{x}^{(\cdot)}$	Ψ	Distorção média entre as partições de uma iteração do algoritmo LBG
Λ	Conjunto de tamanhos de <i>codebook</i>	ρ	Porção-modelo de uma imagem no cálculo da α -NCD(x, y), obtida quantizando a imagem
λ	Tamanho do conjunto de cores ou <i>codevectors</i> resultante de uma quantização	τ	Taxa de acurácia mínima para determinar se a discriminabilidade de um cenário δ é relevante para resolução do problema
\mathbb{R}	Conjunto dos números reais		
\mathcal{A}	Alfabeto de símbolos de uma mensagem	$ \cdot $	Cardinalidade de um conjunto ou comprimento de uma string
\mathcal{C}	Conjunto de pontos no espaço U -dimensional	\mathbf{c}	Cor ou vetor de treinamento em uma quantização
\mathcal{D}	Conjunto de amostras de teste	\mathbf{v}	Cor representativa ou <i>codevector</i> resultante de uma quantização
$\mathcal{D}(\omega_w)$	Conjunto de amostras de teste pertencentes à classe ω_w	\mathbf{x}^T	Vetor de características transposto da amostra \mathbf{x}
\mathcal{J}	Conjunto de janelas extraídas de uma imagem	$\mathbf{x}_w^{(i)}$	i -ésima amostra da classe ω_w
$\mathcal{N}(\mathbf{x}_w^{(i)})$	Conjunto de k -vizinhos mais próximos da amostra $\mathbf{x}_w^{(i)}$	CK-1(x, y)	Distância de Campana-Keogh entre os objetos x e y
$\mathcal{N}(\mathbf{x}_w^{(i)}, \omega_m)$	Subconjunto de amostras em $\mathcal{N}(\mathbf{x}_w^{(i)})$ que pertencem a uma classe diferente ω_m	E(x, y) E _C (x, y)	Distância da informação entre os objetos x e y Distância da compressão entre os objetos x e y
\mathcal{P}	Conjunto de partições induzidas em uma quantização	MNCD(x, y)	Distância normalizada de compressão modificada entre os objetos x e y
\mathcal{R}	Conjunto de cores ou <i>codevectors</i> resultante de uma quantização	NCCD'(x, y)	Distância normalizada de compressão condicional modificada entre os objetos x e y
\mathcal{T}	Conjunto de amostras de treinamento	NCCD(x, y)	Distância normalizada de compressão condicional entre os objetos x e y
\mathcal{U}	Conjunto universo de amostras de um <i>dataset</i>		
\mathcal{V}	Conjunto de entradas de um dicionário de compressão. Podem ser entradas de uma compressão adaptativa ou <i>codevectors</i> organizados em um <i>codebook</i>	NCD(x, y) NID(x, y)	Distância normalizada de compressão entre os objetos x e y Distância normalizada da informação entre os objetos x e y
\mathcal{X}	Mensagem sem compressão	NRC(x, y)	Distância de compressão normalizada relativa entre os objetos x e y
\mathcal{X}_c	Representação comprimida de um mensagem \mathcal{X}	θ	Número limite de vizinhos pertencentes à outra classe para que uma amostra não seja considerado em uma área de interseção
\mathcal{Y}	Resultado da descompressão de uma mensagem		

ξ	String vazia	L	Tamanho do conjunto de tamanhos de <i>codebook</i>
b	Descrição binária de uma string		
b'	Descrição binária de uma string comprimida	M	Largura de uma imagem ou janela de uma imagem
C	Compressor do mundo real	M'	Quantidade horizontal de janelas de largura M
$C(\cdot)$	Tamanho da compressão individual de um objeto	N	Altura de uma imagem ou janela de uma imagem
$C(x y)$	Tamanho da compressão do objeto x condicionada ao objeto y	N'	Quantidade vertical de janelas de altura N
$C(x y)$	Tamanho da compressão do objeto x condicionada exclusivamente ao objeto y	n_{bits}	Número mínimo necessário de bits para representar os símbolos de um dicionário
$C(xy)$	Tamanho da compressão da concatenação dos objetos x e y	$o(\mathbf{X}_w^{(d)})$	Valor da medida de complexidade da w -ésima versão quantizada da imagem $\mathbf{X}^{(\cdot)}$
D	Quantidade de amostras de teste		
$d_S(x, y)$	Distância baseada na representação esparsa dos objetos x e y	$P(a)$	Probabilidade de ocorrência do símbolo a
FCD	Distância de compressão rápida entre os objetos x e y	$q(\cdot)$	Função de quantização das cores ou vetores de treinamento para as cores quantizadas ou <i>codevectors</i> , respectivamente, no processo de quantização
G	Medida de capacidade de generalização de árvores filogenéticas		
$g(\cdot)$	Função de mapeamento das cores ou vetores de treinamento para os índices das cores quantizadas ou <i>codevectors</i> , respectivamente, no processo de quantização	$R(\mathcal{U})$	Grau de interseção total entre as classes de um <i>dataset</i>
H	Entropia de Shannon de uma fonte	$R(\omega_w)$	Grau de interseção entre a classe ω_w e as demais classes do <i>dataset</i>
h	Tupla composta pelo rótulo de uma classe e seus respectivos parâmetros de treinamento	$R(\omega_w, \omega_m)$	Grau de interseção entre as classes ω_w e ω_m
$I(\cdot)$	Quantidade de informação individual de um objeto	R_C	Redundância relativa dos dados comprimidos
$I(y : x)$	Informação algorítmica de x contida em y	$S(\cdot, \mathcal{V})$	Complexidade esparsa de um objeto quando este é comprimido com o seu próprio dicionário \mathcal{V} ou um dicionário \mathcal{V} extraído de outro objeto
J	Tamanho do conjunto de tamanhos de janela	T	Quantidade de amostras de treinamento
K	Complexidade de Kolmogorov		
k	Número de vizinhos mais próximos	T_C	Taxa de compressão
$K(x)$	Complexidade individual de Kolmogorov do objeto x	U	Dimensionalidade do espaço de características
$K(x y)$	Complexidade condicional de Kolmogorov do objeto x em relação ao objeto y	W	Quantidade de classes
		$y^{(\cdot)}$	Variável resposta da amostra $\mathbf{x}^{(\cdot)}$

Lista de Figuras

1.1	Exemplo de imagens com padrões visuais (a) [6] e aleatória (b).	25
2.1	Modelo típico de um método de reconhecimento estatístico de padrões [73].	33
2.2	Processo de codificação de uma string pelo algoritmo de Huffman.	44
2.3	Quantização de amostras de cores pelo particionamento do espaço RGB sob a visão dos eixos verde (G) e azul (B) [104].	47
2.4	Diagramas de Voronoi criados pelas três primeiras iterações do algoritmo LBG em um espaço bidimensional [45]. Os pontos azuis são os vetores de treinamento e os pontos pretos são os centroides de cada partição, os <i>codevectors</i> .	50
2.5	Organização dos níveis de descritores de uma imagem [103].	53
2.6	Área de interseção entre 2 conjuntos [119]	55
2.7	Bases da complexidade de Kolmogorov [21]	57
3.1	Linhas de investigação das técnicas de reconhecimento de padrões em imagens por compressão de dados.	66
3.2	Contribuições e problematizações sobre similaridade de imagens utilizado compressão de dados.	90
4.1	Exemplos de cenas utilizadas nos experimentos das classes, da esquerda para a direita e de cima para baixo: avião, praia, construção, carro, campo, fogos de artifício, flor, corrida de moto, moto, montanha, pôr do sol e árvore.	96
4.2	Principais etapas do modelo de treinamento e classificação proposto.	97
4.3	Etapa de prototipação, base do módulo de treinamento.	99
4.4	Exemplo de imagem, após a conversão para escala de cinza e quantização de cor, preparada para fornecer amostras de treinamento para a classe praia com janelamento 15x15.	100
4.5	Etapas de quantização e cálculo da medida de classificação das imagens de treinamento.	101
4.6	Exemplo de uma imagem original da classe praia (a) após as quantizações com janelas de tamanho 5x5 e dicionário com 64 símbolos para a classe praia (b), montanha (c) e árvore (d).	102
4.7	Imagem quantizada com os codevectors mais próximos de cada janela da imagem original.	103
4.8	Exemplo das aproximações realizadas nas quantizações convergentes e divergentes visualizadas em um espaço bidimensional.	105
4.9	Imagens formadas por 4 níveis de intensidade, divididos em proporções semelhantes, com os pixels agrupados em sequência (a) e distribuídos aleatoriamente (b).	106

4.10	Etapa de validação, integrando as subetapas iterativas de decisão e otimização, para determinar os melhores parâmetros de cada classe.	107
4.11	Número de cenários avaliados com otimização e sem otimização para o contexto de 12 classes.	109
4.12	Modelo de classificação das imagens de teste descrito pelas etapas de quantização e decisão.	110
5.1	Tempo de prototipação das amostras de treinamento (em minutos) para as partições A (a), B (a), C (a) de acordo com as taxas de convergência ϵ .	113
5.2	Distorção média entre os vetores de treinamento em os codevectors para dicionários com 64, 128, 256 e 512 símbolos para as partições A (a), B (b) e C (c).	115
5.3	Variação média da diferença das medidas de qualidade MSE (a), SSIM (b), PSNR (c) e VIF (d) entre as imagens quantizadas para as classes verdadeiras e para as classes falsas.	117
5.4	Exemplo de uma cena da classe praia quantizada com codebook convergente, janela 5x5 e 512 símbolos, da partição A para $\epsilon = 0,1$ ((a), $\epsilon = 0,001$ ((b) e $\epsilon = 0,00001$ ((c)).	118
5.5	Detalhamento das diferenças das medidas de qualidade entre as imagens de treinamento quantizadas com suas respectivas classes convergentes e as imagens quantizadas com classes divergentes com $\epsilon = 0,001$.	119
5.6	Distorções médias das classes de acordo com os parâmetros de treinamento da partição A.	120
5.7	Distorções médias das classes de acordo com os parâmetros de treinamento da partição B.	121
5.8	Distorções médias das classes de acordo com os parâmetros de treinamento da partição C.	121
5.9	Frequência dos parâmetros de tamanhos da janela e do dicionário de cada classe, obtidos a partir dos melhores cenários na partição A e medidas de compressão.	122
5.10	Frequência dos parâmetros de tamanhos da janela e do dicionário de cada classe, obtidos a partir dos melhores cenários na partição A e medida de entropia.	124
5.11	Dispersões dos vetores de treinamento de cada classe, proporcionais à classe de menor distorção fogos de artifício.	127
5.12	Possíveis casos de distorção e dispersão de uma distribuição hipotética de amostras de treinamento no espaço bidimensional.	129
5.13	Maiores taxas de acurácia médias obtidas nos conjuntos de teste para cenários de 6 a 12 classes (detalhes nas Tabelas 5.7 e 5.8).	133
5.14	Delineamento experimental utilizado como critério para discussão da acurácia do RPA.	134
5.15	Matriz confusão do melhor resultado (itens 1.1, 1.1.1 e 1.1.3) obtido pelas medidas de compressão em cenários com 12 classes - incluindo variações de partição de treinamento, taxa de convergência ϵ e compressor - treinado e testado na partição A com $\epsilon = 0,00001$ e LZW+Huffman, conforme Tabela B.14 (ID: 106, acurácia: 0,6350, precision: 0,6487, recall: 0,6350, F1 score: 0,6337).	134

- 5.16 Taxas de acurácia da compressão LZW+Huffman comparada entre os treinamentos obtidos a partir das partições A, B e C e $\epsilon = \{0,1; 0,01; 0,001; 0,0001 \text{ e } 0,00001\}$ classificando o conjunto de teste da partição A. 135
- 5.17 Matriz confusão do pior resultado (item 1.1.2) obtido no conjunto de teste A pela compressão LZW+Huffman em cenários com 12 classes, variando as partições de treinamento e as taxas de convergência ϵ , treinado na partição C com $\epsilon = 0,1$ conforme Figura 5.16 (Acurácia = 0,5050). 135
- 5.18 Matriz confusão do pior resultado (item 1.1.4) obtido, variando os compressores, pelo melhor cenário de 12 classes, treinado e testado na partição A com $\epsilon = 0,00001$, quando utilizada a compressão *bzip2* (Acurácia = 0,17). 136
- 5.19 Matriz confusão do pior resultado (item 1.2) obtido pelas medidas de compressão em cenários com 12 classes, treinado e testado na partição C com $\epsilon = 0,001$ e ZIP, conforme Tabela B.15 (ID: 221, acurácia: 0,2931, precision: 0,3521, recall: 0,2931, F1 score: 0,2732). 137
- 5.20 Matriz confusão do melhor resultado (item 2.1) obtido pelas medidas de compressão em cenários com 7 classes, treinado e testado na partição A com $\epsilon = 0,01$ e *bzip2*, conforme Tabela B.14 (ID: 24, acurácia: 0,9286, precision: 0,9369, recall: 0,9286, F1 score: 0,9294). 137
- 5.21 Matriz confusão do pior resultado (item 2.2) obtido pelas medidas de compressão em cenários com 7 classes, treinado e testado na partição A com $\epsilon = 0,00001$ e ZIP, conforme Tabela B.14 (ID: 108, acurácia: 0,7200, precision: 0,7589, recall: 0,7200, F1 score: 0,7253). 137
- 5.22 Matriz confusão do melhor resultado (itens 1.1 e 1.1.1) obtido pela medida de entropia em cenários com 12 classes - incluindo variações de partição de treinamento e taxa de convergência ϵ - treinado e testado na partição A com $\epsilon = 0,00001$, conforme Tabela B.16 (ID: 93, acurácia: 0,6967, precision: 0,7260, recall: 0,6967, F1 score: 0,6911). 138
- 5.23 Matriz confusão do pior resultado (item 1.1.2) obtido no conjunto de teste A pela medida de entropia em cenários com 12 classes, variando as partições de treinamento e as taxas de convergência ϵ , treinado na partição C com $\epsilon = 0,1$ conforme Figura 5.24 (Acurácia = 0,52). 138
- 5.24 Taxa de acurácia da entropia comparada entre os treinamentos obtidos a partir das partições A, B e C e $\epsilon = \{0,1; 0,01; 0,001; 0,0001 \text{ e } 0,00001\}$ classificando o conjunto de teste da partição A. 139
- 5.25 Matriz confusão do pior resultado (item 1.2) obtido pela medida de entropia em cenários com 12 classes, treinado e testado na partição C com $\epsilon = 0,00001$, conforme Tabela B.16 (ID: 242, acurácia: 0,4049, precision: 0,5028, recall: 0,4049, F1 score: 0,3852). 139
- 5.26 Matriz confusão do melhor resultado (item 2.1) obtido pela medida de entropia em cenários com 7 classes, treinado e testado na partição A com $\epsilon = 0,1$, conforme Tabela B.16 (ID: 4, acurácia: 0,9029, precision: 0,9040, recall: 0,9029, F1 score: 0,9030). 140
- 5.27 Matriz confusão do pior resultado (item 2.2) obtido pela medida de entropia em cenários com 7 classes, treinado e testado na partição A com $\epsilon = 0,001$, conforme Tabela B.16 (ID: 59, acurácia: 0,8029, precision: 0,8373, recall: 0,8029, F1 score: 0,7984). 140

5.28	Acurácia do conjunto de teste sob transformações geométricas e processamentos de imagem utilizando as medidas de aleatoriedade de compressão LZW+Huffman e entropia	142
5.29	Taxas de acurácia obtidas pelo classificador k-NN aplicado às imagens da partição A em diferentes escalas e valores de vizinhança k.	144
5.30	Taxas de acurácias obtidas pelo classificador k-NN aplicado às imagens da partição A divididas em janelas de diferentes tamanhos e valores de vizinhança k.	145
5.31	Imagens do conjunto de testes utilizadas para exemplificação e discussão do funcionamento do RPA	151
5.32	Visualização bidimensional com t-SNE para o caso de classificação correta com diferença máxima (0,209607 para $k=3$ e $\theta=1$) entre a imagem de teste e suas classes verdadeira (a) e falsa (b) utilizando compressão LZW+Huffman de acordo com a Tabela B.17.	152
5.33	Visualização bidimensional com t-SNE para o caso de classificação correta com diferença mínima (0,000090 para $k=15$ e $\theta=7$) entre a imagem de teste e suas classes verdadeira (a) e falsa (b) utilizando compressão LZW+Huffman de acordo com a Tabela B.17.	152
5.34	Visualização bidimensional com t-SNE para o caso de classificação incorreta com diferença máxima (0,498963 para $k=7$ e $\theta=3$) entre a imagem de teste e suas classes verdadeira (a) e falsa (b) utilizando compressão LZW+Huffman de acordo com a Tabela B.17.	153
5.35	Visualização bidimensional com t-SNE para o caso de classificação incorreta com diferença mínima (0,002858 para $k=13$ e $\theta=6$) entre a imagem de teste e suas classes verdadeira (a) e falsa (b) utilizando compressão LZW+Huffman de acordo com a Tabela B.17.	153
5.36	Visualização bidimensional com t-SNE para o caso de classificação correta com diferença máxima (0,293234 para $k=3$ e $\theta=1$) entre a imagem de teste e suas classes verdadeira (a) e falsa (b) utilizando entropia de acordo com a Tabela B.18.	154
5.37	Visualização bidimensional com t-SNE para o caso de classificação correta com diferença mínima (0,002378 para $k=19$ e $\theta=9$) entre a imagem de teste e suas classes verdadeira (a) e falsa (b) utilizando entropia de acordo com a Tabela B.18.	154
5.38	Visualização bidimensional com t-SNE para o caso de classificação incorreta com diferença máxima (0,633303 para $k=3$ e $\theta=1$) entre a imagem de teste e suas classes verdadeira (a) e falsa (b) utilizando entropia de acordo com a Tabela B.18.	155
5.39	Visualização bidimensional com t-SNE para o caso de classificação incorreta com diferença mínima (0,001670 para $k=3$ e $\theta=1$) entre a imagem de teste e suas classes verdadeira (a) e falsa (b) utilizando entropia de acordo com a Tabela B.18.	155

Lista de Tabelas

2.1	Comparativo de métodos de compressão utilizados nos principais formatos de imagens.	51
5.1	Percentuais de interseção entre os holdouts dos conjuntos de treinamento, considerando valores máximos, médios e mínimos.	114
5.2	Percentuais de interseção entre os holdouts dos conjuntos de teste, considerando valores máximos, médios e mínimos.	114
5.3	Ocorrências dos parâmetros observados nos melhores cenários, obtidos na validação do treinamento da partição A e medidas de compressão.	123
5.4	Ocorrências dos parâmetros observados nos melhores cenários, obtidos na validação do treinamento da partição A e medida de entropia.	125
5.5	Ordenação crescente das medidas de distorção e dispersão das classes e suas relações com as maiores ocorrências de parâmetros de janela e dicionário, de acordo com as medidas de compressão e entropia.	128
5.6	Limites máximos e mínimos e valores médios de desvio-padrão observados entre as reamostragens dos dados de validação e teste para as métricas de avaliação de desempenho.	131
5.7	Detalhes das condições de testes das maiores acurácias obtidas em cenários de 6 a 12 classes utilizando compressão.	132
5.8	Detalhes das condições de testes das maiores taxas de acurácia obtidas em cenários de 7 a 12 classes utilizando entropia.	133
5.9	Valores da medida de generalização G das clusterizações realizadas pelo DAMICORE para os conjuntos de teste e treinamento da partição A utilizando diferentes compressores.	143
5.10	R-value $R(\mathcal{U})$ do conjunto de treinamento da partição A com 512 codevectors por classe.	149
5.11	R-values $R(\omega_w, \omega_m)$ e $R(\omega_w)$ do conjunto de treinamento da partição A a partir de janelas 30x30 com 512 codevectors por classe e parâmetros $k = 21$ e $\theta = 10$	150
A.1	Cenários dos experimentos	186
B.1	Distorções médias da prototipação da partição A.	188
B.2	Distorções médias da prototipação da partição B.	189
B.3	Distorções médias da prototipação da partição C.	190
B.4	Distâncias médias absolutas e relativas dos centroides dos codebooks entre as partições de treinamento com $\varepsilon = 0, 1$.	191
B.5	Distâncias médias absolutas e relativas dos centroides dos codebooks entre as partições de treinamento com $\varepsilon = 0, 01$.	191

B.6	Distâncias médias absolutas e relativas dos centroides dos codebooks entre as partições de treinamento com $\epsilon = 0,001$.	192
B.7	Distâncias médias absolutas e relativas dos centroides dos codebooks entre as partições de treinamento com $\epsilon = 0,0001$.	192
B.8	Distâncias médias absolutas e relativas dos centroides dos codebooks entre as partições de treinamento com $\epsilon = 0,00001$.	193
B.9	Medidas objetivas utilizadas para avaliar a qualidade média das imagens quantizadas com os dicionários convergentes e divergentes com $\epsilon = 0,1$.	194
B.10	Medidas objetivas utilizadas para avaliar a qualidade média das imagens quantizadas com os dicionários convergentes e divergentes com $\epsilon = 0,01$.	195
B.11	Medidas objetivas utilizadas para avaliar a qualidade média das imagens quantizadas com os dicionários convergentes e divergentes com $\epsilon = 0,001$.	196
B.12	Medidas objetivas utilizadas para avaliar a qualidade média das imagens quantizadas com os dicionários convergentes e divergentes com $\epsilon = 0,0001$.	197
B.13	Medidas objetivas utilizadas para avaliar a qualidade média das imagens quantizadas com os dicionários convergentes e divergentes com $\epsilon = 0,00001$.	198
B.14	Desempenho do conjunto de testes da partição A utilizando compressores.	199
B.15	Desempenho do conjunto de testes das partições B e C utilizando compressores.	200
B.16	Desempenho do conjunto de testes utilizando entropia.	201
B.17	R-values entre as imagens de teste, com as maiores e menores diferenças observadas, e as classes do melhor cenário utilizando compressão LZW+Huffman	202
B.18	R-values entre as imagens de teste, com as maiores e menores diferenças observadas, e as classes do melhor cenário utilizando entropia	203

Introdução

Neste capítulo são fornecidos os elementos fundamentais que nortearam a proposta e desenvolvimento da pesquisa. Os campos de reconhecimento de padrões e compressão de dados são contextualizados e discute-se de forma sucinta como a compressão pode ser utilizada como ferramenta de reconhecimento de padrões. São realizadas considerações acerca das limitações que permeiam os métodos baseados na compressão, levantando hipóteses e definindo objetivos para contribuir com o avanço das técnicas. Por fim, propõe-se uma arquitetura para um novo método de classificação.

1.1 Contexto

Reconhecimento de padrões é definido por [154] como a ciência do estudo da organização dos dados e pode ser realizada a partir de objetos representados em diversos tipos de dados como textos, áudios, sinais, vídeos, entre outros. Dentre esses tipos de dados, destacam-se as imagens com aplicação na área de Visão Computacional [74]. O reconhecimento de padrões em imagens é uma tarefa relativamente difícil em razão da complexidade, volume e alta variabilidade das informações analisadas e presença de ruídos nas fontes de dados [134]. Assim, a pesquisa e o desenvolvimento de novas técnicas e abordagens, capazes de lidarem com análise de imagens em problemas com diversidade de informações, são temas de relevância.

No contexto de processamento de imagens para reconhecimento de padrões, [63] definem uma imagem como uma função bidimensional, denominada $f(x,y)$, onde x e y são pontos representados por coordenadas espaciais e f determina a intensidade luminosa ou a cor nesse ponto. Para uma imagem ser digital, os pontos das coordenadas (x,y) , conhecidos como pixels, e o valor da intensidade devem ser finitos e discretos. Em [145], os autores trabalham com uma classe específica de imagens, denominada *cenas*, obtidas utilizando dispositivos de captura que operam no espectro visual. De acordo com os autores, essa classificação diferencia as cenas das imagens de textos, desenhos gerados por computador, ilustrações, pinturas ou quaisquer outras não provenientes de estímulos visuais como radar, sonar, raios-x, ultrassom e outros. A classificação de cenas são objeto do presente trabalho e formam um subconjunto de todas as imagens digitais. O trabalho

discutirá essa classe de imagens utilizando apenas o termo *cen*as. O termo *imagens* será utilizado para se referir a um arquivo ou nas discussões sobre trabalhos correlatos e métodos genéricos de processamento desse tipo de dados. O termo *cen*as será utilizado quando se referir à extração de semântica ou análise do conteúdo visual da imagem.

Segundo [134], as técnicas de reconhecimento de padrões em imagens podem ser divididas, principalmente, mas não exclusivamente, em três abordagens: estatística, estrutural e neural. Em [73] os autores também consideram a abordagem de casamento de modelos (*template matching*). De acordo com [134], na abordagem estatística são utilizadas fronteiras de decisão baseadas na distribuição estatística das características. A abordagem estrutural entende que a informação significativa é descrita pelas características, bem como suas inter-relações, e os objetos são reconhecidos de acordo com a similaridade da sua descrição ou representação estrutural. Na abordagem neural as fronteiras de decisão são definidas com base na minimização do erro de classificação. De acordo com [73], o casamento de modelos é o tipo mais simples de abordagem para reconhecimento de padrões e é realizado determinando a similaridade entre duas entidades de mesmo tipo. Entretanto, todas as abordagens de reconhecimento de padrões não são necessariamente independentes e, frequentemente, um mesmo algoritmo pode ser implementado sob diferentes interpretações [73]. Nessa perspectiva, acrescenta-se uma abordagem mais recente que introduz uma distância de compressão para reconhecer objetos similares, por meio da aproximação de suas informações mútuas [88].

Segundo [96], a compressão de dados é utilizada para reduzir o tamanho de dados para processamento, transmissão e armazenamento. O princípio da compressão de imagens está relacionado com a alta probabilidade de, ao selecionar um pixel arbitrário em uma imagem, os pixels vizinhos serem da mesma cor ou muito similares. Portanto, a compressão de imagens explora, principalmente, a redundância espacial da informação bidimensional, considerando que pixels vizinhos são altamente correlacionados em imagens não-aleatórias [139]. A Figura 1.1 apresenta um comparativo entre 2 imagens: imagem com padrões visuais, cujos pixels são altamente correlacionados espacialmente, (Figura 1.1(a)) e imagem com pixels distribuídos espacialmente de forma aleatória (Figura 1.1(b)). As redundâncias de dados presentes nas imagens podem ser divididas em três categorias: redundâncias de codificação, interpixel e psicovisual. Os métodos atualmente utilizados para compressão de imagens, normalmente, combinam estratégias que exploram mais de uma categoria de redundância e podem ser classificados como: compressão sem perda ou compressão com perda. Os métodos de compressão sem perda exploram, principalmente, as redundâncias de codificação e interpixel. Os métodos de compressão com perda exploram, mais ativamente, a redundância psicovisual (conjuntamente com a de codificação e interpixel) [129].

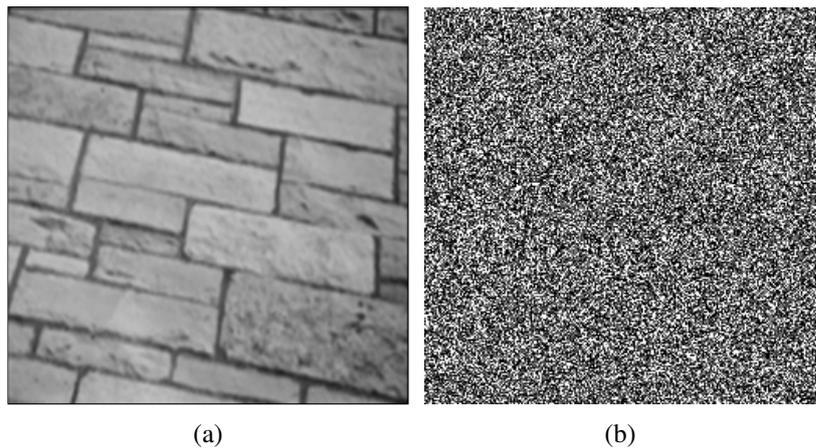


Figura 1.1: Exemplo de imagens com padrões visuais (a) [6] e aleatória (b).

1.2 Problema

A utilização da compressão de dados no contexto de reconhecimento de padrões foi proposta originalmente por [11]. A técnica é baseada na extração de semântica por meio da entropia dos objetos, determinada pela Teoria da Complexidade de Kolmogorov, denotada por K . Segundo essa teoria, a entropia algorítmica de uma cadeia de caracteres é o comprimento em bits do menor programa capaz de produzir essa cadeia [80]. A aleatoriedade intrínseca de um objeto¹ x é denotada por $K(x)$ ² e a complexidade condicional de dois objetos quaisquer x e y é denotada por $K(x|y)$ ³. Kolmogorov estabelece um limite inferior teórico de complexidade, portanto incomputável [25], para essas descrições algorítmicas individual e condicional. Portanto, em [10] os autores utilizaram a compressão de dados como aproximação para calcular um limite superior dessa complexidade algorítmica. A ideia básica é que dois objetos distintos podem ser agrupados como semelhantes se o conteúdo da informação de um deles explicar, de forma significativa, o conteúdo da informação do outro. Nesse caso, o tamanho resultante da compressão entre dois objetos mais similares tende a ser menor em relação aos outros, desde que normalizada.

Diversos trabalhos foram propostos utilizando medidas teóricas de similaridade baseadas na Teoria de Kolmogorov. A primeira foi a Distância da Informação (E) [11]. Posteriormente, foi proposta uma versão normalizada para tratar arquivos de tamanhos diferentes, a Distância Normalizada da Informação (NID - *Normalized Information Dis-*

¹Entidades compostas de informações digitais que podem ser expressas em símbolos próprios de sua representação. No contexto do presente trabalho são arquivos de imagens digitais. Entretanto, em outros contextos, podem ser outros formatos de arquivos ou entidades que podem ser processadas algorítmicamente.

²A complexidade $K(x)$ é o tamanho em bits do menor programa capaz de produzir x .

³A complexidade $K(x|y)$ é o tamanho em bits do menor programa capaz de produzir x dado que y é conhecido.

tance) [88]. A partir dessas formulações foram desenvolvidas métricas de similaridade baseadas em compressão, sendo a Distância Normalizada de Compressão (NCD - *Normalized Compression Distance*) [34], a mais amplamente utilizada, e outras [19, 116, 132]. Esses trabalhos utilizam a abordagem de aprendizagem não-supervisionada para validação de novas métricas ou de aplicações. O processo de montagem e definição dos *clusters*⁴ é realizado por meio de heurísticas, a partir da combinação de pares de objetos. Grande parte dos autores utiliza compressores comerciais, de propósito geral e sem perda na reconstrução dos dados originais. Nesse cenário, os resultados reportados na literatura são competitivos para dados unidimensionais, porém, apresentam desempenho limitado para dados bidimensionais [131].

Em [71], os autores afirmam que os compressores comerciais não são ideais para reconhecimento de padrões utilizando compressão. Eles promovem uma aproximação ineficiente das informações mútuas pois não distinguem tipos de dados e não se preocupam como estes estão estruturados. No caso de aplicação em problemas de agrupamento de imagens, ressalta-se que estas abordagens desprezam a correlação espacial dos dados - princípio fundamental da compressão de imagens [139]. Além disso, os compressores utilizados são sensíveis a ruídos pois não aceitam variações mínimas entre dados similares no processo de compressão sem perda [71]. Ainda como desvantagens dos métodos baseados nas derivações da NID, são elencadas:

- Desprezo à semântica: Apesar de ser uma das vantagens apontadas pelos autores, ao desenvolver um método livre de parâmetros, informações importantes de estruturas de dados bidimensionais podem ser perdidas no processo de compressão dos arquivos byte-a-byte. Deve ser lembrado que esse problema não ocorre em estruturas unidimensionais, visto que a semântica é preservada pela direção da leitura e compressão dos dados;
- *Lazy learning*: Seu aprendizado é dito preguiçoso pois a generalização do problema não é definida até que um objeto desconhecido seja submetido à clusterização [58]. Como não possui a etapa de treinamento, todo o processamento é concentrado na predição, tornando-a computacionalmente dispendiosa. É necessária a preservação de todo o conjunto de dados e reconstrução da matriz de distâncias produzida pela NCD a cada novo objeto submetido ao agrupamento. Isso significa aumento progressivo do volume de dados armazenados e do tempo de processamento. No caso das imagens o armazenamento demandado é maior e o processamento computacionalmente mais oneroso;
- Rotulação dos clusters: Como o processo de agrupamento de dados é um método não-supervisionado, não existe nenhum tipo de informação preexistente

⁴Grupo de objetos similares [135].

que promova uma classificação ótima para o conjunto de dados. Este problema é NP-difícil e necessita de uma heurística para encontrar uma solução aproximada [91]. Outro fator limitante é o fato das métricas utilizadas para formação dos agrupamentos não evidenciarem as características de cada cluster [43]. Para resolver esse problema, métodos adicionais, como a inclusão de uma etapa de aprendizado supervisionado [94], são necessários para rotulação automática dos clusters. Mesmo nesse caso, a classificação é realizada a partir do resultado de uma descoberta de conhecimento automática. Assim, os acertos resultantes podem ser comprometidos ao se utilizar métricas inadequadas ou ocorrer falhas nas heurísticas de clusterização.

Os métodos baseados na NCD contribuíram de forma significativa para a área de mineração de dados, se apresentando como uma solução livre de parâmetros e de propósito geral [34]. No trabalho de [140], por exemplo, é demonstrado que a compressão é capaz não somente de tornar a representação de um objeto mais eficiente, mas também encontrar padrões para agrupamento de dados. Entretanto, essa técnica se mostrou ineficiente para determinados tipos de problemas, motivando o desenvolvimento de métodos específicos para cada tipo de dados. Explorando a aplicabilidade desse método, pode-se investigar em quais tipos de problemas de classificação de imagens seria possível obter melhores resultados.

De acordo com [4], um problema de similaridade de imagens pode ser entendido por meio de duas etapas fundamentais:

1. Encontrar um conjunto de características que descreva de forma eficiente a assinatura das imagens;
2. Aplicar uma métrica adequada para calcular as distâncias entre as imagens.

Entretanto, em grande parte dos problemas de classificação de imagens as características discriminantes são desconhecidas e/ou as fronteiras das classes não são bem definidas. Nesses casos, quando há necessidade de um processo de tomada de decisão baseada em aprendizagem supervisionada, a aplicação de técnicas de reconhecimento de padrões tradicionais é difícil de ser implementada. Para [130], embora as escolhas mais populares para avaliação de similaridade de imagens ainda envolvam a extração de características, a maior dificuldade está na escolha de quais características são discriminantes para o problema.

1.3 Hipóteses

As contribuições do presente trabalho são complementares aos métodos antecessores de agrupamentos de dados baseados em compressão. Esses métodos são limitados

na extração de semântica de dados bidimensionais e os métodos tradicionais de reconhecimento de padrões são dependentes de uma etapa de seleção de características para atingir o máximo desempenho. Para isso, estabelece-se a hipótese norteadora: *dado o nível de abstração de uma cena, a quantidade de informação contida nela, obtida por uma medida de aleatoriedade intrínseca, é capaz de discriminá-la de outras imagens*. Ao encontrar esse nível de abstração, onde as características de uma cena emergem, permitindo o reconhecimento do seu contexto, a complexidade pode ser medida por métodos de compressão. Assim, espera-se contribuir com o estado da arte elencando e desenvolvendo, no presente trabalho, as seguintes hipóteses:

1. A compressão com perdas, sem o compromisso de reconstrução dos dados, auxilia no processo de maximização das diferenças visuais entre classes de cenas;
2. As características discriminantes de uma cena, ao serem realçadas, tendem a aumentar sua complexidade intrínseca;
3. Transpondo a Complexidade de Kolmogorov, que descreve um objeto pela sua aleatoriedade, o valor da entropia algorítmica de uma cena pode ser utilizado como medida de classificação;
4. Relacionando o método à Teoria Algorítmica da Informação, outras medidas da Teoria da Informação Clássica podem ser utilizadas como avaliação de similaridade;
5. Visto que a literatura indica que é possível encontrar padrões entre pares de imagens utilizando a compressão de dados, é possível construir um classificador supervisionado treinado com poucas amostras.

1.4 Objetivos

Em consonância com as hipóteses elencadas para o desenvolvimento do presente trabalho, objetiva-se desenvolver um método de classificação de imagens similares baseado em compressão de dados. Para isso, pretende-se ainda:

1. Definir um modelo de treinamento baseado em compressão que abstraia detalhes e amplie as principais características dos objetos;
2. Determinar um método para mensurar a quantidade de informações nos objetos após as transformações de características;
3. Avaliar o desempenho em problemas de similaridade de cenas.

1.5 Proposta da Solução

O sistema cognitivo humano é capaz de selecionar as informações mais relevantes ao receber estímulos sensoriais. Este fenômeno mental é conhecido como atenção seletiva [137]. O processo da atenção seletiva é responsável por filtrar e considerar apenas uma pequena parcela da informação recebida como importante, para entendimento das mensagens ou percepção do ambiente. Segundo [137], a abordagem conhecida como Teoria da Atenuação do Sinal explica esse processo de seleção da atenção como um mecanismo de atenuação dos estímulos irrelevantes, atuando como enfraquecedor desse sinal. Os autores afirmam que a atenção visual é estudada por meio de dois modelos gerais: um modelo baseado na localização espacial dos estímulos e outro baseado nas características intrínsecas de cada objeto. O primeiro modelo é fundamentado nos aspectos fisiológicos dos estímulos, como os aspectos espaciais e temporais, conhecidos como processos de *baixa ordem*. O segundo processa a significância individual dos objetos de acordo com o sistema cognitivo humano, sendo conhecido como processo de *alta ordem*.

Ainda segundo [137], em condições normais da visão humana, ambos os processos, de baixa e alta ordem, realizam a seleção dos estímulos mais relevantes no campo de busca. Esse aspecto em comum pode ser relacionado aos métodos de compressão que, ao aceitarem perdas explorando redundâncias nas imagens, principalmente a psicovisual, permitem a manutenção apenas dos atributos relevantes que as descrevem. Dentro dessa perspectiva, o presente trabalho propõe um classificador de cenas que utiliza técnicas de compressão de dados com e sem perdas. Os processos de compressão com perda são responsáveis pela atenuação dos detalhes e manutenção das características relevantes das imagens. A aplicação de técnicas sem perda visa mensurar a quantidade de informação preservada após os processos de atenção seletiva do modelo, por meio da aleatoriedade intrínseca aproximada pelos compressores e fundamentada na Teoria da Complexidade de Kolmogorov.

Para construir o método de classificação de cenas é utilizada a quantização vetorial⁵. Esse método atua no desafio central da pesquisa: *a transformação de características, fazendo emergir os principais atributos de cada objeto, explorando as redundâncias interpixel e psicovisual das imagens*. Para calcular o limite superior da Complexidade de Kolmogorov (tamanho do arquivo comprimido), o resultado da quantização vetorial é comprimido com algoritmos de propósito geral. Adicionalmente, é utilizado um método de compressão baseado em dicionário em conjunto com um método de compressão

⁵Algoritmo de compressão com perda baseado na substituição de blocos de uma imagem por índices de um dicionário previamente gerado. Esse dicionário é obtido a partir de blocos extraídos de imagens similares e agrupados por proximidade no espaço euclidiano [82]. Quantização vetorial é considerada uma generalização da quantização escalar [83].

estatística.

Nos capítulos seguintes, a discussão da pesquisa é organizada da seguinte forma: no Capítulo 2 são apresentados os fundamentos do reconhecimento de padrões, a teoria da compressão de dados e discute-se a métrica precursora baseada em compressão para reconhecimento de padrões; no Capítulo 3 é realizado um estudo de dois *toolkits* para agrupamentos de dados de propósito geral e de trabalhos dedicados ao agrupamento de imagens baseado em compressão; descreve-se a construção e parametrização do classificador proposto no Capítulo 4; no Capítulo 5 são apresentados e discutidos os resultados obtidos; e no Capítulo 6 as conclusões e os pontos em aberto para trabalhos futuros são elencados.

Reconhecimento de padrões e compressão de dados

Neste capítulo são apresentadas as bases conceituais que permitem a utilização da compressão de dados para aprendizagem. Discute-se os fundamentos da área de reconhecimento de padrões, sob a ótica dos principais conceitos que permeiam o entendimento e a proposição de novos métodos. São apresentados os aspectos cognitivos do reconhecimento de cenas pelo sistema visual atento humano, discutidos sob as abordagens da psicologia e da neurociência. Motivado pelas etapas do método proposto, é fornecida uma visão geral dos processos de compressão e reconstrução de dados que exploram redundâncias e eliminam detalhes desnecessários para descoberta da informação. Por fim, introduz-se a teoria utilizada como justificativa para extração de semântica por meio da entropia algorítmica e as principais métricas de similaridade derivadas dessa teoria.

2.1 Fundamentos de Reconhecimento de Padrões

Segundo [14], o reconhecimento de padrões (*pattern recognition*), com origem na engenharia, e o aprendizado de máquina (*machine learning*), com origem na ciência da computação, podem ser vistos como duas perspectivas de um mesmo campo de pesquisa. Dependendo da aplicação, os objetos podem ser imagens, sinais ou qualquer tipo de dado que precisa ser classificado ou agrupado e são referenciados pelo termo genérico *padrões*.

Assim, [63] conceituam um padrão como sendo um arranjo de descritores denotados por *características*. As características são expressas por valores, obtidos diretamente pela observação e mensuração das propriedades, combinação de duas ou mais propriedades primárias ou transformação de dados, com capacidade de discriminação. A escolha dessas propriedades depende do domínio da aplicação e, de acordo com [154], a representação de um padrão é obtida organizando esses valores em *vetores de características*. Isso permite a categorização dos padrões em *classes de padrões* que, segundo [63], é realizada de acordo com propriedades em comum. Dado um conjunto de U características x_j , onde

$j = 1, 2, \dots, U$, o vetor de características do i -ésimo padrão $\mathbf{x}^{(i)}$ é definido como

$$\mathbf{x}^{(i)} = [x_1, x_2, \dots, x_U]^T, \quad (2-1)$$

onde T denota transposição [154]. Na prática, esse vetor de características pode ser uma estrutura complexa como uma imagem, uma sentença, uma mensagem de e-mail, uma série temporal, entre outras [113]. As classes de padrões ω_w são indicadas como $\omega_1, \omega_2, \dots, \omega_W$ onde $w = 1, 2, \dots, W$. [63]

De acordo com [50], qualquer método que incorpore amostras de treinamento no projeto emprega aprendizado, que pode ser classificado como: supervisionado, não-supervisionado e por reforço. Na *aprendizagem supervisionada* ou *preditiva*, assume-se a disponibilidade de um conjunto de dados de treinamento e projeta-se o classificador explorando essa informação conhecida *a priori* [154]. Segundo [113], o objetivo da aprendizagem preditiva é mapear as entradas \mathbf{x} para as saídas y dado um conjunto rotulado de pares (\mathbf{x}, y)

$$\mathcal{T} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^T, \quad (2-2)$$

onde \mathcal{T} é chamado *conjunto de treinamento* e T é o número de exemplos de treinamento. A maioria dos métodos assume que a variável resposta $y^{(i)}$ é um valor categórico ou nominal de um conjunto finito, ou seja, $y \in \{1, 2, \dots, W\}$, ou um valor escalar real. Quando $y^{(i)}$ é categórica, o problema é conhecido como *classificação* e quando $y^{(i)}$ está no domínio contínuo \mathbb{R} é conhecido como *regressão*. Quanto à quantidade de classes W envolvidas no problema, se $W = 2$ é chamada de *classificação binária*, ou seja, $y \in \{0, 1\}$; se $W > 2$ é chamada *classificação multi-classe*; ou se os rótulos das classes não são mutuamente exclusivos é chamada *classificação multi-rótulo*. Formaliza-se o problema de classificação como encontrar uma *função de aproximação* assumindo que

$$y = f(\mathbf{x}) \quad (2-3)$$

para uma função desconhecida f , com o objetivo de estimar a saída dessa função, dado um conjunto de treinamento \mathcal{T} , para fazer previsões usando

$$\hat{y} = \hat{f}(\mathbf{x}). \quad (2-4)$$

O objetivo do treinamento é *generalizar* a aprendizagem obtida a partir do conjunto de amostras conhecidas para novas entradas, o *conjunto de teste* \mathcal{D} [113]. A Figura 2.1 apresenta o modelo básico de um classificador estatístico que transita entre as etapas de treinamento e teste.

Nos métodos baseados em *aprendizagem não-supervisionada* ou *descritiva*, tam-

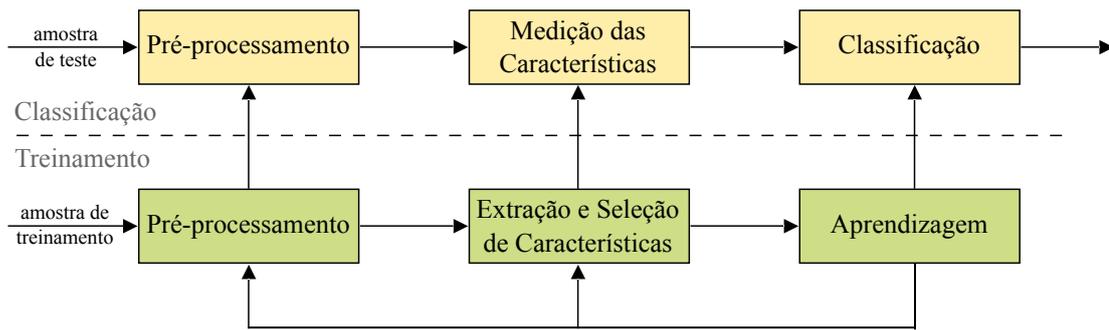


Figura 2.1: Modelo típico de um método de reconhecimento estatístico de padrões [73].

bém conhecida como *clusterização* ou *agrupamento*, não existe um modelo de treinamento explícito. Nesse caso, de acordo com [113], existem apenas as entradas \mathbf{x} de modo que

$$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^D. \quad (2-5)$$

Assim, os padrões de entrada são agrupados em clusters de forma explícita ou implícita pelo algoritmo [50], desvendando similaridades subjacentes [154]. Esse tipo de aprendizagem também é conhecida como descoberta de conhecimento, pois não são claras quais semelhanças devem ser encontradas e não há nenhuma métrica de erro. Ao contrário, a aprendizagem supervisionada permite comparar a predição de y dando um \mathbf{x} observado [113].

Na *aprendizagem por reforço*, a estratégia é treinar o classificador utilizando padrões de entrada rotulados, fazendo tentativas iterativas de predição das classes e utilizando os resultados em uma retroalimentação para melhorar o processo de decisão [50]. Nesse tipo de abordagem, os métodos se preocupam em como mapear situações em ações, de modo a maximizar a recompensa. Seus principais mecanismos de mapeamento são a tentativa-e-erro e a recompensa atrasada. Na tentativa-e-erro, o modelo não conhece *a priori* quais ações devem ser executadas, porém precisa descobrir quais geram mais recompensas. Na recompensa atrasada, o modelo assume que uma ação atual pode influenciar não somente a recompensa imediata mas também as subsequentes [153].

Quanto ao resultado do processamento de um algoritmo de reconhecimento de padrões, é importante pontuar que os dados podem ser classificados ou agrupados. Na *classificação*, os algoritmos se utilizam de padrões conhecidos, ou seja, já rotulados com suas respectivas classes, para treinar um modelo que seja capaz de prever as classes dos padrões desconhecidos. Esses algoritmos, conhecidos como *classificadores*, utilizam métricas de similaridade e dissimilaridade e são aplicados nos aprendizados supervisionado e semi-supervisionado. Já os algoritmos de *clusterização* ou *agrupamento* são projetados para encontrar semelhanças entre os padrões não rotulados, utilizando métricas de distân-

cia, e organizá-los utilizando o aprendizado não-supervisionado. Segundo [12], dividindo esse conjunto de padrões em poucos clusters, admite-se, necessariamente, a perda de detalhamento dessas semelhanças. Entretanto, obtém-se em contrapartida uma simplificação. Ainda segundo o autor, esse conceito pode ser sintetizado como: os clusters representam os padrões ocultos, a busca por esses padrões é realizada por métodos de aprendizagem não-supervisionada e o resultado é um conceito de dados, ou seja, uma descoberta de conhecimento.

A divisão dos algoritmos de aprendizagem não-supervisionada, de acordo com [154], pode ser realizada em algoritmos de clusterização *hierárquica* e *não-hierárquica*. Métodos de clusterização hierárquicos podem ser implementados por meio das abordagens *aglomerativa* e *divisiva*. Na abordagem aglomerativa, inicialmente, cada objeto dá origem a um cluster primário e, durante sucessivas iterações, esses clusters são recursivamente combinados, dando origem a classes progressivamente mais abrangentes. Na abordagem divisiva, os objetos são agrupados em um único cluster e, recursivamente, utilizando como critério as maiores distâncias, esse cluster é dividido até restar apenas um objeto por cluster [12]. Os métodos de clusterização não-hierárquica não são capazes de explorar todas as possibilidades de agrupamentos pois utilizam heurísticas de otimização iterativa para tornar a execução computacionalmente viável. Diferente dos métodos hierárquicos tradicionais, a cada iteração dos algoritmos não-hierárquicos, os clusters já formados são revisitados e uma nova divisão é realizada, melhorando gradualmente o particionamento [12].

2.2 Sistema Visual Atentivo Humano

A visão humana, dentre os cinco sentidos - visão, audição, olfato, paladar e tato - é considerada a que o homem depende mais e recebe a maior quantidade de informações captadas pelos receptores sensoriais [42]. Estima-se que a visão consiga captar dados de imagens com taxas superiores a 10 megabits por segundo (Mbps). Considerando que grande parte da informação recebida é redundante, as diversas camadas do córtex visual se encarregam de comprimi-la para dar significância sem perder dados importantes. Assim, nas etapas superiores do processo de cognição é processada apenas uma pequena fração dos dados [42].

Segundo [110], os organismos tendem a processar apenas um subconjunto de estímulos sensoriais. Na modalidade visual, os autores propõem uma análise da atenção seletiva em três dicotomias. A primeira trata da direção da atenção seletiva, que pode ser classificada como *top-down* ou *bottom-up*. Segundo [44], a direção *bottom-up* é reflexiva pois a atenção do indivíduo é atraída de forma automática para regiões salientes, ou seja, regiões que se diferenciam do contexto. Já na direção *top-down*, a atenção

é direcionada por um objetivo previamente determinado; nesse caso, portanto, ocorre uma busca pela informação dentro do contexto. Na segunda dicotomia, a atenção é dividida entre espacial (*location-based*) e baseada nas características das partes que compõem a imagem (*feature-based* ou *object-based*). Segundo [137], na atenção *location-based*, uma imagem é entendida a partir de uma representação primária, com dimensões simples e não integradas das partes que a compõe. Partindo dessa representação, após a integração das características da imagem, as informações captadas são processadas em operações cognitivas de alta ordem (*feature-based*). A terceira dicotomia distingue entre o processamento seletivo em *covert attention* e *overt attention*. A *covert attention* é caracterizada pela alternância, sem qualquer orientação de movimento, do foco da atenção de um ponto para outro. Na *overt attention* a atenção atende seletivamente a um ponto específico, em detrimento de outro, por meio do movimento dos olhos na direção indicada.

Dentro dessas abordagens cognitivas propostas por [110], que discutem o processamento visual pelo sistema atentivo humano, considera-se que as duas primeiras dicotomias podem ser aplicações da aleatoriedade¹ no processo de reconhecimento da informação. Os modelos teóricos de avaliação de imagens *bottom-up* e *location-based* são considerados reflexivo e primário, respectivamente. Na análise reflexiva de uma cena, a visão humana é direcionada pelos estímulos que influenciam nossa percepção. Na análise primária, a cena é entendida pela localização espacial dos objetos no campo visual. Nesses dois modelos teóricos o direcionamento da atenção não segue um objetivo conhecido *a priori*, sendo realizado apenas após a visualização completa da cena. Portanto, acredita-se que nesses casos a percepção é regulada pela *complexidade dos estímulos*, de acordo com o nível de abstração empregado pelo sistema visual atentivo.

De acordo com [59], a psicologia relaciona nossa percepção natural de aleatoriedade com a teoria matemática da complexidade algorítmica de Kolmogorov: quanto mais complexo o estímulo, mais aleatório será percebido. Em [47], o autor afirma que a complexidade visual pode ser avaliada utilizando abordagens matemáticas baseadas na teoria da informação algorítmica ou Teoria da Complexidade de Kolmogorov. No trabalho de [99], os autores apresentam um estudo para estimar a percepção humana em relação à complexidade visual de cenas. Os resultados indicam a compressão como um dos melhores preditores para este fim. Os autores afirmam que os processos visuais perceptivos e cognitivos, envolvidos na experiência subjetiva da complexidade, são realizados utilizando o processamento de informação não redundante. Essa experiência, entretanto, é

¹O presente trabalho discute a quantidade de informação presente nas imagens utilizando os termos sinônimos *aleatoriedade* e *entropia*. O termo *entropia* será abordado em duas variações: *entropia algorítmica* ou entropia intrínseca quando se referir à aleatoriedade da imagem considerando a ordem da informação e *entropia média* ou *entropia de Shannon* quando se referir à quantidade média de informação presente na imagem

influenciada, em maior ou menor grau, pelos aspectos semânticos do estímulo visual.

Em [64], os autores discutem a relação entre a esparsidade, considerada uma medida direta da aleatoriedade de um dado, e a quantidade de informação visual em cenas. Eles afirmam que a redundância presente nessas imagens permite que a informação relevante seja representada por meio de uma decomposição de funções-base. Todavia, imagens aleatórias não podem ser representadas com um pequeno número de elementos-base. Adicionalmente, [97] afirma que os sistemas sensoriais biológicos são naturalmente capazes de identificar propriedades estatísticas em sinais provenientes do ambiente natural. Em [7] e [9], os autores discutem essa capacidade biológica como a *hipótese da codificação eficiente*. De acordo com essa hipótese, o sistema sensorial é entendido como uma transformada que elimina redundâncias dos estímulos sensoriais de entrada [97], resultando em um conjunto de respostas neurais estatisticamente independentes [116]. Dentro dessa relação, pode-se afirmar que dar mais significância às características mais discriminantes de uma cena, torna o estímulo mais complexo para o ser humano.

Segundo [120], a representação de uma cena é composta tanto pela variedade de componentes (objetos) quanto de superfícies (texturas). Assim, intuitivamente, cenas complexas contém uma variedade maior de componentes e texturas, bem como relações mais densas entre eles [122]. Em [60], os autores estabelecem que a relação entre quantidade e variabilidade de informação em um objeto pode ser entendida como um parâmetro da complexidade visual. Assim, quanto mais informação um objeto possui, maior sua complexidade. Os autores ainda discutem que a complexidade visual pode influenciar negativamente o reconhecimento de objetos, devido à dificuldade de se estabelecer suas representações perceptivas. Entretanto, contrapõem essa ideia afirmando que, uma vez que essas representações estão estabelecidas, objetos complexos possuem menos representações candidatas e, portanto, podem ser reconhecidos de maneira mais eficiente.

Assim, um método computacional que aproxime a complexidade de Kolmogorov seria capaz de diferenciar cenas em classes de acordo com sua aleatoriedade. Em outras palavras, ao empregar, paralelamente, transformações em uma cena, realçando as características das diversas classes do domínio, a cena pode ser reconhecida como pertencente à sua classe correspondente pela versão modificada com a maior aleatoriedade.

Na abordagem da neurociência, de acordo com [77], o olho humano é constituído por recursos sofisticados na retina que permitem a decomposição das imagens em sinais baseados no contraste e no movimento. O córtex visual primário analisa esses sinais para determinar a forma dos objetos. Primeiro identifica os limites representados por diversos segmentos com orientação específica (*análise local de integração*). Em seguida, integra esses seguimentos em uma representação específica (*integração de contorno*). A análise local de integração é considerada um processamento visual de nível inferior, pois está relacionada com a identificação de elementos locais que compõe a estrutura da cena.

Ainda segundo os autores, o nível intermediário do processamento visual é responsável pela reunião dos elementos locais da imagem. Nesse nível ocorre a percepção unificada dos objetos de interesse e do plano de fundo.

O método computacional proposto nessa tese atua no nível inferior do processamento visual, detectando os estímulos espacialmente. Entretanto, é capaz de encapsular vestígios de um processo cognitivo, classificando as cenas baseado apenas na percepção dos estímulos visuais e cálculo da complexidade da cena. Analogamente, o sistema visual humano, em razão da limitação entre a quantidade de dados capturados e a capacidade de manipulação da informação pelos níveis superiores, realiza um pré-processamento para filtragem do conteúdo das mensagens recebidas na retina. Em termos computacionais, esse processo de filtragem é equivalente a um processo de compressão da informação. Como o nervo óptico tem apenas 1% de células ganglionares em comparação com a quantidade de células receptoras², o circuito da retina precisa tratar a informação nos fotorreceptores antes esta seja enviada para o encéfalo [77]. A execução desse pré-processamento, segundo [32], explica porque a visão humana é realizada inerentemente com perda, sendo necessária a filtragem da informação visual para lidar com a complexidade do mundo real e reconhecer as cenas. Além disso, deve ser considerado que a fóvea³ possui cerca de 1,5 milímetro de diâmetro, menos que 1% da área total da retina. Portanto, o restante da retina, provém apenas visão periférica de baixa resolução e toda a atenção visual é realizada apenas com as informações relevantes da imagem [32]. Dessa forma, a aplicação de técnicas de compressão de dados e a análise da complexidade da descrição algorítmica são justificadas para o desenvolvimento de um método de classificação de cenas. Esse classificador baseia-se nos aspectos do processamento visual humano discutidos pela psicologia cognitiva e pela neurociência.

2.3 Compressão de Dados

De acordo com [142], compressão de dados é a ciência de representar a informação de forma reduzida. Outra definição de [63], diz que compressão de dados é o processamento que objetiva reduzir o volume de dados necessários para representar uma informação. Ainda nessa definição, os autores destacam a diferença entre *dados* e *informação*: dados são os meios pelos quais uma informação pode ser representada e transmitida. Assim, como uma mesma informação pode ser representada utilizando diversas quantidades de dados, diz-se que a representação contém informações irrelevantes ou repetidas e, portanto, possuem *dados redundantes* [63]. Explorando essa redundância, são

²O olho humano é composto por dois tipos de células receptoras: os *cones* que reconhecem as cores e os *bastonetes* que detectam o nível de luminosidade,

³Região da retina responsável pelo detalhamento visual.

utilizados algoritmos para reduzir o número de bits necessários para representar uma imagem, um vídeo ou um áudio. Explorando as limitações biológicas do ser humano, também é possível descartar informação perceptualmente irrelevante. Essas representações compactas são criadas a partir das estruturas originais dos dados. Entende-se por dados os caracteres em um texto, números que representam áudio ou imagens, entre outros.

A *redundância relativa* dos dados, denominada R_C , indica a fração de dados redundantes de uma representação não comprimida de uma informação, sendo definida por [63] como

$$R_C = 1 - \frac{1}{T_C}, \quad (2-6)$$

onde T_C , denominada *taxa de compressão* é calculada como

$$T_C = \frac{|b|}{|b'|}. \quad (2-7)$$

Onde b e b' são, respectivamente, as quantidades de bits (ou unidades de transmissão de informações) das representações original e comprimida da informação. Os valores de T_C e R_C são interpretados da seguinte maneira: se $T_C = 10$, também expresso na forma 10:1, cada 1 bit de dados na representação comprimida da informação, codifica, em média, 10 bits de dados da representação original; como consequência, $R_C = 0,9$, indica que 90% dos dados são redundantes [63].

2.3.1 Redundância de Dados em Imagens

Segundo [129], as redundâncias de dados presentes nas imagens podem ser de três tipos: *redundância de codificação*, *redundância interpixel* e *redundância psicovisual*. A redundância de codificação explora os códigos utilizados para representação dos níveis de intensidade da imagem, quando a quantidade de bits não considera a probabilidade de ocorrência destes, ou seja, quando utiliza uma *codificação de tamanho fixo*. Por exemplo, em uma imagem em escala de cinza, onde os níveis de intensidade variam de 0 a 255, todos os 256 tons de cinza prováveis são codificados utilizando 8 bits cada: o valor 1 é codificado como 00000001, o valor 154 como 10011010, o valor 255 como 11111111. Em razão da maioria das imagens não serem formadas por níveis de intensidade equiprováveis, ou seja, seu histograma não é uniforme, não é razoável fixar o número de bits para cada valor de pixel para obter uma representação ótima em termos do volume de dados [63]. Nesse caso, utiliza-se a *codificação de tamanho variável* para atribuir menos bits para cada valor de pixel de acordo com sua probabilidade. Em [63], os autores explicam a utilização da codificação de tamanho variável presumindo que uma variável aleatória discreta a , no intervalo $[0, Q-1]$, seja utilizada para representar as intensidades de uma imagem de dimensões $M \times N$. A probabilidade de ocorrência

$P(a_i)$ de cada a_i é dada por

$$P(a_i) = \frac{n_i}{M \times N} \quad i = 0, 1, \dots, Q-1, \quad (2-8)$$

onde Q é o número de níveis de cinza utilizados para representar a imagem e n_i é o número de ocorrências de a_i . Optando por representar cada nível de cinza em uma quantidade variável de bits $l(a_i)$, obtida a partir de uma codificação hipotética e arbitrariamente fixada, são atribuídos menos bits a valores de a_i mais frequentes. O número médio de bits \bar{L} para representar cada pixel será definido por

$$\bar{L} = \sum_{i=0}^{Q-1} l(a_i)P(a_i). \quad (2-9)$$

A redundância interpixel, de acordo com [129], explora a característica de que a vizinhança de um dado pixel, em uma imagem não aleatória, normalmente possui alguma relação ou similaridade. O princípio é que os pixels são correlacionados espacialmente. Em razão dessa correlação, a informação transmitida por um único pixel, geralmente, é pequena, fazendo com que grande parte de sua contribuição visual seja redundante, uma vez que pode ser inferida a partir de seus vizinhos [63]. Explorando essa característica, inúmeras estratégias podem ser adotadas para transformar uma imagem em uma representação mais eficiente, porém, não-visual. Em [129], os autores discutem uma abordagem que armazena apenas as diferenças entre os pixels adjacentes. A técnica fornece valores menores que os originais, uma vez que a diferença entre os pixels vizinhos em imagem não-aleatórias tende a ser pequena. Essa abordagem integra o grupo de métodos de compressão diferencial como, por exemplo, o algoritmo *Differential Pulse-Code Modulation* (DPCM), mais utilizado em compressão de áudio [139]. Adicionalmente, uma técnica de redução da redundância de codificação pode ser utilizada conjuntamente para ampliar a taxa de compressão [129]. Em outra abordagem discutida por [63], a representação da imagem pode ser realizada por pares *run-length*⁴. Essa estratégia é aplicada com melhores resultados em imagens com número reduzido de intensidades e compostas de regiões formadas de pixels idênticos. Nesse caso, o mapeamento da imagem é realizado em tuplas de dois números: o primeiro representa um valor de intensidade e o segundo a quantidade sequencial de pixels com o mesmo valor.

Outro tipo de redundância importante, a psicovisual, pode ser explorada na compressão de representações visuais, se beneficiando da forma imprecisa que o ser humano percebe certos detalhes em uma imagem [129]. Como discutido nas estratégias anteriores, os métodos de compressão reduzem o volume da informação descobrindo,

⁴Pares de valor de intensidade e quantidade de pixels sequenciais que ocorrem na imagem [5].

reestruturando e recodificando dados redundantes. Na abordagem psicovisual utiliza-se um novo conceito, a irrelevância. Dessa forma, uma imagem pode ser comprimida removendo dados irrelevantes para o contexto ou o observador, mesmo que não contenha outro tipo de redundância [139]. Obrigatoriamente, qualquer imagem digital passa por um processo de eliminação de informações irrelevantes quando é adquirida pelo dispositivo de captura, durante os processos chamados amostragem e quantização. Segundo [63], uma cena pode ser contínua tanto em relação às coordenadas x e y quanto em relação à amplitude do sinal captado pelo sensor. Por ser contínua, presume-se que em uma imagem analógica existam infinitos pontos de coordenadas e de intensidade, o que a tornaria, em teoria, de tamanho infinito. Em linhas gerais, a amostragem é o processo de converter uma imagem analógica em uma matriz finita de pontos e o processo de quantização é responsável por discretizar os valores de intensidade luminosa nesses pontos. Assim, devido às próprias limitações dos sensores dos dispositivos, algumas informações sempre são perdidas, garantindo, de acordo com a aplicação da imagem, que não haja prejuízo ao nível de fidelidade necessário.

2.3.2 Conceitos de Teoria da Informação

A Teoria Matemática da Comunicação - mais conhecida com Teoria da Informação - fundada por Claude E. Shannon em 1948, é considerada a fundamentação teórica da compressão de dados e o marco para que inúmeras técnicas de compressão fossem propostas na prática [1]. Garantindo que a geração de informações pode ser modelada como um processo probabilístico, a Teoria da Informação de Shannon nos proporciona uma estrutura conceitual matemática para modelar o limite de compressão de uma imagem sem que haja perda de informação [63]. Em [1], os autores definem *fonte de memória zero* como uma fonte de informação em que os símbolos⁵ são estatisticamente independentes, e é descrita pelo alfabeto⁶

$$\mathcal{A} = \{a_1, a_2, \dots, a_Q\} \quad (2-10)$$

e as respectivas probabilidades de ocorrências associadas aos símbolos do alfabeto \mathcal{A}

$$\{P(a_1), P(a_2), \dots, P(a_Q)\}. \quad (2-11)$$

A relação entre a probabilidade de ocorrência de um símbolo e o tamanho do

⁵Símbolo é a unidade de informação utilizada para representar uma mensagem.

⁶Alfabeto é o conjunto de todos os símbolos possíveis utilizados para representar uma mensagem emitida por um fonte de informação.

código binário que o descreve é determinada por

$$I(a_i) = \log_2 \frac{1}{P(a_i)} = -\log_2 P(a_i), \quad (2-12)$$

onde $I(a_i)$ é a quantidade de conteúdo da informação para um símbolo a em termos da probabilidade de ocorrência $P(a_i)$ associada a esse símbolo. Diz-se, então, que o símbolo a_i pode ser melhor representado em aproximadamente $-\log_2 P(a_i)$ bits⁷. Como consequência, um símbolo com alta probabilidade de ocorrência é codificado com menos bits [1].

Essa definição nos fornece um conceito de quantificação média do conteúdo da informação presente em cada símbolo, também conhecida como *entropia da fonte*. Diferentemente do significado de desordem e caos, originalmente aplicado pela física na termodinâmica, a entropia, no sentido estatístico, mede o grau de surpresa ou incerteza [1]. A entropia média da fonte de informação, denominada H , representa o tamanho médio do código binário de todos os possíveis símbolos utilizados para representar uma imagem [1], sendo definida como

$$H = -\sum_{i=1}^N P(a_i) \log_2 P(a_i). \quad (2-13)$$

O limite de codificação mínima definido pela entropia de Shannon é conhecido como *Teorema da Codificação de Fonte* ou *Primeiro Teorema de Shannon*. Esse valor fornece o limite superior alcançável por um processo de compressão, que reconstrua de forma exata os dados originais [1].

2.3.3 Compressão e Reconstrução dos Dados

No processo de compressão existem dois algoritmos: um *algoritmo de compressão* que, dada uma entrada \mathcal{X} é gerada uma representação comprimida \mathcal{X}_c ; e um *algoritmo de descompressão* que, dada como entrada \mathcal{X}_c os dados são reconstruídos para \mathcal{Y} [142]. Após o processo de descompressão, se $\mathcal{Y} \equiv \mathcal{X}$, os dados foram reconstruídos de forma idêntica ao original, ou seja, o método utilizado foi de *compressão sem perda*. Se $\mathcal{Y} \approx \mathcal{X}$, os dados descomprimidos se aproximam da representação original, preservando informação suficiente para manter a compreensão e, portanto, o método utilizado foi de *compressão com perda*.

As técnicas de compressão sem perda fornecem uma representação exata dos dados originais após a descompressão. O fator restritivo dessas técnicas é o limite máximo de bits/símbolo definido pela entropia média de Shannon. Esse tipo de compressão

⁷O sinal “-” em $-\log_2 P(a_i)$ é utilizado como operação de módulo, em decorrência de que \log_2 de números menores que 1 é negativo. Caso $P(a_i)$ esteja em forma de fração, o sinal “-” é suprimido.

é normalmente aplicado em problemas que não toleram qualquer diferença entre os dados originais e os reconstruídos, como é o caso da informação textual, por exemplo. É aconselhável a utilização da compressão sem perda somente em situações que a informação reconstruída não fornecerá elementos precisos para tomada de decisão na entrada de outro processamento subsequente. Nesse caso, qualquer informação perdida *a priori* poderia ocasionar em uma solução errada.

Por outro lado, as técnicas de compressão com perda permitem a reconstrução dos dados de forma inexata, ou seja, os dados decodificados diferem dos dados originais até um limite aceitável para a aplicação desejada. Uma vantagem desse tipo de técnica é a possibilidade das taxas de compressão serem maiores que o limite definido pela entropia de Shannon. Para atingir o objetivo, muitas técnicas de compressão com perda procuram explorar a forma como os sentidos humanos percebem e eliminam dados irrelevantes e mantém o sentido da informação. No caso do sistema visual humano, tenta reproduzir sua percepção, ignorando o conteúdo da informação que excede sua capacidade de resolução ou a taxa de contraste [53].

De acordo com a aplicação do modelo de compressão, o nível de abstração determina como será realizado o agrupamento dos dados. Um exemplo notório, apresentado por [142], são os sistemas de codificação do Código Morse e do Braille, originalmente desenvolvidos para serem sistemas otimizados para transmissão e armazenamento da informação, respectivamente. Esses dois exemplos se utilizam da distribuição estatística dos dados para efetuar o processo de compressão, explorando a redundância de codificação. O Código Morse, desenvolvido por Samuel Morse em meados do século XIX, codifica letras, algarismos, pontuações e alguns outros caracteres especiais em pontos e traços para transmissão entre telégrafos. O Braille é um sistema de escrita tátil utilizado por pessoas portadoras de alguma deficiência visual ou com baixa visão. Além da representação de letras individuais em grades 3x2, permite a combinação de letras e palavras frequentemente utilizadas, de acordo com o idioma. Enquanto o Código Morse explora a codificação de tamanho variável, atribuindo menores sequências de pontos e traços para as letras mais utilizadas, o Braille é capaz de trabalhar em um nível de abstração maior ao combinar letras e palavras em representações menores.

Embora o reconhecimento de padrões seja utilizado em diversos domínios, não existe nenhuma teoria que unifique esse campo ou mesmo um método de propósito geral [70]. No entanto, a compressão de dados pode ser um dos componentes-chave [71]. Nessa linha, [22] afirma que compressão de dados, predição, classificação de padrões, aprendizagem e mineração de dados são faces de uma mesma (multidimensional) moeda. Essas observações podem ser reforçadas quando se considera que todas as abordagens de reconhecimento de padrões utilizam - ou necessitam para aprimoramento dos resultados - algum tipo de compressão dos dados de entrada, seja na redução de redundâncias

ou seleção das características mais discriminantes [70]. Nesse sentido, esse campo de investigação se mostra com grande potencial de exploração ao combinar duas áreas de pesquisas, com um objetivo em comum e promissor no avanço do estado da arte em reconhecimento de imagens.

2.3.4 Métodos de Compressão de Imagens

Quanto à forma como os algoritmos identificam, exploram e representam os tipos de redundância discutidos na Seção 2.3.1, eles podem ser divididos em métodos estatísticos, baseados em dicionários e baseados em transformadas, aplicados mais frequentemente em imagens. Os métodos estatísticos reúnem o grupo de algoritmos que exploram a redundância de codificação. Eles comprimem a informação por meio da atribuição de códigos menores a símbolos ou grupo de símbolos com maior probabilidade de ocorrência. Ao projetar e implementar esses algoritmos, dois problemas centrais devem ser tratados: (1) atribuir códigos que podem ser decodificados sem ambiguidade e (2) que tenham o tamanho médio mínimo [139]. O segundo problema, já visto na Seção 2.3.2, não é o único ponto que caracteriza boa codificação [142]. É necessário que uma informação seja *unicamente decodificável*. Essa restrição garante que para cada sequência finita de símbolos, a sequência correspondente de palavras-código⁸ seja diferente da codificação de qualquer outra sequência emitida pela fonte [68].

Um dos algoritmos estatísticos mais utilizados é a *codificação de Huffman*, desenvolvido por David Huffman em 1952 [142], por estar presente em grande parte dos formatos de arquivos comprimidos. Nesse algoritmo, todos os símbolos são ordenados de forma decrescente de acordo com suas respectivas probabilidades de ocorrência. Em seguida, são realizados sucessivos agrupamentos dos símbolos de menor ocorrência em forma de subárvores. Os grupos formados são reposicionados em ordem decrescente de acordo com a soma total das ocorrências dos símbolos que os compõem. O processo é completado quando toda a lista é reduzida em apenas uma árvore, que reúne todos os grupos de símbolos na raiz [139]. A partir da raiz da árvore, são atribuídos ao nós-filho os valores 0 e 1: 0 para os caminhos da esquerda e 1 para os caminhos da direita. O processo de codificação ocorre identificando o símbolo desejado e traçando o caminho de 0's e 1's percorrido da raiz até ele. A Figura 2.2 ilustra a codificação da string *CDBEFADDFACCBBCBDAD* utilizando os passos do algoritmo de *Huffman* descritos acima. A codificação resultante das palavras-código se aproxima do limite inferior definido pela entropia de uma fonte discreta sem memória [68].

⁸Palavra-código é o termo utilizado para designar uma sequência de bits que codifica um símbolo do alfabeto.

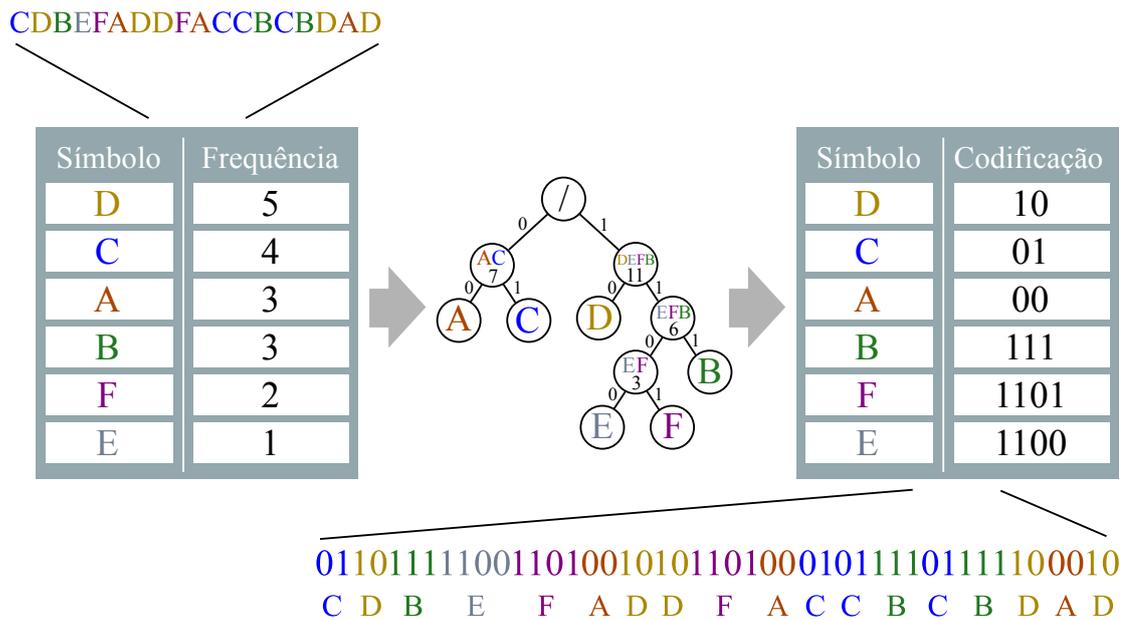


Figura 2.2: Processo de codificação de uma string pelo algoritmo de Huffman.

Os métodos que utilizam dicionários para representar os dados de uma informação comprimida, buscam eliminar repetições de símbolos e de conjuntos de símbolos. Essa busca, de acordo com [142], pode ser estática ou adaptativa. Segundo o autor, a busca estática é mais apropriada quando uma grande quantidade de informação da fonte é conhecida *a priori*. Já a busca adaptativa, mais amplamente utilizada, é capaz de comprimir dados de diversos tipos de fontes, sem necessitar de qualquer informação prévia. Em outras palavras, é capaz de se adaptar às características de qualquer fonte [142]. Uma importante característica desses métodos é que eles não necessitam de conhecimento *a priori* da probabilidade de ocorrências dos símbolos que serão codificados [63]. Entretanto, necessitam que grande quantidade de informação para aproximar do limite teórico da entropia [139]. Uma estratégia para induzir que esses métodos trabalhem próximo a esse limite é gerar redundância relevante [139].

Atualmente, a maioria das técnicas de compressão adaptativa por dicionário foram estendidas e melhoradas a partir das propostas de Jacob Ziv e Abraham Lempel, publicadas nos anos de 1977 e 1978 [142]. Segundo [68], a compressão realizada pelos algoritmos Lempel-Ziv (LZ77 e LZ78) segmenta o fluxo de dados da fonte da informação, de maneira que sejam encontradas subsequências mais breves possíveis que não tenham sido visualizadas anteriormente. Quando uma nova sequência é encontrada, um novo ponteiro é adicionado ao dicionário em referência a essa sequência. O LZW é uma das variações mais amplamente utilizadas e foi desenvolvida por Terry Welsh em 1984 [142]. O algoritmo LZW é iniciado com todos os possíveis símbolos do alfabeto no dicionário, ou seja, para símbolos de 8 bits, as primeiras 256 entradas do dicionário são ocupadas

antes de qualquer leitura dos dados. Com essa estratégia, o *buffer* de saída da compressão é contido apenas por ponteiros do dicionário, diferentemente da saída das versões originais do LZ, composta de pares de ponteiros e símbolos [139]. À medida que o codificador analisa sequencialmente os pixels da imagem, esses valores são concatenados em uma lista e adicionados sempre que formarem uma nova entrada, na primeira posição vazia do dicionário [63].

No caso particular das imagens, podem ser aplicados algoritmos de transformadas na estrutura dos dados. Essas transformadas não são capazes de realizar a compressão da informação em si. Entretanto, auxiliam na modificação da correlação dos dados adjacentes, tornando-os mais comprimíveis do que os originais. De forma geral, [142] afirma que as compressões por transformadas são compostas por três etapas. Na primeira etapa, a sequência de dados é dividida em blocos, mapeados em uma nova sequência transformada. Em seguida, na segunda etapa, essa sequência é quantizada, levando em consideração, principalmente, a distorção dos dados aos serem reconstruídos. Por fim, na terceira etapa, os valores quantizados são codificados. Na etapa de quantização ocorre a eliminação das informações menos importantes, identificadas após a realização da transformada. As informações mais relevantes são preservadas e passam pelo processo de codificação, podendo ser aprimorado com o uso de métodos estatísticos para supressão da redundância de codificação. A Transformada Discreta de Cosseno (DCT - *Discrete Cosine Transform*) e a Transformada Discreta de *Wavelet* (DWT - *Discrete Wavelet Transform*) são técnicas muito utilizadas em diversos formatos de imagens em conjunto com outros algoritmos.

2.3.5 Compressão por Quantização

Devido o tipo numérico da representação das imagens, a técnica da quantização pode ser utilizada para redução do conjunto de cores. O termo *quantização* é definido como o processo de restringir uma quantidade variável de valores a um conjunto discreto [139]. Em outras palavras, é o processo de atribuir *valores representativos* a intervalos de valores de entrada [69], discretizando e/ou simplificando um sinal. A *quantização escalar* é um tipo de quantização intuitiva, que permite controlar a relação entre a taxa de compressão e a quantidade de informação perdida na imagem. Entretanto, em razão da simplicidade do mapeamento entre os dados originais e o resultado da compressão, é recomendada sua aplicação apenas em casos onde é tolerada a perda de muita informação [139].

A quantização consiste em dois componentes: um codificador e um decodificador de mapeamento. O codificador divide a faixa de valores emitidos pela fonte da informação em intervalos, conforme a quantidade de símbolos da representação comprimida. Para cada intervalo é atribuída uma palavra-código e definido um valor representativo,

normalmente a média, a mediana ou qualquer outra medida de centralidade do intervalo. Essa operação torna o mapeamento irreversível. O decodificador, ao receber como entrada a informação comprimida, utiliza o mapeamento para traduzir as palavras-código em seus respectivos valores representativos [142].

A *quantização de cores* é um tipo específico de quantização escalar aplicada em imagens, cujo objetivo é selecionar um conjunto de cores reduzido que substitua a gama de cores originais. Esse conjunto é obtido mapeando espaço de cores completo para as cores representativas, de modo que seja preservada o máximo de informação visual possível [69]. Assim, dado o conjunto $\mathcal{C} = \{\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \dots, \mathbf{c}^{(T)}\}$ de T pontos de entrada, representando cores, o processo de quantização de cores é definido por [69] como:

- a) um conjunto \mathcal{R} de λ pontos de saída ou cores representativas $\mathbf{v}^{(i)}$

$$\mathcal{R} = \{\mathbf{v}^{(i)}\}_{i=1}^{\lambda}; \quad (2-14)$$

- b) um conjunto de partições do espaço de entrada em regiões ou células de quantização

$$\mathcal{P} = \{p^{(i)}\}_{i=1}^{\lambda}; \quad (2-15)$$

- c) um mapeamento das cores de entrada para os índices das cores representativas

$$g(\mathbf{c}) = i \text{ se } \mathbf{c} \in p^{(i)}; \quad (2-16)$$

- d) uma função de quantização que mapeia os pontos de entrada em pontos de saída

$$q(\mathbf{c}) = v[g(\mathbf{c})]. \quad (2-17)$$

Considerando as quatro fases de um algoritmo de quantização de cores proposta por [69]:

1. amostragem da imagem para determinar a distribuição de cores,
2. seleção do novo mapa de cores baseada na distribuição,
3. mapeamento da quantização e
4. reconstrução da imagem,

o autor afirma que a etapa da seleção do novo mapa de cores é a mais importante e desafiadora. O *median cut* é um algoritmo bem eficiente para determinar esse mapa de cores descrito na Equação 2-14. Segundo [69], o conceito desse algoritmo é utilizar cada uma das cores representativas $\mathbf{v}^{(i)}$ para substituir um número igual de pixels $\mathbf{c}^{(i)}$ na imagem original. Para isso, o algoritmo, repetidamente, subdivide o espaço de cores em partições $p^{(i)}$, tomando como referência a mediana dos pontos, até que número de partições seja igual à quantidade de cores representativas desejada. Portanto, nesse

algoritmo, o número de partições é sempre 2^b , onde b é a quantidade de bits das palavras-código utilizadas na codificação das cores quantizadas. Ao final do particionamento é calculada a média, ou centroide, dos pontos em cada partição para determinar sua respectiva cor representativa. A Figura 2.3 ilustra um conjunto de cores identificadas com o símbolo x , na visão dos eixos verde e azul. As áreas retangulares mostram as regiões do espaço de cores que foram , utilizando os pontos medianos que fornecerão os centroides para representação das cores quantizadas.

Segundo [142], a *quantização vetorial* é outra especialização da técnica de quantização, sendo popularmente aplicada em compressão de imagens. Entretanto, para uma mesma taxa de bits por palavra-código, a utilização a quantização vetorial resulta em distorções menores que a quantização escalar na reconstrução das imagens [142]. Esse tipo de quantização explora os vetores obtidos a partir do janelamento não-sobreposto das imagens de treinamento, chamados de *vetores de treinamento*. A concentração desse vetores no espaço U -dimensional determina os pontos de saída do quantizador, chamados de *codevectors* [139]. Considerando que os pixels em uma imagem são correlatos, os vetores de maior proximidade entre si e que representam essas janelas da imagem, tenderão a se agrupar no mesmo cluster [142].

Em linhas gerais, o processo de compressão ocorre com a codificação das janelas da imagem - de tamanho $M \times N$ pixels, normalmente com $M = N$ - na forma de índices, de acordo com o critério de proximidade. Esses índices são obtidos do dicionário produzido pela saída do quantizador, chamado de *codebook*. Cada índice do *codebook* identifica um *codevector*, que são os vetores representativos. Os *codevectors* são os centroides dos clusters criados a partir do agrupamento dos vetores de treinamento mais próximos. No

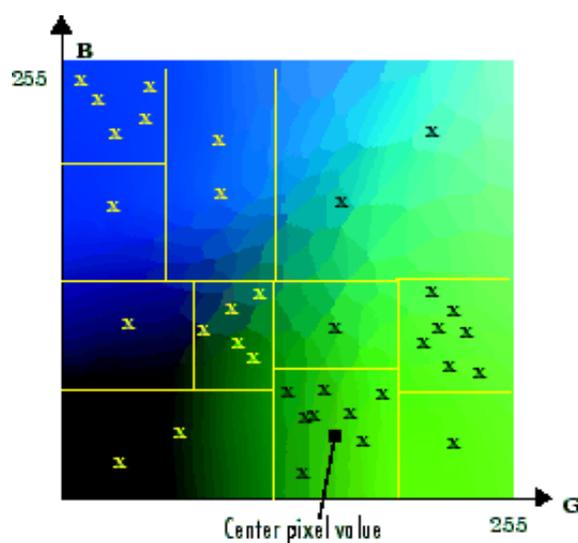


Figura 2.3: Quantização de amostras de cores pelo particionamento do espaço RGB sob a visão dos eixos verde (G) e azul (B) [104].

processo de descompressão, a imagem é reconstruída substituindo os índices pelos *codevectors* correspondentes a partir do *codebook*. O tamanho do janelamento e a quantidade de *codevectors* influencia diretamente no resultado visual da reconstrução [142]. Quanto maior o tamanho das janelas de treinamento, mais detalhes finos são perdidos; quanto maior o tamanho do *codebook*, mais próxima da imagem original é a reconstrução no geral. Uma definição geral da quantização vetorial é apresentada por [118], na qual dado um vetor de entrada \mathbf{c} U -dimensional

$$\mathbf{c} = [c_1, c_2, \dots, c_U]^T, \quad (2-18)$$

um mapeamento completo do *codebook*, de cada vetor de entrada para as saídas do quantizador [139, 142]

$$q(\mathbf{c}) = \mathbf{v}^{(i)}, \quad (2-19)$$

onde $\mathbf{v}^{(i)} \in \{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots, \mathbf{v}^{(\lambda)}\}$ é um *codevector* correspondente à entrada. O conjunto de partições do espaço de entrada induzido por q é denotado por

$$\mathcal{P} = \{p^{(1)}, p^{(2)}, \dots, p^{(\lambda)}\}, \quad (2-20)$$

onde $\mathbf{c} \in p^{(i)}$ se $q(\mathbf{c}) = \mathbf{v}^{(i)}$. Então $\alpha(\mathbf{c}) = i$ se $\mathbf{c} \in p^{(i)}$ é o *codificador* e $\beta(i) = \mathbf{v}^{(i)}$ é o *decodificador*. De acordo com [79], uma vez construído o *codebook*, a aproximação de \mathbf{c} , realizada pelo codificador, é determinada encontrando o *codevector* $\mathbf{v}^{(r)}$ mais próximo de \mathbf{c} . O índice r é definido implicitamente pelo processo de decisão

$$r = \arg \min_i \{\|\mathbf{c} - \mathbf{v}^{(i)}\|\}, \quad (2-21)$$

onde a norma é euclidiana.

Como sugerido por [78], os problemas de clusterização são encontrados tanto nas áreas de reconhecimento e classificação de padrões e mineração de dados quanto na compressão de dados. Devido à alta dimensionalidade do espaço onde os vetores de treinamento da quantização vetorial são representados, a obtenção do *codebook* é uma tarefa exaustiva que depende de técnicas de clusterização [142]. Grande parte das abordagens para projeto de quantizadores vetoriais são baseadas no problema de clusterização *k-means* [142]. Sugere-se como heurística para resolução desse problema o *algoritmo de Lloyd*, que consiste em um simples esquema iterativo que encontra uma solução localmente mínima [78]. Em razão dessa heurística ter sido proposta para processar valores escalares grandes, [142] apresenta o algoritmo Linde-Buzo-Gray (LBG) [90] como uma generalização do algoritmo de Lloyd, que pode ser aplicado à quantização vetorial de imagens. Embora existam inúmeras abordagens para obter os *codebooks*, a maioria delas são baseadas no algoritmo LBG [142].

O diagrama de Voronoi é uma representação muito utilizada para ilustrar a divisão do espaço de entrada em um número finito de partições, com seus respectivos *codevectors*. Esse particionamento em regiões, limitadas por linhas (em geral, hiperplanos), é realizado de modo que, em cada região, haja um vetor de referência como vizinho mais próximo de qualquer outro vetor de entrada na mesma partição [79]. O algoritmo LBG é uma técnica que divide as partições do diagrama de Voronoi até que o número desejado de vetores representativos seja alcançado [45].

Mesmo sem garantir que a execução convergirá para uma solução ótima, o algoritmo LBG assegura que a distorção não aumentará ao longo das sucessivas iterações [142]. O algoritmo é iniciado com apenas um ponto no centro de massa dos vetores de treinamento, ou seja, o *codebook* é formado por apenas um *codevector*. Nas iterações seguintes, o tamanho do *codebook* é dobrado, sendo composto pelos *codevectors* da iteração anterior e da atual, acrescidos de um valor parametrizado de distorção ϵ (Figura 2.4). A cada iteração é calculada a distorção média Ψ entre os *codevectors* e os vetores de treinamento localizados em suas respectivas partições. As iterações são encerradas quando $\frac{\Psi_{\text{atual}} - \Psi_{\text{anterior}}}{\Psi_{\text{atual}}} \leq \epsilon$. A execução é encerrada quando o tamanho desejado do *codebook* é atingido. Para o caso onde tem-se um conjunto de treinamento disponível, [142] descreve formalmente o funcionamento do algoritmo LBG nos seguintes passos:

1. O algoritmo inicia com um conjunto de *codevectors*

$$\mathcal{V} = \{\mathbf{v}_{(k)}^{(i)}\}_{i=1}^{\lambda} \quad (2-22)$$

na origem do espaço, onde λ é número desejado de *codevectors*, e um conjunto de T vetores de treinamento

$$\mathcal{T} = \{\mathbf{c}^{(n)}\}_{n=1}^T. \quad (2-23)$$

Com k representando a k -ésima iteração, é atribuído $k = 0$ e a distorção inicial $\Psi_{(0)} = 0$. Então, o valor de convergência ϵ é selecionado.

2. O conjunto de partições

$$\mathcal{P} = \{p_{(k)}^{(i)}\}_{i=1}^{\lambda} \quad (2-24)$$

é dado por

$$p_{(k)}^{(i)} = \{\mathbf{c}^{(n)} : d(\mathbf{c}^{(n)}, \mathbf{v}^{(i)}) < d(\mathbf{c}^{(n)}, \mathbf{v}^{(j)}) \forall j \neq i\} \quad i \wedge j = 1, 2, \dots, \lambda \quad (2-25)$$

assumindo que nenhuma partição está vazia, onde d é uma função de distância entre os vetores de treinamento e os *codevectors*.

3. Compute a distorção média $\Psi_{(k)}$ entre todos vetores de treinamento e os *codevectors*.
4. Se $\frac{\Psi_{(k)} - \Psi_{(k-1)}}{\Psi_{(k)}} \leq \epsilon$, pare; caso contrário, continue.
5. Inicie uma nova iteração com $k = k + 1$. Encontre os novos *codevectors* $\{\mathbf{v}_{(k)}^{(i)}\}_{i=1}^{\lambda}$ que são os centroides de cada partição $p_{(k-1)}^{(i)}$. Continue para o Passo 2.

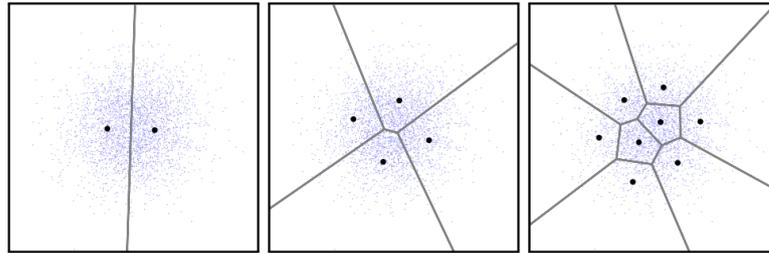


Figura 2.4: Diagramas de Voronoi criados pelas três primeiras iterações do algoritmo LBG em um espaço bidimensional [45]. Os pontos azuis são os vetores de treinamento e os pontos pretos são os centroides de cada partição, os *codevectors*.

2.3.6 Formatos de Imagens

Inúmeros formatos de arquivo estão disponíveis para representação e armazenamento das imagens digitais, conforme definição apresentada por [63] na Seção 1.1. Dentre eles, alguns dos mais populares são: *Bitmap Image File* (BMP), *Graphics Interchange Format* (GIF), *Portable Network Graphics* (PNG), *Joint Photographic Experts Group* (JPEG) e *Tagged Image File Format* (TIFF) e o recente formato WebP. A Tabela 2.1 sintetiza os principais algoritmos de compressão utilizados em cada um desses formatos.

- *Bitmap Image File* (BMP): Um dos formatos de imagens mais simples, desenvolvido para ser o formato de imagem nativo dos sistemas operacionais Microsoft Windows. Suporta versões com 1, 4, 8, 16, 24 e 32 bits por pixel. Embora seja mais comum em arquivos não comprimidos, o BMP possui suporte à compressão *run-length* (RLE), recomendado apenas em imagens com grandes blocos de cores idênticas [108];

⁹Apenas para codificações de imagens com 4 e 8 bits por cor.

¹⁰Suporta até 8 bits de cor por pixel, ou seja, 256 cores escolhidas da paleta RGB completa.

¹¹Com possibilidade de compressão com perda, preprocessando a imagem para reduzir cores.

¹²Além de suportar 24 bits de cor por pixel, pode ser utilizando o canal de transparência alfa (A), totalizando 32 bits por pixel.

¹³Existe um modo sem perda que, por ser ineficiente, não costuma ser implementado [139].

Tabela 2.1: *Comparativo de métodos de compressão utilizados nos principais formatos de imagens.*

Formato	Tipo	Sistema de cores	Estatística	Compressão		
				Dicionário	Transformada	Outro
BMP	Sem perda	RGB	-	-	-	RLE ⁹
GIF	Sem perda	RGBA ¹⁰	-	LZW	-	-
PNG	Sem perda ¹¹	RGBA ¹²	<i>Huffman</i>	LZ77	-	-
JPEG	Com perda ¹³	YCBCR	<i>Huffman</i>	-	DCT DWT	DPCM RLE
TIFF	Com/sem perda	RGB YCBCR CIELab CMYK	<i>Huffman</i>	LZ77 LZW	DCT	RLE
WebP	Com/sem perda	ARGB	<i>Huffman</i>	LZ77	DCT	-

- *Graphics Interchange Format (GIF)*: Permite o armazenamento de imagens utilizando uma paleta de cores definida no cabeçalho do arquivo, com no máximo 256 cores codificadas com até 24 bits por pixel. Essa restrição é tanto um fator limitador para representação da imagem quanto uma vantagem, pois permite boas taxas de compressão. Outra vantagem desse formato é possibilitar o armazenamento de múltiplas imagens dentro de um mesmo arquivo, permitindo, assim, animações [51]. A compressão dos dados é feita por uma variação do algoritmo LZW, usada para codificar imagens gráficas indexadas por bits, desenvolvida pela *Compuserve Information Service*, empresa responsável pelo formato [115];
- *Portable Network Graphics (PNG)*: Formato de arquivo que utiliza compressão sem perda, criado para melhorar e substituir o formato GIF, sem a necessidade de licenciamento de patente para uso [125]. Na representação dos pixels é suportado paleta de cores, níveis de cinza, RGB, além de transparência. A compressão é realizada pelo algoritmo DEFLATE, que combina os algoritmos LZ77 e *Huffman* [108]. É considerada a melhor alternativa ao JPEG para armazenamento de imagens formadas por texto, linhas e transições abruptas que não são transformadas no domínio da frequência com eficiência [125];
- *Joint Photographic Experts Group (JPEG)*: O padrão JPEG é atualmente o mais utilizado para armazenamento e transmissão de imagens, implementando a compressão com perda definida por um parâmetro de qualidade [1]. Suporta imagens coloridas de até 24 bits por pixel e não utiliza paletas indexadas de cores [51]. Possui uma versão sem perda, que utiliza a DPCM, e outra com perda que utiliza a transformada DCT, para eliminação de altas frequências na etapa de quantização. Na versão com perda, os coeficientes preservados da transformada são codificados utilizando *Huffman* ou uma codificação aritmética [139]. A versão melhorada JPEG2000 utiliza a transformada DWT. Com essa proposta sofisticada, porém não complexa, o JPEG tem grande eficiência nas taxas de compressão, principalmente para cenas [108].
- *Tagged Image File Format (TIFF)*: Suporta diversos esquemas de compressão, mas também é usado sem compressão [1]. Essa versatilidade, de acordo com

[125], o torna muito útil como um método genérico para aplicação em edição de imagens, mas também dificulta a manutenção da compatibilidade. A flexibilidade se estende tanto para os sistemas de cores utilizado na representação quanto para os algoritmos de compressão. Os espaços compatíveis incluem o RGB, YCbCr, CIELab e CMYK, tanto em formato de paletas de cores quanto *true-color*, que inclui todas cores possíveis do sistema. Quanto à compressão, as extensões incluem, além o algoritmo básico LZW, os esquemas de compressão do JPG (transformada DCT, RLE e codificação de *Huffman*) e do DEFLATE (LZ77) [2].

- WebP: Formato desenvolvido pela Google para imagens fotográficas da web, que possui desempenho superior na representação dos dados comprimidos com e sem perda. Inclui suporte à transparência do sistema de cores ARGB. Sua implementação permite a parametrização da relação taxa-distorção-complexidade. Na compressão com perdas, utiliza a DCT, a codificação preditiva e explora blocos de tamanho variável. Já na versão sem perdas é utilizado LZ77, codificação de *Huffman* e cache de cor. Mantendo o mesmo nível de qualidade, testes reportam que o WebP chega a ser 39,8% mais eficiente que o JPG [61].

2.4 Algoritmos de Classificação de Cenas

A classificação de cenas é um problema que envolve a existência de múltiplos objetos em uma única imagem, comumente carregada de semântica em um contexto complexo [143]. De acordo com [165], o reconhecimento de cenas é amplamente utilizado em aplicações de interação homem-computador, robótica, vigilância inteligente e carro autônomo; outras aplicações incluem rotinas de indexação em bancos de imagens, sensoriamento remoto e educação [30, 143]. O reconhecimento de cenas também é considerado pré-requisito em outros problemas de visão computacional como recuperação de imagens baseada em conteúdo (CBIR - *Content-Based Image Retrieval*) e detecção de objetos [165]. É comum que a classificação de cenas seja subdividida em cenas internas (*indoor*) e cenas externas (*outdoor*) [128, 133, 167, 29, 156].

Em relação à representação das características, segundo [143], o problema de classificação de cenas pode ser basicamente resolvido utilizando 3 abordagens: 1) extração de características de baixo nível, 2) extração de características de médio nível e 3) extração automática de características. Na primeira abordagem, a cena é entendida como um conjunto de características com pouca ou nenhuma carga semântica individual; mas quando combinadas têm sentido para entendimento da cena. Na segunda abordagem, de médio nível, a cena é avaliada por meio de detectores dos objetos que a compõe.

Na terceira abordagem, os autores incluem os métodos de aprendizado profundo (*Deep Learning*) como extração automática de características.

Para [103], os descritores de baixo nível (*low-level*) são extraídos diretamente dos pixels da imagem (sinal) e podem ser histogramas, textura, formas, entre outros. Segundo [156], a desvantagem dos descritores de baixo nível é ignorar a informação espacial e estrutural do fundo e dos objetos das imagens. Os descritores de médio nível (*mid-level*) são obtidos a partir da integração dos descritores de baixo nível e envolvem, por exemplo, a detecção de objetos, segmentação de regiões e casamento de similaridades. Por fim, os descritores de alto nível (*high-level*) simulam a forma como o ser humano interpreta a informação semântica. A Figura 2.5 mostra como estão organizados os descritores em relação à representação primária do sinal.

Na prática, os métodos de *Deep Learning* integram a abordagem de extração de descritores de baixo nível e a combinação destes para criação de descritores de nível intermediário, que formam representações de média complexidade [15]. Segundo [76], o aprendizado profundo também permite extrair características de alto nível de abstração combinando descritores de baixo nível.

Algoritmos tradicionais de *machine learning* ainda são opções de escolha para a resolução de problemas de classificação de cenas. Dentre eles, podem ser citados trabalhos recentes com *k*-Vizinhos mais Próximos¹⁴ (*k*-NN - *k*-Nearest Neighbors) [16], Máquinas de Vetores de Suporte¹⁵ (SVM - *Support Vector Machines*) [100, 152, 16],

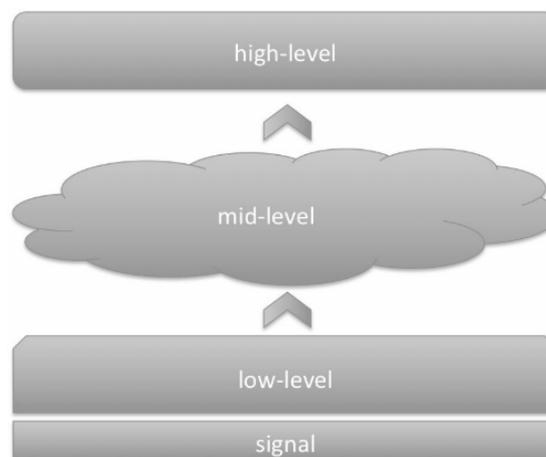


Figura 2.5: Organização dos níveis de descritores de uma imagem [103].

¹⁴Algoritmo de aprendizado dito "preguiçoso" (*lazy learning*) cujos resultados são baseados na distância de amostras similares. A tomada de decisão é realizada por votação majoritária entre os *k* representantes mais próximos das classes [123].

¹⁵Método para cálculo do hiperplano de separação ideal entre as classes, de modo que o erro de classificação para amostras de teste seja minimizado [100]. Para isso, os padrões localizados nas fronteiras são utilizados para maximizar a distância extraclasse.

Árvores de Decisão¹⁶ (*tree decision*) [167, 16], *k*-Médias¹⁷ (*k-Means*) [152], Análise de Discriminante Linear¹⁸ (LDA - *Linear Discriminant Analysis*) [100, 75] e o método Saco-de-Palavras¹⁹ (BoW - *Bag-of-Words*) [133].

De acordo com [84], classificadores tradicionais exigem extratores de descritores que resolvam a dualidade entre discriminabilidade entre os objetos e invariância aos aspectos irrelevantes do problema. Ainda segundo os autores, a solução está no projeto de extratores especializados no problema, mas que requerem grande conhecimento do domínio de aplicação. A principal vantagem do aprendizado profundo é a extração automática dessas características. Essa vantagem faz com que os modelos de *Deep Learning* tenham grande flexibilidade e sejam capazes de extrair semântica das cenas com desempenho eficiente e alta acurácia [143].

Entretanto, [143] apontam algumas desvantagens dos modelos de *Deep Learning*, entre elas: arquiteturas complexas que demandam grande capacidade computacional; a extração de características e a classificação não são realizadas de maneira transparente; e a necessidade de grandes volumes de dados para treinamento dos modelos. Em relação a essa última desvantagem, [76] afirmam que mesmo utilizando o recurso de aumento de dados (*data augmentation*), na prática são necessárias, pelo menos, algumas centenas de amostras para treinamento dos modelos. Outro ponto desfavorável é que os modelos são sensíveis ao contexto do problema em que foram treinados [76], remetendo novamente à necessidade de um grande volume de exemplos que representem ao máximo situações para generalização da aprendizagem. Embora o aprendizado profundo aborde a questão da dimensionalidade dos dados, não a trata. Ao contrário, a alta dimensionalidade é compensada pela grande quantidade de exemplos.

2.5 Medida para Avaliação de Sobreposição de Classes *R-value*

Em [107], o autor afirma que a modelagem do espaço de características, selecionando as mais importantes, pode ser utilizada como etapa complementar para aumentar a eficiência dos métodos de reconhecimento de padrões. O objetivo da seleção de ca-

¹⁶Modelo preditivo que divide recursivamente o espaço de características em subespaços, de modo que a partir de cada subespaço seja obtida uma função individual de predição [136].

¹⁷Método iterativo não-determinístico de aprendizagem não-supervisionada. É baseado na divisão do espaço de características em *k* clusters, de modo que as amostras pertençam ao cluster mais próximo da média [147].

¹⁸Algoritmo que conserva a informação discriminante em um espaço otimizado para um domínio transformado [75].

¹⁹Baseado em descritores de imagem que consistem em um conjunto de termos que descrevem a percepção de entrada [133].

racterísticas é obter, a partir das observações iniciais dos objetos, um espaço qualificado como: baixa dimensionalidade, manutenção das características discriminantes, separabilidade entre as classes e consistência entre as características de todas as amostras [107].

De acordo com [119], a qualidade de uma base de dados (*dataset*) é fator determinante para o desempenho de qualquer método de classificação. Essa qualidade está diretamente relacionada ao grau de sobreposição (*overlap*) entre as classes. Para quantificar o grau de *overlap* entre conjuntos de dados U -dimensionais é proposta a medida *R-value* [119].

Intuitivamente, suponha-se que C_1 a C_2 são diferentes conjuntos de um problema e $\mathbf{x}^{(1)}$ é um elemento U -dimensional de C_1 . Se $\mathbf{x}^{(1)}$ está contido em uma área de sobreposição com o conjunto C_2 , então muitos elementos de C_2 serão próximos de $\mathbf{x}^{(1)}$; ao passo que há também elementos $\mathbf{x}^{(i)}$ pertencentes a C_1 em uma área sem interseção. A Figura 2.6 ilustra um exemplo com dois conjuntos parcialmente interseccionados. O algoritmo *R-value* busca pelos k -vizinhos mais próximos de $\mathbf{x}^{(i)} \in C_w$ e conta quantos destes pertencem ao conjunto C_m . Ao final, a soma das ocorrências, dado um *threshold* como limite, determina o grau de interseção, normalizado no intervalo $[0..1]$ [119]. Portanto, o *R-value* é capaz de capturar o grau de interseção de conjuntos espacialmente dispostos em um plano U -dimensional, dada a dispersão de seus elementos.

A seguir, é apresentada a descrição formal do *R-value*, com a seguinte notação básica:

- $\omega_1, \omega_2, \dots, \omega_W$: classes do *dataset* f onde W é o número de classes;
- \mathcal{U} : conjunto universo de todos os objetos do *dataset* f ;
- $\mathcal{D}(\omega_w)$: conjunto de objetos pertencem à classe ω_w ;
- $|\mathcal{U}|$: cardinalidade do conjunto \mathcal{U} ;
- k : número de vizinhos mais próximos de um objeto;
- $\mathbf{x}_w^{(i)}$: i -ésimo objeto da classe ω_w ;
- $\mathcal{N}(\mathbf{x}_w^{(i)})$: conjunto de k -vizinhos mais próximos do objeto $\mathbf{x}_w^{(i)}$;

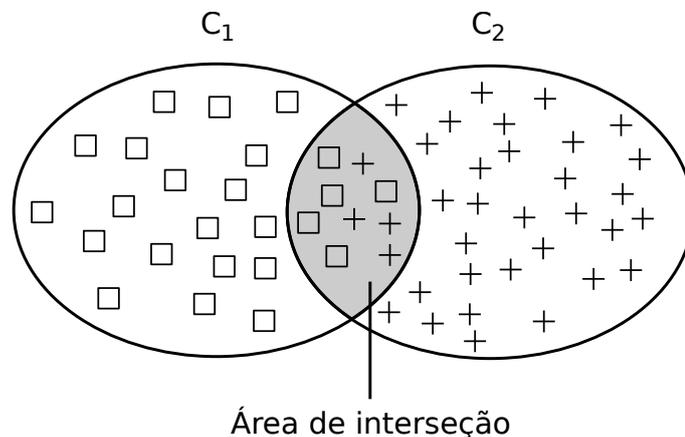


Figura 2.6: Área de interseção entre 2 conjuntos [119]

- $\mathcal{N}(\mathbf{x}_w^{(i)}, \omega_m)$: subconjunto de objetos em $\mathcal{N}(\mathbf{x}_w^{(i)})$ que pertencem a uma classe diferente ω_m .

O *R-value* entre duas classes ω_w e ω_m é definida pela equação

$$R(\omega_w, \omega_m) = \frac{1}{|\omega_w| + |\omega_m|} [r(\omega_w, \omega_m) + r(\omega_m, \omega_w)] \quad (2-26)$$

onde o cálculo parcial $r(\omega_w, \omega_m)$ é definido por

$$r(\omega_w, \omega_m) = \sum_{i=1}^{|\omega_w|} v(|\mathcal{N}(\mathbf{x}_w^{(i)}, \omega_m)| - \theta) \quad (2-27)$$

em que θ é o limite aceitável de vizinhos pertencentes a outra classe para $\mathbf{x}_w^{(i)}$ não ser considerado em uma região de interseção, estabelecido, razoavelmente em $0 \leq \theta \leq k/2$; $v(a) = 1$ se $a > 0$, senão $v(a) = 0$; e $\mathcal{N}(\mathbf{x}_w^{(i)}, \omega_m)$ é definido como

$$\mathcal{N}(\mathbf{x}_w^{(i)}, \omega_m) = \{\mathbf{x} | \mathbf{x} \in \mathcal{N}(\mathbf{x}_w^{(i)}) \wedge \mathbf{x} \in \omega_m\} \quad (2-28)$$

O valores *R-value* de uma classe ω_w em relação à todas as demais ($R(\omega_w)$) e de todo o *dataset* ($R(\mathcal{U})$) são definidos pelas Equações 2-29 e 2-30, respectivamente:

$$R(\omega_w) = \frac{1}{|\omega_w|} \sum_{i=1}^{|\omega_w|} v(|\mathcal{N}(\mathbf{x}_w^{(i)}, \mathcal{U} - \mathcal{D}(\omega_w))| - \theta) \quad (2-29)$$

$$R(\mathcal{U}) = \frac{1}{|\mathcal{U}|} \sum_{w=1}^W \sum_{i=1}^{|\omega_w|} v(|\mathcal{N}(\mathbf{x}_w^{(i)}, \mathcal{U} - \mathcal{D}(\omega_w))| - \theta) \quad (2-30)$$

2.6 Complexidade de Kolmogorov

A Teoria da Complexidade de Kolmogorov foi desenvolvida pelos matemáticos Solomonoff [151], Kolmogorov [80] e Chaitin [27], de forma independente nos anos 60, como uma teoria algorítmica da aleatoriedade. É considerada por [21] uma construção profunda e sofisticada para quantificar a informação contida em objetos individuais. Essa Teoria de Kolmogorov tem suas raízes na Teoria da Probabilidade, Teoria da Informação e conceitos filosóficos sobre a aleatoriedade, consolidando suas proposições na Teoria dos Algoritmos [89]. Diferentemente da Teoria Clássica da Informação de Shannon [144], que postulou que a complexidade de um objeto depende de sua distribuição de probabilidade [40], a Teoria de Kolmogorov discute um conceito moderno de aleatoriedade, pelo qual se entende a complexidade intrínseca de um objeto [21]. Nesse conceito de aleatoriedade pontual, de acordo com [11], a complexidade de Kolmogorov pode ser definida como

uma quantificação objetiva e absoluta da informação contida em uma *string*. No contexto dessa afirmação, *string* é uma cadeia binária finita, cujo conjunto de todas as strings é denotado por $\{0, 1\}^*$. Isso nos leva ao entendimento de informação *absoluta* contida em objetos *individuais*, em contraste com a Teoria Clássica da Informação, que lida com a informação *média* para representação de objetos produzidos por uma fonte aleatória [11].

2.6.1 Bases da Complexidade de Kolmogorov

Para preencher as lacunas deixadas pelas Teorias da Probabilidade e da Informação, [89] discutem a contribuição da complexidade de Kolmogorov. Segundo os autores, a Teoria da Probabilidade não é capaz de expressar a aleatoriedade de uma sequência individual. Ela apenas expressa expectativas de propriedades de resultados em processos aleatórios. Portanto, não nos auxilia no entendimento do evento depois que este acontece. Esse problema encontrou solução na combinação dos conceitos de computabilidade e estatística, realizada por Kolmogorov, para expressar a complexidade de um objeto finito. Na Teoria Clássica da Informação, a quantidade de informação é atribuída a um determinado conjunto de símbolos possíveis de uma mensagem, considerando que todos os símbolos são equiprováveis. Na prática, essa quantificação da informação é expressa em número médio de bits necessários para codificar cada símbolo. Entretanto, essa abordagem não considera o número exato de bits demandados para transmissão de qualquer mensagem individual [89]. A grande vantagem da complexidade de Kolmogorov, frente a essa teoria, é o fato de não ser preciso obter, a partir de amostragens de uma fonte geradora, a probabilidade de ocorrência dos eventos para entender um objeto; basta observar o objeto em si [33].

Sob uma perspectiva complementar, [21] discutem as origens da complexidade de Kolmogorov (Figura 2.7) destacando:

- Aleatoriedade: uma das principais lacunas deixadas pela Teoria da Probabilidade é a distinção prática entre uma sequência regular e uma sequência ale-

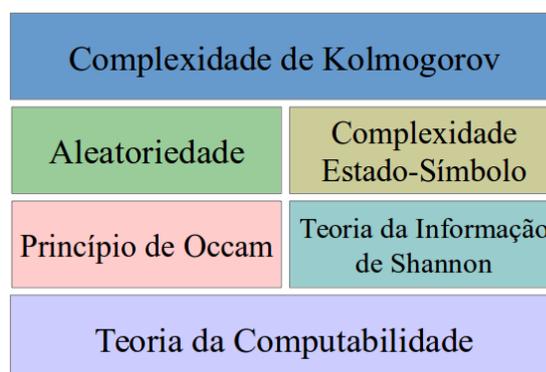


Figura 2.7: Bases da complexidade de Kolmogorov [21]

atória. Em [89], os autores apresentam um exemplo em que é proposto 100 lançamentos consecutivos de uma moeda honesta. No caso da ocorrência de uma sequência de 100 caras, acredita-se que a moeda não é honesta. Entretanto, os fundamentos da probabilidade nos assegura que qualquer sequência é equiprovável. Portanto, a probabilidade de ocorrência de uma sequência totalmente regular, como no exemplo, é a mesma de uma sequência aleatória, ou seja, $\frac{1}{2}^{100}$;

- Complexidade Estado-símbolo: durante os anos de 1950, dois pesquisadores da Bell Labs, Edward F. Moore e Claude E. Shannon, se dedicaram a simplificar máquinas de Turing com o objetivo de aplicar em projetos de calculadoras [109]. Desse trabalho, culminou uma importante contribuição de Shannon, demonstrando que eram necessários apenas dois símbolos para construir uma máquina universal de Turing, fornecendo estados suficientes [18]. De acordo com [101], Shannon definiu o problema da complexidade descritiva, que se dedica a encontrar o número necessário de estados e símbolos para se construir máquinas universais. Assim, reduzindo o número de estados ou de símbolos para dois, o produto estado-símbolo permanece estável, sugerindo uma medida de complexidade para descrição dos algoritmos no contexto do formalismo da Máquina de Turing [89];
- Princípio de Occam: conhecido como princípio da parcimônia e simplicidade, essa premissa é atribuída ao filósofo William de Occam, para sintetizar a ideia de que a explicação mais simples de um fenômeno é a melhor [40]. Portanto, dentre as diversas hipóteses construídas para explicar algo, desde que estas sejam consistentes com os fatos, deve-se optar pela mais simples [21]. Seguindo esse princípio, segundo [40], Solomonoff e Chaitin argumentam que é possível construir uma generalização universalmente boa por meio de uma combinação ponderada de todos os programas que explicam os dados, observando suas respectivas saídas;
- Teoria Clássica da Informação de Shannon: como sugerido por [40], a sintonia entre a entropia de Shannon (H) e complexidade de Kolmogorov ocorre de forma perfeita quando se considera que elas são aproximadamente iguais, quando a sequência for emitida a partir de uma distribuição que tenha entropia H;
- Teoria da Computabilidade: A construção da Teoria de Kolmogorov é realizada utilizando uma máquina de Turing. Isso garante que essa medida de complexidade seja objetiva, uma vez que o cálculo é independente da máquina escolhida para realizá-lo [21].

2.6.2 Complexidades $K(x)$ e $K(x|y)$

Expressa como $K(x)$, a complexidade de Kolmogorov, também conhecida como *entropia algorítmica* [11], é definida como o comprimento do menor programa executado em uma máquina universal que computa a string binária x [20]. Dessa forma, $K(x)$ representa a quantidade mínima de informação necessária para qualquer processo gerador x [11]. Em outra definição, apresentada por [33], K é dada como uma função unária que mapeia strings para inteiros e é implicitamente baseado em uma máquina de Turing, representada pela função Φ . O autor discute que essa definição Φ representa uma Máquina de Turing com regras particulares e uma função parcial que é definida por todos os programas que executam e param. Sendo Φ a transformação que mapeia uma string x em uma saída o , então $K_{\Phi}(x)$ representa o comprimento em bits do menor programa $|p|$, dentre todos os programas válidos de modo que $\Phi(p) = x$. Assim K pode ser interpretado como a menor quantidade de informação necessária para recriar um objeto [33]. Em [40], a complexidade de Kolmogorov $K_{\mathcal{M}}(x)$ de uma string x em relação à uma máquina universal \mathcal{M} é definido como

$$K_{\mathcal{M}}(x) = \min_{p:\mathcal{M}(p)=x} |p|, \quad (2-31)$$

onde $|p|$ denota o comprimento do programa p , $\mathcal{M}(p)$ é a saída do computador \mathcal{M} quando executado o programa p .

A complexidade condicional de Kolmogorov, expressa como $K(x|y)$, ou seja, a complexidade K de x em relação a y , é definida como o menor programa que computa x se y é fornecido como uma entrada auxiliar [88]. O princípio é que se y carrega uma quantidade considerável de informação acerca de x , então $K(x|y) \ll K(x)$. Entretanto, se x e y são completamente independentes, então $K(x|y) \approx K(x)$ [33]. Considerando uma máquina \mathcal{M} que computa a string x a partir de y , utilizando um programa p , [21] afirmam que \mathcal{M} interpreta p como uma descrição de x na presença da informação lateral y . Dessa forma, em [40], assumindo que o computador tem conhecimento *a priori* do tamanho de x , denotado por $|x|$, sendo a informação lateral y , a complexidade condicional é definida como

$$K_{\mathcal{M}}(x|x) = \min_{p:\mathcal{M}(p,|x|)=x} |p|. \quad (2-32)$$

Segundo [105], considerando que $K(x|y) \ll K(x)$, diz-se que y contém informação relevante sobre x e a informação algorítmica de x contida em y é definida por

$$I(y : x) = K(x) - K(x|y), \quad (2-33)$$

com $K(x|x) = 0$ e $I(x : x) = K(x)$. A complexidade incondicional de Kolmogorov da string

x , definida anteriormente como $K(x)$, também pode ser expressa na forma $K(x|\xi)$, onde ξ significa uma string vazia [11].

2.6.3 Universalidade de K e Aleatoriedade

Deve-se destacar, segundo [33], a característica da universalidade de K , pela qual se garante que todas suas variações são, de alguma forma, equivalentes. Quaisquer dois diferentes $K(x)$, obtidos por duas máquinas de Turing, nunca irão se diferenciar além de uma constante aditiva fixa, ou seja, o tamanho do pré-programa que permite que um computador imite o outro [40]. Essa constante depende exclusivamente da máquina utilizada e não da entrada. Portanto, essa característica endossa a ideia de que a medida teórica K pode ser utilizada como medida absoluta da informação contida em um determinado objeto [33]. Segundo [40], para sequências longas, de alta complexidade, essa constante é insignificante, fazendo com que essa descrição mínima atue como um código universal uniformemente bom para todas as distribuições de probabilidade. O autor ainda define a universalidade de Kolmogorov considerando que se \mathcal{M} é um computador universal, para qualquer outro computador \mathcal{A} existe uma constante $c_{\mathcal{A}}$ de modo que

$$K_{\mathcal{M}}(x) \leq K_{\mathcal{A}}(x) + c_{\mathcal{A}}. \quad (2-34)$$

Na definição acima, a constante $c_{\mathcal{A}}$ pode ser muito grande se o computador \mathcal{A} for dotado de muitas funções embutidas no sistema, ao passo que a máquina \mathcal{M} é um simples microprocessador. Essa constante $c_{\mathcal{A}}$ é equivalente ao programa de simulação que contém os detalhes das implementações das funções do computador \mathcal{A} . Como o tamanho $c_{\mathcal{A}}$ desse programa de simulação independe do comprimento da string x a ser computada, ele pode ser negligenciado para x suficientemente longos. Portanto a complexidade de Kolmogorov pode ser discutida ignorando essas constantes [40].

Segundo [57], a principal motivação do trabalho de Kolmogorov era formalizar o conceito de sequência aleatória. Ao contrário do que acredita o senso comum, uma sequência é dita aleatória não apenas por não ser possível encontrar padrões. Uma string x é aleatória se $K(x) > |x| - c$, onde $|x|$ é o comprimento da string x e c é uma constante fixa, que depende exclusivamente da máquina utilizada e não da entrada [33]. Então, se $K(x) > |x| - c$, x é c -incompressível para algum $c > 0$ [20]. Em [57], essa definição é apresentada na forma de: se x pertence a um conjunto \mathcal{L} , de elementos de baixa complexidade K , a complexidade $K(x)$ não pode ser muito maior que $\log_2 |\mathcal{L}|$. Esse limite superior permite definir x como aleatória em \mathcal{L} se $K(x)$ se aproxima de $\log_2 |\mathcal{L}|$. Em síntese, a complexidade descritiva de uma sequência binária verdadeiramente aleatória é tão longa quanto a própria sequência [40].

2.6.4 Aleatoriedade e Informação

Conceitua-se *entropia algorítmica*, desenvolvida dentro da Teoria Algorítmica da Informação, como uma especialização da entropia de Shannon. Na Teoria da Informação de Shannon - ou simplesmente Teoria da Informação - voltada para comunicação de dados, a quantidade de informação é entendida como o tamanho da representação das mensagens em bits [21]. Para Shannon, a entropia de uma informação pode ser quantificada ao observar somente a probabilidade de ocorrência dos símbolos, desprezando a ordem dessa ocorrência. Na complexidade de Kolmogorov, a informação é quantificada de acordo com a complexidade dos objetos em si, ou seja, sua aleatoriedade. Enquanto a segunda definição é considerada uma teoria sofisticada e moderna, por se referir a um conceito pontual da aleatoriedade de objetos individuais, a primeira trata a aleatoriedade como uma aproximação [21]. Por ser uma definição abstrata e representar um limite teórico, é impossível desenvolver, na prática, um método de compressão de propósito geral que atinja esse limite sem promover degradação dos dados ao serem reconstruídos.

Dentro do conceito de entropia da Teoria da Informação, segundo [160], quanto mais uma imagem é aleatória, mais uniforme é a distribuição de probabilidade das ocorrências de cada intensidade de cor. Portanto, maior a incerteza e, conseqüentemente, mais bits por pixel são necessários para representação dos dados. Isso significa que, dentro dessa teoria, a aleatoriedade implica mais informação, no sentido de quantitativo, presente no valor de cada pixel. De acordo com a complexidade de Kolmogorov, quanto mais aleatória uma imagem, maior sua complexidade e menor a previsibilidade dos pixels para serem representados em sequências comprimidas. Portanto, segundo Shannon, conclui-se que imagens aleatórias necessitam de mais quantidade de dados para sua representação. No entanto, segundo Kolmogorov, essas imagens estabelecem mais obstáculos para extração de características e, conseqüentemente, reconhecimento dos padrões devido à imprevisibilidade da vizinhança de pixels.

Em se tratando de imagens capturadas no mundo real, a emersão de características ocorre de acordo com o nível de abstração e a aplicação desejados. Conforme o nível de abstração aplicado à cena, mais aleatoriedade estará presente na imagem resultante. Isso não significa que a imagem é aleatória, mas que ela possui mais complexidade, segundo Kolmogorov, em relação à mesma imagem capturada em outro nível de abstração, com menos detalhes. Em problemas clássicos de reconhecimento de padrões, é de comum acordo na literatura que a aleatoriedade deve ser contornada com o uso de métodos de extração e seleção de características. Assim, a aleatoriedade não é considerada um elemento fundamental para classificação. Em problemas clássicos, o processo de tomada de decisão é efetuado de acordo com características já estruturadas e validadas previamente, por especialistas ou por métodos que busquem pela similaridade intraclasse. Vale ressaltar ainda que a aleatoriedade é:

- *Intratável* no contexto da compressão de dados pois, dado um objeto aleatório, sua descrição comprimida será maior que sua representação original;
- *Insignificante* no contexto do reconhecimento de padrões pois podem representar ruídos ou atributos irrelevantes, que não descrevem a relação entre objetos de uma mesma classe.

2.7 Similaridade Baseada em Compressão de Dados

Segundo o conceito fundamental da complexidade de Kolmogorov, de decisão pela descrição mais curta, o programa p deve explorar quaisquer redundâncias para computar as entradas da máquina [11]. O problema é descrito como a codificação eficiente de uma fonte aleatória em um número mínimo de bits, de forma que esta possa ser reconstruída, garantindo a confiabilidade da descrição [33]. A complexidade K é uma medida teórica e, portanto, incomputável [114]. Portanto, os métodos de aprendizagem de máquina, usualmente, substituem os termos das complexidades incondicional $K(\cdot)$ e condicional $K(\cdot|\cdot)$ de Kolmogorov por aproximações de descrições obtidas por compressores do mundo real [34]. Essas aproximações estimam os limites superiores de K [32], para proposição de métricas de similaridade como em [11, 88, 34]. Entretanto, apenas nos últimos anos as pesquisas têm discutido que as taxas de compressão significam conhecimento da informação estatística [33].

2.7.1 Distância da Informação

Uma métrica de similaridade calcula a distância entre dois objetos individuais. Quanto menor seu valor, mais semelhantes esses objetos são. As métricas de similaridade baseadas em compressão são fundamentadas no conceito da complexidade de Kolmogorov $K(x)$ e na complexidade condicional de Kolmogorov $K(x|y)$ [39]. A primeira proposta de uma métrica de similaridade, baseada no conteúdo da informação de objetos individuais e finitos de Kolmogorov, foi proposta por [11], denominada Distância da Informação denotada por $E(x,y)$. Os autores a definem formalmente como uma métrica que satisfaz todas as propriedades desejadas (não-negatividade, reflexividade, simetria, e desigualdade triangular [50]), e é definida como

$$E(x,y) = \max\{K(x|y), K(y|x)\}. \quad (2-35)$$

Sendo uma aproximação de K , [88] afirmam que $E(x,y)$ é semi-computável até um limite superior,. Combinando a execução de todos os programas p que computam x a partir de y , e vice-versa, podem ser encontradas melhores soluções de modo que $|p| = E(x,y)$, desprezando o tempo de computação. Entretanto, E não é capaz de fornecer

uma métrica de similitude invariante aos tamanhos dos objetos avaliados pois não é normalizada. Um par de objetos muito grandes, que se diferem em poucas partes, possuem distância muito similar em relação a outro par de objetos pequenos, que se diferenciam pela mesma quantidade de dados [39]. Considerando esse exemplo, o primeiro caso deveria ser considerado como os objetos mais próximos, pois a proposta de uma métrica de similaridade de informação é baseada na quantidade de informação mútua e não na diferença entre os objetos.

2.7.2 Distância de Compressão

Em [34], é proposta uma distância de compressão baseada no conceito de compressor normal. Em razão da incomputabilidade de K , os termos de E são substituídos, na prática, por compressores C , obtendo o número de bits da versão comprimida de x quando y é dado como entrada adicional para o compressor [132]. Segundo [34], um compressor C aproxima-se de $E(x, y)$ por $E_C(x, y)$ definido como

$$E_C(x, y) = C(xy) - \min\{C(x), C(y)\}, \quad (2-36)$$

onde $C(xy)$ denota o tamanho comprimido da concatenação de x e y ; $C(x)$ e $C(y)$ denotam o tamanho da compressão de x e y , respectivamente. A escolha da compressão conjunta $C(xy)$ foi realizada em razão da grande maioria dos compressores conhecidos não oferecerem suporte nativo para a compressão condicional $C(x|y)$ [131].

Segundo [32], a partir de uma versão comprimida de uma string x por um compressor C , o algoritmo de descompressão é projetado para ser o mais rápido possível, ou seja, produzir x a partir de $C(x)$ com o mínimo de esforço. Assim, pode-se argumentar que $C(x)$ compreende em uma grande e significativa porção do programa binário hipotético que, executado em uma máquina de Turing, produz x como saída. Portanto, o comprimento de $C(x)$ é um limite superior para o comprimento desse programa, pois as saídas dos compressores reais são maiores que o mínimo teórico de K [32].

Um compressor C é dito normal se satisfaz as propriedades de idempotência, monotonicidade, simetria e distributividade. As igualdades e desigualdades das propriedades são consideradas verdadeiras até a adição de um termo de complexidade $O(\log n)$, onde n é o comprimento máximo em bits de um elemento do conjunto de strings [34]:

1. Idempotência: $C(xx) = C(x)$, $C(\xi) = 0$, onde ξ é uma string vazia;
2. Monotonicidade: $C(xy) \geq C(x)$;
3. Simetria: $C(xy) = C(yx)$;
4. Distributividade: $C(xy) + C(z) \leq C(xz) + C(yz)$.

Especificamente em relação à propriedade de simetria, [34] discutem que compressores baseados no Lempel-Ziv (*gzip*) e no *Prediction by Partial Matching* (PPM) (*PPMZ*), não são precisamente simétricos. Eles relacionam a causa à propriedade do fluxo de leitura que depende da ordem em que os arquivos são concatenados. Apesar da possibilidade do processo causar alguma imprecisão na simetria, ela desaparece assintoticamente com o comprimento de x e y . Em outra família de compressores baseados na codificação por bloco (*bzip2*), observa-se que são, em grande parte, simétricos, com experimentos reais não demonstrando nenhum desvio da simetria.

2.7.3 Distância Normalizada da Informação

A Distância Normalizada da Informação (NID - *Normalized Information Distance*) foi proposta por [88] como uma versão normalizada de $E(x, y)$ [34]. A limitação da métrica E é considerar a distância absoluta entre os objetos, e não a relativa, fazendo com que objetos de tamanhos diferentes, por mais que sejam muitos similares, não sejam agrupados conjuntamente. Citando o exemplo de duas espécies irmãs de bactérias, *H. influenza* formada por 1.856.000 pares de base e a *E. coli* formada por 4.772.000 pares de base, [88] argumentam que utilizando a distância E , *H. influenza* seria classificada como similar a outras espécies irrelevantes. Isso aconteceria em razão da diferença entre os tamanhos de seus DNAs.

Como uma métrica de similaridade independente de qualquer campo de aplicação ou conhecimento prévio [88], a NID é definida por [34] como

$$\text{NID}(x, y) = \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}}. \quad (2-37)$$

A NID não pode ser computada pois é expressa em termos das complexidades não-computáveis de Kolmogorov. Em razão disso, [88] substituíram os termos K por compressores.

2.7.4 Distância Normalizada de Compressão

A Distância Normalizada de Compressão (NCD - *Normalized Compression Distance*) foi proposta por [34]. É uma aproximação da complexidade condicional de Kolmogorov utilizada na NID [39], que é considerada por [157] como uma medida de distorção teórica da informação. Em [88], a lacuna entre a rigorosa prova de otimalidade da NID e a proposta da NCD é tratada substituindo os termos incomputáveis de K pelos comprimentos em bits dos respectivos arquivos comprimidos, utilizando compressores normais. A NCD entre dois objetos x e y é expressa por um valor positivo entre $[0; 1+e]$ [43], representando o quanto são similares. Nesse intervalo, e representa o erro do

processo de compressão [34], quando duas strings compartilham pouca ou nenhuma informação [112] e é definida por

$$\text{NCD}(x, y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}. \quad (2-38)$$

Considerando que $K(yx) \approx K(xy)$ até uma constante, [32] descrevem uma aproximação dentro da própria NID com $K(x|y) \approx K(yx) - K(x)$. Portanto, a compressão condicional $C(x|y)$ derivada da NID foi substituída por $C(xy) - C(x)$ na NCD. Essa substituição pode ser interpretada, no contexto de compressores baseados em fluxo, como o tamanho da compressão de x baseado no dicionário extraído de y [88]. Em [19], os autores definem a compressão $C(x|y)$ como o tamanho resultante quando o compressor é primeiramente treinado em y e, em seguida, utiliza essas informações para comprimir x .

Enquanto a NID é considerada por [162] como uma métrica universal, pois minoriza cada métrica computável, garantindo que seja aplicada com sucesso em qualquer domínio, a NCD mostrou ser uma medida quase universal. Ela minoriza a distância computável de similaridade até um limite de erro, que depende diretamente da qualidade da aproximação do compressor normal de referência para K [34]. Assim, a NCD viola a NID apenas no sentido que $C(x)$ é um limite superior de $K(x)$ [88]. Para o caso de similaridade entre imagens, [112] afirma que essa aproximação é limitada pois não é possível projetar um algoritmo de compressão ótimo sem perda.

Medidas de similaridade de imagens baseadas em compressão

Este capítulo revisa a literatura desde a origem das técnicas de similaridade por compressão de dados até os mais recentes avanços em aplicações de similaridade entre imagens. A Figura 3.1 sumariza o escopo do capítulo, que será detalhadamente discutido ao longo do texto. Os fatores discutidos incluem as diferentes famílias dos algoritmos de compressão e suas particularidades em relação ao tipo de dado processado; influência dos diversos algoritmos combinados em esquemas de compressão de dados e implementados nos compressores comerciais; formas de representação dos dados em formatos de imagens; estratégias de linearização dos dados bidimensionais; entre outros. Discute-se as vantagens da aplicação dessa abordagem em relação às medidas de qualidade de imagem. Por fim, são apresentadas outras medidas de similaridade que foram desenvolvidas para contornar as limitações da NCD e que utilizam a compressão como extrator de semântica.

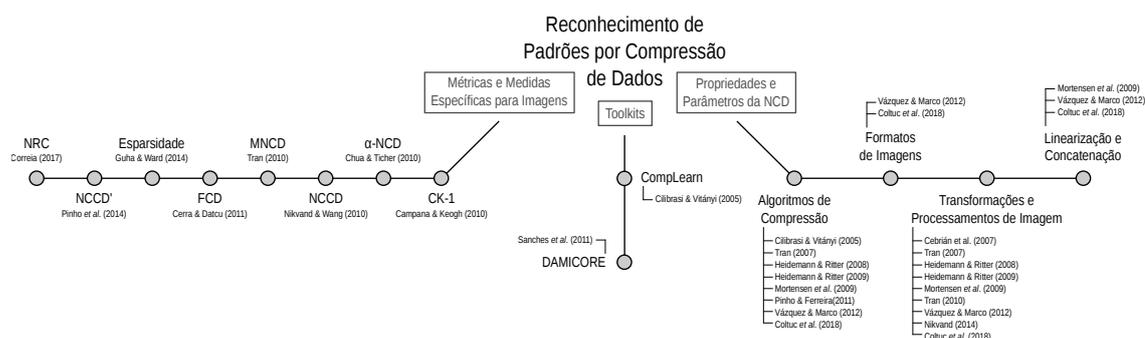


Figura 3.1: Linhas de investigação das técnicas de reconhecimento de padrões em imagens por compressão de dados.

3.1 Compressão como Medida de Similaridade

Segundo [64], a avaliação de similaridade entre pares de imagens tem importante aplicação em sistemas de processamento de informações multimídia como recuperação de imagens, aprimoramento, detecção de cópia, avaliação de qualidade, clusterização

e classificação. Nos últimos anos, técnicas de reconhecimento de padrões utilizando compressão de dados tem ganhado a atenção de pesquisadores. Entretanto, em problemas envolvendo imagens, as aplicações não tem acompanhado o sucesso dos métodos que lidam com dados unidimensionais. Isso se deve a dois principais fatores discutidos em [64]:

1. Dependência direta da disponibilidade de um compressor normal, como definido por [34]; e
2. A aproximação da NID, que substitui os termos da compressão condicional $C(x|y)$ pela compressão conjunta $C(xy)$ da NCD e suas derivações.

Avaliação de similaridade é frequentemente relatada na literatura como uma forma de medir a qualidade entre pares de imagens, onde uma é a versão original (referência) e a outra é degradada por alguma transformação ou processamento [148]. Em razão disso, algumas medidas de qualidade têm sido comparadas com as medidas baseadas em compressão, como: Erro Quadrático Médio (MSE - *Mean Squared Error*) [117, 116], Similaridade Estrutural (SSIM - *Structural Similarity*) [117, 116], Relação Sinal-Ruído (SNR - *Signal-to-Noise Ratio*) [114] e Relação Sinal-Ruído de Pico (PSNR - *Peak Signal-to-Noise Ratio*) [116, 64]. Normalmente, essas medidas são utilizadas em experimentos iniciais para avaliar a idempotência do compressor e a capacidade de discriminação perceptual dos métodos propostos.

Utilizando a mesma estratégia de medidas baseadas na norma, como a distância euclidiana [64], o MSE compara a similaridade das imagens pixel-a-pixel, medindo a distância entre eles. Portanto, o MSE não é considerada uma medida adequada para avaliação de duas imagens transladadas [162]. Outro problema relatado é que imagens com diferentes tipos e níveis de distorção podem ter o mesmo valor de MSE, embora algumas abordagens utilizem a ponderação dos diferentes aspectos do erro, baseada em aspectos psicológicos [116]. A SSIM é fundamentada na hipótese de que cenas são altamente estruturadas e compostas de fortes dependências entre os pixels vizinhos [116]. Considerada uma medida genérica de similaridade, a SSIM é restrita a algumas aplicações. Em transformações geométricas, como escala de fator 2 ou rotações de 90°, a medida não funciona de modo aceitável para comparação de 2 imagens [117]. No caso da PSNR, [139] afirma que essa medida tem uma relação limitada com a forma de percepção dos erros pelo sistema visual humano: valores altos indicam semelhanças entre as imagens mas não garantem a mesma percepção na análise subjetiva por humanos. O MSE e a PSNR são inconsistentes com o sistema visual humano e não refletem com fidelidade a qualidade de imagens [116].

Algoritmos tradicionais de reconhecimento de padrões em imagens são escolhas populares e atuais em problemas de agrupamento e classificação. Entretanto, essas

técnicas são diretamente dependentes da disponibilidade de características significativas, o que requer conhecimento prévio dos dados e da aplicação [64]. Diferentes métodos de extração e seleção de características fornecem diferentes representações estruturadas das imagens. Isso pode ocasionar descrições enviesadas ou incompletas das informações contidas nesses objetos, em razão das relações ocultas presentes nos diversos tipos de dados [34]. Embora, mesmo com essas descrições limitadas, seja possível encontrar similaridades entre os objetos, os métodos tradicionais requerem conhecimento específico e detalhado do problema [34]. Ao contrário, a abordagem de extração de semântica, empregada pelos métodos que utilizam a compressão, é baseada na teoria que o conteúdo visual de uma imagem é codificado na representação primária dos dados [112]. Assim, não é necessária nenhuma etapa anterior para encontrar os atributos discriminantes de uma imagem, ou seja, sua própria descrição algorítmica é capaz de significar o conteúdo de sua informação.

Nas seções seguintes, serão discutidos vários trabalhos que investigaram o comportamento na NCD aplicada a problemas de similaridade de imagens e propuseram novas medidas para extração de semântica de pares de objetos bidimensionais. Entretanto, preliminarmente, é importante salientar que uma extensão direta das métricas, que funcionam bem no domínio unidimensional discreto, não é adequada para aplicação em problemas de imagens [64].

3.2 Toolkits: CompLearn e DAMICORE

No trabalho de [34] foi desenvolvida a ferramenta CompLearn, que encapsula em um pacote todo o processo de compressão de dados para clusterização. Dado um conjunto de objetos, são calculadas as distâncias entre todos os pares do conjunto. Empregando a NCD como métrica, os autores reportam distâncias bem próximas nos experimentos, entre 0,85 e 1,1. Esses valores, organizados em uma matriz simétrica de distâncias, representam a informação compartilhada entre os objetos e, portanto, necessita de uma descrição que favoreça o entendimento das relações. Assim, a hierarquia dos clusters é construída utilizando dendrogramas, a partir de árvores binárias com o método do quarteto. No problema de clusterização de imagens, os autores realizaram experimentos com apenas 10 amostras de 3 classes de dígitos (4, 5 e 6) obtendo 0,93 de acurácia. Ao considerar todos os dígitos, eles relatam baixa acurácia no agrupamento. Para contornar esse problema, eles aplicam a NCD como técnica de extração de características para treinar uma SVM, discutido com mais detalhes na Seção 3.6.

Em [140] os autores propõem uma abordagem de mineração de dados para clusterização de códigos-fonte e representações simbólicas, aptos a serem implementados conjuntamente em arquiteturas de hardware. A motivação para agrupar *kernels* de soft-

ware similares, segundo os autores, permite economizar recursos computacionais, reduzindo o custo do hardware e mantendo o desempenho similar. Para isso, eles combinam técnicas utilizadas em outras áreas da ciência - NCD da Teoria da Informação, *Neighbor Joining* (NJ) da Filogenética e o *Fast Newman* de Redes Complexas - e testam uma nova ferramenta denominada *Data-Mining of Code Repositories* (DAMICORE). Nos experimentos foram utilizados 11 códigos-fonte de algoritmos de processamento de imagens, sinais digitais, áudio, ordenação, entre outros. Os resultados reportados mostram, de acordo com os clusters recomendados pelo DAMICORE, redução de até 7,45% nas tabelas de indexação de entrada e saída (LUT - *LookUp Table*) e 35,39% nos registradores.

Embora o CompLearn e o DAMICORE tenham sido concebidos com o mesmo objetivo, isto é, explorar as ferramentas de compressão de propósito geral para reconhecimento de padrões, esses *toolkits* possuem diferenças na etapa de clusterização. Isso se deve em razão da NCD, por si só, não ser capaz de definir agrupamentos com grandes quantidades de objetos [34]. Esses trabalhos promoveram significativos avanços nas pesquisas em agrupamento de dados utilizando a compressão. A contribuição desses trabalhos é dupla: metodologia empregada para a descoberta de conhecimento e disponibilização do pacote compilado (CompLearn¹) ou o código-fonte (DAMICORE²) para a comunidade científica. Assim, pode ser observado que muitos autores citados nesse capítulo investigaram as vantagens e limitações da NCD em seus respectivos trabalhos utilizando esses *toolkits*, sem a necessidade de implementações diretas dos métodos.

3.3 Medida Quantitativa de Agrupamentos G

No trabalho de [8], é proposta uma medida quantitativa para avaliação da qualidade dos modelos de agrupamento hierárquico de dados. A medida G calcula a capacidade de generalização de árvores filogenéticas e pode ser aplicada, por exemplo, nos modelos gerados pelo DAMICORE.

Essa medida, além de ser utilizada para comparar agrupamentos, é equivalente à acurácia da clusterização dos dados. Segundo o autor, a partir de um agrupamento hierárquico, pode ser realizada uma análise quantitativa, levando em conta o número de congruências em associações par-a-par de elementos de um dendrograma. Considerando que o modelo de agrupamento gerado pelo DAMICORE é representado por uma árvore filogenética parcimoniosa, de evolução mínima, a medida G assume que cada amostra é ligada a outra amostra com vizinhança mais próxima. Assim, o autor propõe que o modelo seja decomposto em unidades mínimas de agrupamento, que recebem o valor 1 para pares

¹Disponível em <http://complearn.org>

²Disponível em <https://github.com/brunokim/damicore-python>

de objetos com mesmo rótulo e 0 caso contrário. Portanto, a medida G é definida por

$$G = \frac{\sum_{i=0}^{|\mathcal{U}|-2} c(x_i, x_{i+1})}{|\mathcal{U}| - W} \quad (3-1)$$

onde $c(x_i, x_{i+1}) = 1$ para pares de objetos com rótulos coincidentes e $c(x_i, x_{i+1}) = 0$ para pares com rótulos discordantes, entre os objetos x_i da árvore e seu sucessor mais próximo x_{i+1} , considerando a ordem de evolução dos objetos na árvore filogenética parcimoniosa; $|\mathcal{U}|$ é o número de objetos e W é o número de classes do problema. É condição que $|\mathcal{U}| > W$ pois para $|\mathcal{U}| = W$ qualquer isomorfismo da árvore filogenética é solução do problema (caso trivial).

3.4 Propriedades e Parâmetros da NCD

Embora seja difícil discutir separadamente os aspectos envolvidos no desempenho da NCD, propomos uma abordagem *bottom-up* nesta seção, segmentando as principais propriedades e parâmetros investigados na literatura nos últimos anos. Isolando essas variáveis, entende-se de forma mais clara e objetiva cada componente envolvido e seu impacto no desempenho da métrica. Eventualmente, informações em duplicidade podem ser encontradas ao longo da discussão, propositalmente incluídas para contextualizar as condições experimentais nas quais foram observadas ou sintetizar conclusões. O primeiro fator que deve ser analisado é a influência da técnica de compressão utilizada, principal etapa do cálculo da NCD. Em seguida, o quanto a escolha do formato de imagens impacta nos resultados. Também é discutido quais tipos de transformações geométricas e de processamento de imagem são detectadas pela NCD. Por fim, os métodos de linearização e concatenação de imagens que mais preservam a espacialidade da informação.

3.4.1 Algoritmos de Compressão

De acordo com [131], a propriedade de idempotência é a mais crucial na denominação de um compressor normal. Entretanto, segundo [34], se observadas as demais propriedades³, um algoritmo (ou um conjunto de algoritmos do esquema de compressão) está apto a coletar conhecimento à medida que avança no processo de compressão [131]. Quanto mais um compressor do mundo real é próximo de um compressor normal, melhor é a aproximação da complexidade de Kolmogorov [64], ou seja, a entropia algorítmica é obtida com mais confiança. Os algoritmos dedicados para imagens, apesar de

³Monotonicidade, simetria e distributividade.

obterem ótimas taxas de compressão, não preenchem as características necessárias para serem considerados compressores normais [130].

A maioria dos métodos de compressão de imagens considerados estado da arte, segundo [130], destroem a correlação entre os dados com as transformadas de domínio. Esses algoritmos trabalham com um modelo essencialmente estático, ao contrário dos métodos que constroem o conhecimento acerca dos dados à medida que avança no processo de compressão. Como essas transformadas eliminam a dependência dos dados, na etapa de codificação de entropia os símbolos são codificados assumindo serem de uma fonte independente [130].

Um amplo estudo que explora as melhores condições para aplicação da NCD em imagens coloridas é realizado em [162]. Os autores utilizam o *toolkit* CompLearn e investigam o tipo de compressor e o formato de imagem ideais e a resistência à transformações espaciais. Os compressores utilizados foram os das famílias LZ (*gzip*, ZIP e GZ), compressão por bloco (*bzip2*), PPM (*7z*, *paq8px* e WinRAR) e os de aplicação específica em imagens, baseados em transformadas, JPEG e JPEG2000. Os autores afirmam que, embora não existam muitos algoritmos de compressão disponíveis, a diversidade de esquemas de compressão que incorporam esses algoritmos é maior. Assim, compressores baseados no mesmo algoritmo podem alcançar resultados diferentes. Os autores destacam que ao longo dos experimentos os compressores tiveram comportamentos bem irregulares quando aplicados em problemas de imagens.

Ainda em [162], é discutido que não necessariamente os melhores compressores utilizados para cálculo da NCD são os que atingem as maiores taxa de compressão, pois isso não tem relação com a eficiência da aproximação dada pela NCD. Entretanto, um fator que influencia diretamente os resultados é o tamanho dos objetos avaliados. No caso dos compressores da família LZ, que favorecem a simetria, quanto maior o arquivo, melhor a taxa de compressão, pois o método tende a aproximar-se do limite estabelecido pela entropia de Shannon [139]. No caso dos compressores baseados em bloco, os arquivos maiores tornam o processo de comparação menos efetivo, pois são baseados no tamanho da janela. Já os compressores da família PPM não são precisamente simétricos, embora, para strings longas, tendem a se aproximar da simetria [162]. Segundo [48], os algoritmos dessa família utilizam modelos estatísticos baseados em árvores, são lentos e consomem muita memória. Os compressores *paq*, concorrentes da família PPM, utilizam redes neurais e são igualmente lentos.

Nos experimentos de [162], os autores avaliaram a similaridade entre uma imagem e sua cópia, investigando quais compressores apresentavam a propriedade de idempotência. Os resultados mostram que a maioria dos compressores não alcançaram esse objetivo. Para a família LZ, os resultados reportados mostram valores mínimos obtidos pela NCD próximos a 0 para os arquivos no formato JPEG e JPEG2000. Entretanto, os

valores médio e máximo ficaram em torno de 0,5 e 1, respectivamente. Utilizando o *bzip2*, os valores mínimos ficaram bem próximos a 0,25 e os máximos próximos a 1, esse último valor coincidindo com concatenações maiores que 900KB; seus melhores resultados foram obtidos com os formatos BMP, TIFF e PPM (*Portable Pixel Map*). As menores distâncias foram reportadas pelos compressores da família PPM, com destaque para o *7z* e o *paq8px*. Esse resultado é atribuído ao fato de que eles possuem suporte nativo à compressão de imagens como JPEG e PPM. De modo geral, os formatos de imagens sem nenhum esquema de compressão obtêm os melhores resultados com esses algoritmos. Observado esse desempenho na propriedade de idempotência, para os testes seguintes foram selecionados apenas os compressores com melhores resultados: *bzip2*, WinRAR, *7z* e *paq8px*. Os autores obtiveram bons resultados com o *bzip2* e *paq8px* em testes de clusterização. Embora na avaliação geral de desempenho tenha sido observado que o *paq8px* é o melhor, os autores afirmam que seu tempo de execução elevado inviabiliza sua aplicação. Entretanto, mesmo não explicando por que o *7z* não foi incluído nessa avaliação, visto que foram observados bons resultados anteriores, afirmam que ele tem melhor custo-benefício. Os autores classificam ao final dos experimentos a família PPM com bom desempenho, a compressão por bloco com médio e a LZ com desempenho ruim.

Nos trabalhos de [10, 34, 70, 71, 140] foram utilizados compressores comerciais que encapsulam diversos métodos de compressão sem perda. Esses compressores como o ZIP, *gzip* e *bzip2* não são considerados por [71] como ideais para aproximação das informações mútuas, além de não serem as melhores soluções de compressão. Os autores ainda sugerem que os resultados da classificação podem ser aprimorados se utilizados algoritmos especializados em cada tipo de dado. Diante disso, constata-se que para explorar todos os benefícios da compressão na classificação de dados, os métodos de propósito geral não podem ser considerados ideais isoladamente. Embora utilizados nos primeiros passos dessa metodologia, pois reconhecem bem padrões unidimensionais, desprezam a correlação espacial em imagens e não são capazes de consolidar a contribuição da compressão de dados no reconhecimento de padrões. Sugerimos que esses métodos sejam combinados com outras técnicas que compensem essa limitação. Em contraponto, em [130] os resultados indicam que o *bzip2* e o *gzip* funcionam bem para a similaridade de faces, com uma discreta vantagem para o *gzip*, baseado na família LZ. Isso sugere que os métodos estendidos dessa família são potencialmente candidatos à aproximação da complexidade de Kolmogorov.

Em [130], os autores argumentam que apesar dessa abordagem ter interessante e desafiadora aplicação em problemas de imagens, o desempenho da compressão dos dados é o maior obstáculo. Segundo eles, métodos de compressão específicos para imagens, que atingem alto desempenho, não são normais e os enquadrados nessa categoria não conseguem resultados satisfatórios. Para compensar essa limitação, é proposto um

algoritmo baseado em Modelos de Contexto Finito (FCM - *Finite-Context Model*) , que atua diretamente no domínio de intensidade da imagem. As imagens de entrada são quantizadas para 4 níveis de cinza, devido à limitação dos FCMs em lidar com grandes alfabetos e ao consumo dos recursos de memória. O codificador proposto é semelhante à compressão do JBIG, com duas principais diferenças: em vez de segmentar a imagem em planos de bits, os pixels são considerados uma única unidade; e são utilizados múltiplos contextos, combinados com um modelo de mistura, baseados nos respectivos desempenhos anteriores desses contextos.

Para comparação dos resultados de [130], a NCD foi aplicada às imagens em escala de cinza, sem quantização, com outros 3 algoritmos de compressão de imagens sem perda JPEG2000, JPEG-LS e JBIG e 4 algoritmos de propósito geral *gzip*, *bzip2*, LZMA e PPMd. A base de dados utilizada é composta de 10 imagens de face, em diferentes oclusões, de 40 indivíduos. Essas imagens foram divididas em 2 grupos: imagens de referência, compostas pelas primeiras imagens de cada indivíduo, e imagens de teste, contendo as demais. Após calculada a NCD entre as imagens de referência e as imagens de teste, os autores escolheram as 9 imagens mais próximas e observaram as ocorrências verdadeiras. O método proposto obteve taxas ligeiramente melhores que o *bzip2* e o *gzip*. Entretanto, os autores não discutem o fato que, para grande parte dos indivíduos, o método tem resultados iguais ou muito próximos (com diferença oscilando em 1 acerto para mais ou para menos) em relação aos demais algoritmos de propósito geral. A surpresa está nos resultados ruins do PPMd em relação aos demais, diferente do sugerido em [162] que reportam melhores resultados alcançados pelo compressor *7z*. Apesar da alta degradação dos dados ocasionada pela quantização, os resultados mostram que com poucos níveis de cinza é possível manter as características das imagens. Isso demonstra que o método de compressão proposto é tanto capaz de extrair semântica, a partir de uma fonte de dados simplificada, quanto limitado por não ser capaz de processar alfabetos maiores sem perder o desempenho. Acreditamos que se o FCM fosse capaz de lidar com alfabetos maiores em um tempo de execução razoável, os resultados poderiam ser aprimorados. Destacamos, ainda, a necessidade do compressor utilizado na aproximação da NID ser normal, que não é discutida pelos autores no método proposto.

3.4.2 Formatos de Imagens

Em problemas de similaridade de imagens, a informação de cor é ativamente utilizada como característica de classificação, mais do que em outras áreas da Visão Computacional. Isso se deve do potencial discriminante da cor no domínio tridimensional, quando comparado ao domínio bidimensional das imagens em escala de cinza [149]. Em contraponto, [71] obtiveram bons resultados ao agrupar imagens em escala de cinza, suge-

rindo uma alta capacidade de abstração da técnica para tornar o problema de agrupamento de imagens independente da cor. Como melhoria dos resultados, os autores lembram que os motores de compressão utilizados, *gzip* e *bzip2*, realizam compressão sem perda e que para classificar imagens similares, variações dos valores dos pixels devem ser toleradas. Assim, apontam que a aplicação da compressão com perda eliminaria ruídos e melhoraria o desempenho.

No trabalho de [162], foram realizados experimentos com diversos compressores nos formatos de imagens PPM, BMP, TIFF, JPEG, JPEG2000 e PNG. O PPM obteve os melhores resultados com todos os compressores em razão de ser um formato livre de compressão. Adicionalmente, esse formato representa a informação de maneira mais simples, com o mínimo de metadados possível no cabeçalho. No geral, os demais formatos são considerados bons para aplicação da NCD, com exceção do JPEG e JPEG2000 que obtiveram desempenho aceitável apenas para o *paq8px*. Os autores sugerem que a NCD é diretamente dependente dos formatos de imagens e dos compressores utilizados. Dessa forma, a extração de semântica é influenciada tanto pela forma de armazenamento dos dados nos arquivos, quanto pelas técnicas utilizadas para compressão das redundâncias. Geralmente, os métodos que utilizam a NCD e suas variações atuam no nível de arquivo, lendo os dados para compressão de forma sequencial. Consequentemente, as imagens são convertidas em dados unidimensionais antes de calcular as distâncias. Dependendo da localização das principais características, uma leitura orientada por colunas pode fornecer melhores resultados que a clássica leitura orientada por linhas [38]. Em [112], os autores investigaram o efeito da linearização em cópias rotacionadas de imagens, que será discutido na Subseção 3.4.4.

3.4.3 Transformações e Processamentos de Imagem

Em [157] a NCD é utilizada em dois tipos de problemas: simulação do experimento de Goldmeier [62] e distinguibilidade de imagens similares no contexto da proteção de direitos autorais utilizando *watermarking*. Os autores relacionam a *Teoria da Informação Estrutural* (SIT - *Structural Information Theory*) de Leeuwenberg [86] com a codificação da informação por uma máquina de Turing para justificar o uso da NCD. A SIT propõe que a imagem é codificada na retina como uma linguagem LOGO em uma representação mais compacta. No experimento original de Goldmeier são apresentados 79 grupos de figuras aos participantes, dos quais uma imagem é chamada de desenho-padrão e as demais são os desenhos de comparação. Em seguida, é solicitado que os participantes escolham os desenhos de comparação mais similares ao desenho-padrão. Na simulação desse experimento com a NCD, os autores utilizam 36 grupos de figuras e 4 algoritmos de compressão - *paq7*, *gzip*, *bzip2* e JPEG - aplicados às figuras no formato *Portable Gray*

Map (PGM) (ASCII). O melhor resultado foi obtido pelo *gzip* com uma taxa de erro de 0,25 e o pior pelo JPEG com erro superior a 0,5, apesar desse último de ter sido projetado para explorar os vieses visuais perceptivos. De modo geral, os experimentos mostraram que a NCD funciona bem para encontrar similaridade nos desenhos que tem partes adicionadas ou removidas, mas falha em variações de forma e estrutura. Para o problema de *watermarking*, os resultados demonstram que a NCD não oferece uma boa predição. Então é sugerido que seja utilizada a diferença pixel-a-pixel d entre as imagens. Assim o termo $C(xy) - \min\{C(x), C(y)\}$ da NCD seria substituído por $C(d)$.

Embora os resultados reportados em [157] demonstrem que a NCD funciona bem para identificar similaridade entre 2 imagens com partes adicionadas ou subtraídas, seu desempenho é ruim em transformações geométricas [38]. No trabalho de [116], o autor acredita que esse resultado se deve, principalmente, à fraca aproximação de $K(x|y)$ usando um compressor normal para comprimir duas imagens concatenadas $C(xy)$. Em [162], os autores reportam melhores resultados com o compressor $7z$, baseado na família PPM, para similaridade de imagens escaladas e rotacionadas.

No trabalho de [23], os autores hipotetizam que em razão da NCD ser baseada em compressores, ruídos inseridos nos dados podem influenciar os resultados. Portanto, foi investigado se em um problema de clusterização a NCD poderia considerar como dissimilares dois arquivos similares corrompidos por ruído. A justificativa é fundamentada na teoria de que um compressor perde sua capacidade de reduzir o volume de dados à medida que ruídos são adicionados ao arquivo original. Nos experimentos foram utilizados arquivos do tipo texto (ASCII), sequências de DNA mitocondrial (mtDNA), áudios (WAV) e imagens de face (GIF). Foram adicionados, progressivamente, níveis de ruído ao segundo arquivo-parâmetro da NCD e observado o comportamento da métrica utilizando o *toolkit* CompLearn. Os resultados mostraram que, para determinados tipos de arquivos, a distorção média entre o par de arquivos aumenta com o ruído adicionado e, a partir de um ponto, começa a decair. Esse comportamento não foi observado somente nas imagens e explica porque em alguns casos a NCD retorna valores maiores que 1. Quando a distorção média do par atinge o pico, a NCD apresenta esse comportamento, ou seja, no momento em que os arquivos se diferenciam ao máximo na média. Os autores ainda discutem que o processo de clusterização não é influenciado significativamente com a adição de ruído. Apesar disso, relatam que as sequências de DNA se degradaram mais rapidamente em comparação com os outros arquivos. Esse resultado é esperado em razão do tamanho reduzido do alfabeto. Entretanto, a conclusão dos autores se torna subjetiva pois nenhuma métrica de análise de agrupamento foi apresentada, além de não utilizarem um maior volume de dados nos experimentos.

Em [158], os autores propõem um estudo do comportamento da NCD em operações de processamento de imagens e transformações geométricas: translação, rotação,

reflexão, desfoque, suavização, nitidez, equalização de histograma, adição de ruído, ocultação de dados, anotação e marca d'água. Os autores afirmam que a NCD é simétrica e transitiva, podendo ser usada como uma medida razoável de similaridade perceptual, porém, apenas no domínio espacial. Os resultados das clusterizações indicam que a NCD tende a agrupar as operações de processamento e transformação em três grupos, de maneira consistente com a percepção humana. O primeiro agrupa as transformações menos sensíveis à percepção como esteganografia e translação. O segundo grupo concentra as operações de média sensibilidade como filtros espaciais e adição de ruídos. O terceiro grupo reúne as operações que são facilmente percebidas pela visão humana como suavização, desfoque e equalização de histograma. Os autores ainda propõem duas modificações na NCD que serão apresentadas e discutidas oportunamente na Seção 3.5.

Em diferentes cenários experimentais, [162] executam testes de avaliação de similaridade entre: uma imagem e sua cópia, duas imagens com o mesmo objeto deslocado no espaço, rotação de objetos, o mesmo objeto com diferentes ângulos de visão e imagens iguais em diferentes escalas. As imagens utilizadas foram obtidas de diversos *datasets* com contexto complexo, cores dominantes e variadas, diferentes tamanhos e proporções e transformadas com escala, translação, rotação e outras transformações. Nos testes de translação, os compressores da família PPM forneceram as menores distâncias para os formatos de imagem sem compressão; o WinRAR apresentou desempenho inferior em relação aos demais. Para avaliar a rotação, os autores realizaram 2 tipos de transformação: pequenas rotações em passos de 1° e grandes rotações em passos de 15°. Apenas rotações múltiplas de 90° mantiveram alguma relação de similaridade, mas com valores distantes de 0 mesmo nos melhores resultados. A solução encontrada foi utilizar o *paq8px* removendo os cabeçalhos que identificavam os arquivos como imagens. Assim, foi possível obter melhorias em pequenas rotações, embora a compressão tenha sido computacionalmente mais intensiva. Na mudança do ângulo de visão, foram utilizados os compressores *bzip2*, *7z* e *paq8px* com os cabeçalhos das imagens também removidos. Os resultados foram similares à rotação, mantendo alguma similaridade nas mudanças iniciais, diminuindo rapidamente para as mudanças seguintes e depois estabilizando. Nas transformações de escala, as imagens foram reduzidas em passos de 5% do seu tamanho original. Os resultados descrevem comportamento semelhante observado nas rotações e mudanças de ângulo de visão, com pequenas variações e remoção dos metadados alcançando resultados coerentes para os compressores *7z* e *paq8px*.

3.4.4 Linearização e Concatenação

Um estudo sobre o efeito de diferentes métodos de linearização é realizado em [112]. Os autores relatam que trabalhos anteriores obtiveram sucesso utilizando

dados unidimensionais, diferentemente para as imagens que possuem correlação espacial dos dados. Assim, justificam que a etapa de transformação de dados bidimensionais em unidimensionais pode ser decisiva nos resultados dos agrupamentos. As técnicas de linearização utilizadas foram: linha-a-linha (*row-major*), coluna-a-coluna (*column-major*), curva de Hilbert-Peano e Auto-descrição de Conteúdo Baseada na Ordem dos Pixels (SCPO - *Self-Describing Context Based Pixel Ordering*) . Essas técnicas foram aplicadas às imagens após as transformações de: translações de 35 pixels para a esquerda e para baixo (o espaço vazio deixado foi preenchido com pixels brancos), rotações em sentido horário de 90°, 180° e 270° e reflexões vertical e horizontal. As linearizações das imagens foram convertidas em arquivos de texto, que consideramos inapropriado visto que os dígitos que representam os níveis de intensidade são interpretados como caracteres individuais da tabela ASCII. Devido à restrição da curva de Hilbert-Peano, que preenche totalmente apenas espaços quadrados, as imagens foram redimensionadas para manterem as mesmas dimensões de largura e altura. Embora tenham discutido superficialmente sobre possíveis distorções, os autores não investigam seus efeitos para similaridade. Devido à aplicação computacionalmente dispendiosa do SCPO, as imagens foram reduzidas para 35% do seu tamanho original, mantendo a proporção.

Os cálculos da NCD foram executados por [112] utilizando o *toolkit* CompLearn com os compressores *bzip2* e JPEG. Os resultados mostram que os métodos de linearização linha-a-linha e coluna-a-coluna não foram eficientes em imagens rotacionadas. Entretanto, produziram valores significativamente mais baixos para a NCD nas imagens transladas. Em contraste, os métodos Hilbert-Peano e SCPO não foram capazes de manter valores baixos nas translações. Acreditamos que esse resultado poderia ser diferente se os espaços vazios deixados pelas translações tivessem sido preenchidos, continuamente, pelos próprios pixels deslocados. Como esperado, as linearizações linha-a-linha para reflexões horizontais e coluna-a-coluna para reflexões verticais, também obtiveram bons resultados. Os valores da NCD para versões rotacionadas aplicando Hilbert-Peano foram relevantes estatisticamente, quando comparados com outros métodos de linearização. Embora os estudos preliminares de [41] tenham demonstrado que o SCPO aumenta as taxas de autocorrelação e compressão para uma única imagem, os resultados reportados por [112] demonstram que esse método não contribui na similaridade entre pares. Os piores resultados foram reportados com a compressão JPEG. Em razão das particularidades de cada método de compressão e dos desempenhos das técnicas de linearização, acreditamos que o estudo pode ser aplicado em *datasets* reais, sem a utilização de versões modificadas das imagens.

No trabalho de [162], os autores exploraram o processo de compressão conjunta da NCD com 3 tipos de concatenação: anexação dos arquivos binários, concatenação das imagens e entrelaçamento de pixels. Os autores discutem que as diferenças entre as 2

primeiras abordagens são insignificantes do ponto de vista da compressão. Entretanto, a abordagem de entrelaçamento é útil apenas na compressão realizada por algoritmos projetados especificamente para imagens, apesar de perderem sua eficiência quando as imagens passam por transformações que quebram a coerência espacial.

3.5 Outras Medidas Baseadas em Compressão

Após pesquisas reportarem o potencial da NCD para descoberta de conhecimento em conjuntos de dados unidimensionais, outros trabalhos apontaram as limitações em problemas envolvendo imagens. Inúmeras propostas foram publicadas na literatura com o objetivo de aprimorar os resultados para dados bidimensionais. Essa seção discute alguns desses trabalhos e suas propriedades observadas experimentalmente e apresenta possíveis extensões das metodologias utilizadas.

3.5.1 Decomposição NID (α -NCD)

Partindo de trabalhos anteriores que sugerem melhores resultados com métricas de dissimilaridade baseadas em vetores de características, [32] propõem uma decomposição teórica da NID para explicar os resultados inferiores obtidos pela NCD. Essa decomposição é fundamentada nas porções de informação relevante e irrelevante das imagens. É sugerido que a NID pode ser melhorada quando a informação irrelevante é descartada, mantendo apenas a relevante para medição da similaridade. Os autores utilizaram nos experimentos uma imagem original conhecida como *Lenna* e mais duas versões modificadas: uma versão contendo ruído aditivo Gaussiano, com média $\mu = 0$ e desvio-padrão $\sigma = 1$, e outra versão com a face substituída pela imagem, também conhecida, *Mandrill*. Os resultados mostram que a NCD original entre as imagens visualmente semelhantes é maior quando comparada com imagens que possuem regiões de interesse para o sistema visual humano editadas, como é o caso da região de face. Assim, a NCD entre *Lenna* e a versão ruidosa foi maior que entre a versão modificada na face. Segundo os autores, essa limitação é explicada em razão da NCD tratar todos os bits da informação com a mesma relevância. Sugerimos que a quantização por plano de bits, eliminando planos mais ou menos aleatórios, poderia beneficiar o modelo. Segundo os autores, como a NCD não considera a relevância da informação do ponto de vista da percepção visual, isso pode resultar em uma avaliação de similaridade que não condiz com a similaridade subjetiva. Nesse contexto, hipotetizamos que a abordagem poderia ser melhorada utilizando-se um método de compressão que desse mais atenção às características (do ponto de vista semântico) ou aos bits (do ponto de vista quantitativo) mais relevantes. Isso justificaria a aplicação dos métodos de compressão com perda.

Com a decomposição das imagens de entrada em duas partes - *porção-modelo*, denotado por ρ_x , e *porção-residual*, denotado por $(x|\rho_x)$ - os autores propõem uma decomposição analítica da NID e, conseqüentemente, da NCD. A porção-modelo de uma imagem x representa uma aproximação e é obtida quantizando a imagem original. A porção-residual traz a informação necessária para reconstruir a imagem x a partir de sua porção-modelo e é obtida pela diferença entre a imagem original e sua porção-modelo, com os valores resultantes centrados em 128. Após quantizadas, embora as imagens passem a ter menos informação com apenas 2 cores, elas ainda mantêm informação visualmente relevante. Assim, os autores definem a decomposição da NCD como

$$\alpha\text{-NCD}(x, y) = \alpha \times \text{modelo-NCD}(x, y) + (1 - \alpha) \times \text{residual-NCD}(x, y), \quad (3-2)$$

onde

$$\alpha = \frac{C(\rho_y)}{C(\rho_y) + C(y|\rho_y)}, \quad (3-3)$$

e os componentes *modelo-NCD* e *residual-NCD* são calculados pela NCD original e definidos, respectivamente, por

$$\text{modelo-NCD}(x, y) = \text{NCD}(\rho_x, \rho_y) \text{ e} \quad (3-4)$$

$$\text{residual-NCD}(x, y) = \text{NCD}((x|\rho_x), (y|\rho_y)). \quad (3-5)$$

Nessa proposta, o termo α pode ser substituído por um escalar arbitrário entre 0 e 1, que define um *bias* subjetivo entre as porções modelo e residual. Para $\alpha = 1$, a avaliação da distância é realizada em função da porção-modelo (nível de abstração mais elevado) e, para $\alpha = 0$, a avaliação é direcionada para os detalhes da porção-residual (nível de abstração mais baixo). Essa abordagem é definida como uma versão da NCD com perda. Embora sejam utilizados compressores sem perda, a porção-modelo contém apenas uma aproximação dos dados originais. Os resultados sugerem que a porção-modelo da imagem contém informação suficiente para uma avaliação de similaridade. Em contrapartida, as porções-residuais contém detalhes que são irrelevantes.

3.5.2 Distância Normalizada de Compressão Condicional (NCCD)

Em [117], os autores relatam que trabalhos anteriores demonstram que a NCD funciona bem para identificar similaridade entre imagens, com partes adicionadas ou subtraídas. No entanto, o desempenho é ruim quando uma das imagens sofre transformações geométricas. A justificativa é a fraca aproximação de $K(x|y)$ para $C(xy)$ utilizando um compressor normal. Para solucionar esse problema, é proposto um framework baseado

em uma série de transformações aplicadas às imagens. A partir dessas transformações, utiliza-se a menor descrição comprimida que realiza a aproximação entre os pares de imagens. Na aproximação da NID, eles substituem a concatenação e assumem um compressor condicional C_F de modo que, sem perda da generalidade, $K(\cdot|\cdot) \approx C_F(\cdot|\cdot)$. A Distância Normalizada de Compressão Condicional (NCCD - *Normalized Conditional Compression Distance*) é definida como

$$\text{NCCD}(x, y) = \frac{\max\{C_T(x|y), C_T(y|x)\}}{\max\{C(x), C(y)\}}. \quad (3-6)$$

A construção de C_F é realizada a partir de um conjunto de transformações $\mathcal{F} = \{F_i | i = 1, 2, \dots, N\}$. $F_i(x)$ representa a imagem transformada com a i -ésima transformação da imagem x e $p(F_i, x)$ denota os parâmetros utilizados na transformação. Cada transformação é associada a um compressor de parâmetros C_i^p . Assim, C_F é definido como

$$C_F(y|x) = \min_i \{C[y - F_i(x)] + C_i^p[p(F_i, x)] + \log_2(N)\}. \quad (3-7)$$

Os autores citam como vantagem da NCCD a flexibilidade de incorporar diferentes tipos de transformação. Os resultados foram comparados com outras métricas de qualidade, MSE e SSIM. Os autores relatam que na transformação de contraste, a NCCD e a SSIM obtiveram resultados razoáveis. Nas transformações geométricas, somente a NCCD produziu avaliações significativas de similaridade. Por fim, a NCCD mostrou bons resultados em imagens de diferentes tamanhos e formatos, remoção de partes e inversão das cores dos pixels.

3.5.3 Método Campana-Keogh (CK-1)

Uma nova aproximação da complexidade de Kolmogorov para medir a similaridade entre texturas, usando um algoritmo de compressão de vídeo, é introduzida em [19]. Os autores justificam que a grande maioria dos algoritmos tradicionais requerem o ajuste cuidadoso dos parâmetros no domínio da aplicação. Como desvantagem, problemas essencialmente dependentes de muitos parâmetros são suscetíveis a *overfitting*. Essa parametrização excessiva é apontada por [37] como a maior dificuldade encontrada nas tentativas de reprodução dos resultados experimentais e entendimento das propostas. Em relação aos métodos livres de parâmetros e baseados em compressão, é discutido que os trabalhos anteriores são dependentes da linearização dos dados, ocasionando perda de informação espacial.

A proposta utiliza o princípio da compressão de vídeo, que explora tanto as redundâncias espaciais quanto as temporais de uma sequência de *frames*. A ideia é baseada na criação de vídeos artificiais, colocando os pares de imagens a serem avaliados

em sequência na forma de *frames*. Diante disso, propomos uma extensão de entrada para avaliação simultânea de diversas imagens; isso poderia permitir a avaliação do grau de similaridade de todo um grupo de imagens e não apenas de pares. Alternativamente, sugerimos a análise do comportamento da curva da taxa de compressão, concatenando o mesmo par de imagens sucessivamente. Acreditamos que essa estratégia pode melhorar a avaliação de similaridade, pois promove a suavização de possíveis ruídos à medida que o par é duplicado em sequência. O método de compressão utilizado foi o MPEG-1 em razão da ampla disponibilidade de implementações e das taxas de compressão alcançadas. Como o objetivo era maximizar a taxa de compressão e não manter o aspecto visual, foram utilizadas grandes escalas de quantização, dando preferência à compressibilidade em detrimento da qualidade. A medida nomeada CK-1 é definida por

$$\text{CK-1}(x,y) = \frac{C(x|y) + C(y|x)}{C(x|x) + C(y|y)} - 1. \quad (3-8)$$

A medida apresenta as propriedades de não-negatividade e simetria. Embora também apresente idempotência para um par de imagens idênticas, devido à compressão com perda, é possível obtermos distâncias 0 entre imagens não idênticas; portanto, não é uma métrica. Utilizando o *filtro de Gabor* e um *dicionário Texton*, os autores realizam experimentos em *datasets* com imagens em escala de cinza para testar o método sob invariância de cor. Dentre eles, os autores evidenciam os resultados obtidos em um *dataset* de aranhas e outro de mariposas. A medida CK-1 obteve resultados melhores que o trabalho original de [138] para o *dataset* de aranhas, considerando que o algoritmo original necessita de intervenção humana ocasional. Para o *dataset* de mariposas, os resultados foram equivalentes ao trabalho de [106], obtidos utilizando a informação de cor. Apesar dos excelentes resultados experimentais, os autores reconhecem que a medida CK-1 não é a melhor para todos os domínios. Pelo contrário, medidas que incorporem especificidades do domínio em questão podem ser melhores. Entretanto, para mineração de dados, onde não se conhece as características do problema, a CK-1 oferece a simplicidade e a generalidade necessárias para descoberta de conhecimento.

3.5.4 Distância Normalizada de Compressão Modificada (MNCD)

Duas modificações na aplicação e fórmula da NCD são propostas e discutidas em [158]. A primeira, em relação ao pré-processamento das imagens de entrada, sugere que sejam realizadas operações de suavização e *thresholding*. O objetivo é a supressão de detalhes visuais desnecessários para regularização do desempenho da NCD. A segunda modificação é realizada diretamente na fórmula com objetivo duplo: obter o tamanho da menor descrição que transforma uma imagem em outra e considerar o esforço computacional para construção dessa descrição. Assim, é proposto a Distância Normalizada de

Compressão Modificada (MNCD - *Modified Normalized Compression Distance*) como

$$\text{MNCD}(x,y) = \frac{C(xy) - \min\{C(x), C(y)\} + C(\text{abs}(x-y))}{\max\{C(x), C(y)\}}. \quad (3-9)$$

onde $\text{abs}(x-y)$ é o resultado da diferença absoluta entre os pixels das imagens x e y .

Os autores discutem que as modificações propostas permitem distribuições mais uniformes dos valores da NCD nas operações de processamentos e transformações sobre as imagens. Adicionalmente, leva em conta a similaridade entre as imagens considerando a diferença entre os pixels correspondentes. Com a adição desse termo na equação, a NCD se torna mais sensível à variações visuais sutis.

3.5.5 Distância de Compressão Rápida (FCD)

No trabalho de [25], os autores problematizam questões relacionadas à aplicação da técnica em *datasets* de médio e grande porte que, anteriormente, ou não foram abordadas ou foram discutidas superficialmente. É proposta uma nova medida utilizando dicionários para reduzir a complexidade do método, sem perda da eficiência. Segundo eles, o problema associado aos métodos baseados na NCD, quando aplicados em grandes bases, está relacionado à necessidade de processar, iterativamente, todos os dados sem utilizar representações mais compactas destes. Eles se baseiam em uma metodologia de classificação chamada Representação de Padrões baseada em Compressão de Dados (PRDC - *Pattern Representation based on Data Compression*) proposta por [164]. A ideia do PRDC é contornar a redundância da matriz de distâncias da NCD, gerando dicionários a partir da compressão LZ individual de todos os objetos. Em seguida, cada um desses dicionários são utilizados para comprimir os demais objetos e medir a similaridade [24]. Apesar da acurácia inferior, a PRDC é mais rápida pois evita a compressão dos objetos concatenados, etapa mais computacionalmente intensiva. Para contornar esse problema, os autores propõem uma nova medida de distância chamada Distância de Compressão Rápida (FCD - *Fast Compression Distance*), que alcança o mesmo desempenho da PRDC, mantendo a concatenação dos objetos realizada pela NCD.

Para eliminar a correlação dos canais RGB, as imagens são convertidas para o espaço HSV, mantendo cada canal independente na composição da cor. Em seguida, as cores são quantizadas em 16 níveis para o canal H e 4 níveis para os canais S e V, permitindo no máximo 256 cores de representação. A decisão de quantizar as imagens é justificada tanto para redução da complexidade dos dicionários, quanto para forçar a identificação de padrões recorrentes. Após processadas, as imagens são linearizadas em strings, preservando parte da informação de textura. Nessa etapa, os autores discutem que ocorre perda de informação espacial, destruindo a textura na direção vertical e mantendo, implicitamente, a textura na direção horizontal. Para contornar essa perda, foi adicionado

1 bit extra a cada valor de pixel para descrever a transição vertical entre seus vizinhos: 0 para transições suaves e 1 para transições acentuadas. Testes relevaram que o aumento de informação textural, além de 1 bit de informação, piora a performance do modelo causando *overfitting* na descrição da imagem. A partir da string, o codificador gera um dicionário com a compressão LZW e o ordena de forma crescente. Esse dicionário \mathcal{V} é utilizado na FCD definida por

$$\text{FCD}(x,y) = \frac{|\mathcal{V}(x)| - \cap(\mathcal{V}(x), \mathcal{V}(y))}{|\mathcal{V}(x)|}, \quad (3-10)$$

onde $|\mathcal{V}(\cdot)|$ representa o número de entradas do dicionário e $\cap(\mathcal{V}(x), \mathcal{V}(y))$ é o número de entradas encontradas, simultaneamente, em ambos dicionários (equivalente à compressão conjunta da NCD). Cada entrada do dicionário representa 1 unidade no tamanho do dicionário, independentemente do seu comprimento.

Em termos de desempenho, os autores reportam que quando os objetos são totalmente aleatórios (pior caso), a FCD é 4 vezes mais rápida que a NCD. É discutido que a FCD não privilegia nenhuma parte da imagem na avaliação de similaridade, ou seja, avalia o todo incluindo objetos e contexto. Os resultados mostram a robustez da FCD, explorando padrões significativos que descrevem a maioria do conteúdo da informação. Segundo os autores, o uso apenas do dicionário para descrição de um objeto elimina a dependência do tamanho do buffer utilizado pelos compressores. Mesmo passando por um processo de transformação com perda de informação espacial, a intersecção entre os dicionários permite que a informação mútua, deslocada espacialmente dentro do objeto, seja alinhada na avaliação de similaridade.

3.5.6 Distância Normalizada de Compressão Condicional Modificada (NCCD')

Em [131], os autores reforçam a argumentação de que a complexidade de Kolmogorov fornece uma descrição algorítmica para métricas de similaridade livres de extração e seleção de características. Entretanto, a aproximação por compressores do mundo real não funciona para imagens como nas demais áreas. Apesar da utilização da concatenação de objetos em substituição à compressão condicional ser a maneira mais fácil e intuitiva, estudos mostram que essa abordagem dificulta a eficiência das métricas aproximativas. Assim, qualquer outra estratégica pode ser utilizada para combinar um par de objetos. Para contornar essa limitação, a partir da NCCD, [131] analisam o desempenho comparado com a NCD e propõem uma variação, chamada Distância Normalizada de Compressão Condicional Modificada (NCCD' - *Modified Normalized Conditional Compression Distance*).

Os autores isolam o efeito do tipo de compressor propondo um novo algoritmo de compressão específico para imagens, que suporta tanto as compressões básica e conjunta, quanto a compressão condicional em 2 modos. O processo de compressão condicional de x e y do primeiro modo, denotado por $C(x|y)$, é realizado em 2 fases. Na primeira fase, y é representado utilizando uma modelagem de contexto finito de várias ordens. Na segunda fase, esse modelo é utilizado na compressão de x , conjuntamente com os próprios modelos aprendidos à medida que x é processado. Então, cada símbolo de x é codificado com uma mistura das probabilidades obtidas na construção dos 2 modelos. No segundo modo, denotado por $C'(x|y)$, a diferença principal está na codificação de x , que não passa por uma modelagem de contexto finito. Nesse caso, x é codificado exclusivamente com os modelos fixos obtidos de y . A $NCCD'$ é definida por

$$NCCD'(x,y) = \frac{\max\{C'(x|y), C'(y|x)\}}{\max\{|x|, |y|\}}, \quad (3-11)$$

onde $C'(x|y)$ denota o número de bits necessários para representar a versão comprimida de x , exclusivamente com as estatísticas de y ; e $|x|$ é o comprimento em bits de x sem compressão. Embora não seja discutido pelos autores, acrescentamos que devido a esse comprimento as imagens precisam estar na mesma escala para evitar o enviesamento do método, optando sempre pelo comprimento da maior imagem no denominador.

Foram realizados experimentos utilizando um conjunto de 11 imagens - 9 faces (3 de cada indivíduo), 1 randômica e 1 de baixa complexidade - para avaliar quanto a propriedade de idempotência é respeitada pela distância. As imagens foram quantizadas para 4 níveis de cinza para reduzir o número de contextos condicionados e a complexidade da execução. Nos resultados, os autores destacam a ampla faixa de valores obtida pela $NCCD'$. Além disso, foi observado que a medida proposta não ultrapassou o valor máximo de 1 e se comportou de forma simétrica em todos os casos. Os valores obtidos pela $NCCD'(x,x)$ também foram muito menores em relação à NCD e $NCCD$, sugerindo melhor conformidade da medida com a idempotência. Nesse ponto, destacamos que os autores poderiam ter investigado as demais propriedades de monotonicidade e distributividade de um compressor normal. Se confirmadas, a proposta poderia se enquadrar na melhor opção para aproximação da NID frente aos compressores comerciais. Os autores ainda discutem que as pesquisas caminham para explorar o potencial da complexidade de Kolmogorov aplicada em medidas efetivas de similaridade de imagens. Isso representa um grande nicho de investigação, diferentemente das demais abordagens de reconhecimento de padrões em imagens que são bem exploradas.

3.5.7 Distância Baseada na Representação Esparsa

Uma abordagem baseada em representação esparsa, codificando uma imagem com a informação de outra é proposta em [64]. A esparsidade dessa nova representação é utilizada como medida de compressibilidade em relação à outra imagem. Embora não sejam diretamente utilizadas técnicas de compressão de dados para avaliação de similaridade, a ideia principal é quanto mais esparsa for a representação de uma imagem, mais ela poderá ser comprimida. Segundo os autores, os dicionários são adequados para aproximar a forma como as imagens são percebidas, pois imitam a maneira como o cérebro as interpreta. O método é baseado no trabalho de [121] que demonstrou ser possível, a partir de imagens de treinamento, construir elementos básicos de representação inspirados no campo receptivo do córtex visual primário. Segundo os autores, essa representação em forma de dicionário deve ter 2 propriedades fundamentais: *sparsityprior*, assumindo que uma imagem de entrada pode ser descrita utilizando um grupo reduzido de elementos-base; e *overcompleteness*, na qual o tamanho desse grupo de elementos-base é maior que a dimensionalidade dos vetores de entrada. Assim, dado um par de imagens para avaliação, é gerado um dicionário para cada entrada. Em seguida, é computada a esparsidade de uma imagem representada com o dicionário extraído de outra.

Os autores discutem que representações esparsas de dados são ótimas candidatas para alcançar altas taxas de compressão. Eles também lembram que a superioridade do JPEG2000 está relacionada com a capacidade da DWT de tornar uma imagem mais esparsa no domínio tempo-frequência em comparação com a DCT no JPEG. O uso da esparsidade é justificado considerando-a como uma medida direta da aleatoriedade de um dado, que pode ser utilizada no reconhecimento de cenas. Isso é possível devido à redundância presente nesse tipo de imagem, que permite que a informação relevante possa ser representada por meio de uma decomposição de funções-base. Ao contrário, imagens aleatórias não podem ser decompostas em um pequeno número de elementos-base. Utilizando o K-SVD para gerar os dicionários \mathcal{V} , são definidas 2 medidas S de compressibilidade da imagem x . Na primeira, denominada *Complexidade Esparsa*, denotada por $S(x, \mathcal{V}_x)$, é medido o quanto a imagem pode ser comprimida utilizando seu próprio dicionário. Essa medida é equivalente à complexidade incondicional $K(x)$ e à compressão individual $C(x)$. Na segunda, denominada *Complexidade Esparsa Relativa*, denotada por $S(x, \mathcal{V}_y)$, é medida a compressibilidade de x utilizando o dicionário obtido a partir de y . Essa medida é equivalente à complexidade condicional $K(x|y)$. Assim, a distância d_S é determinada por

$$d_S(x, y) = \frac{S(x, \mathcal{V}_y) + S(y, \mathcal{V}_x)}{S(x, \mathcal{V}_x) + S(y, \mathcal{V}_y)} - 1. \quad (3-12)$$

Discutindo as propriedades de S , os autores destacam comportamentos de idem-

potência, monotonicidade e simetria. Em relação à medida de distância d_S , afirmam que ela não é considerada uma métrica pois não satisfaz o axioma de desigualdade triangular; entretanto, é não-negativa e simétrica. Para avaliação da generalidade da distância, foram realizados experimentos com imagens em escala de cinza, comparando a coerência da proposta com a percepção humana. Os problemas investigados foram de similaridade, clusterização, recuperação de imagens e classificação, todos comparados com a medida CK-1 proposta por [19]. No problema de clusterização, os autores utilizaram 3 *datasets* de faces. Em 2 *datasets*, a acurácia da distância proposta foi superior à CK-1 em aproximadamente 5% e 11%, e inferior em aproximadamente 2% no outro. Nos experimentos de recuperação foi utilizado um *dataset* de textura. Os resultados foram medidos com a curva *Precision and Recall* e se mostraram superiores à CK-1. Nos testes de classificação, os autores utilizaram a validação cruzada *leave-one-out*. Foram executados testes em 8 bases diferentes, que demonstraram superioridade da distância proposta em relação à CK-1 em 5 delas. Os autores ainda discutem que a maioria dos métodos de similaridade baseados em compressão usam o compressor como uma caixa-preta. Isso tornando difícil o entendimento de qual componente é responsável pela estimativa de aleatoriedade e medida de similaridade. Por isso, o método se propõe como uma abordagem direta da aproximação da complexidade, podendo ser facilmente estendido para qualquer outro contexto de dados.

3.5.8 Compressão Normalizada Relativa (NRC)

Em [39] é apresentado um método para classificação de dígitos manuscritos utilizando a Compressão Normalizada Relativa (NRC - *Normalized Relative Compression*) proposta por [132]. Essa medida foi desenvolvida para utilizar uma classe específica de compressores capazes de realizar a compressão condicional exclusiva, na qual a compressão de x é realizada exclusivamente utilizando a informação de y e é definida por

$$\text{NRC}(x, y) = \frac{C(x|y)}{|x|}, \quad (3-13)$$

onde $C(x|y)$ denota a quantidade mínima de bits necessários para representar x a partir de y , y é o modelo de compressão estatística treinado e $|x|$ é o comprimento string x .

As imagens de treinamento e de teste são submetidas a sucessivos processamentos e transformações para favorecer a identificação correta dos dígitos. Primeiramente, é realizado um processamento de alinhamento do ângulo de escrita para remover inclinações. Em seguida, as imagens são escaladas e binarizadas para reduzir o alfabeto utilizado. Os processamentos de escala e *thresholding* são classificados pelos autores como componentes de perda de informação do modelo proposto. O processo de compressão é realizado utilizando um FCM, como proposto em [130]. Os autores discutem que para dados multi-

dimensionais a noção de passado recente (profundidade do contexto) não necessariamente está ligada somente à ordem de processamento, mas também à proximidade espacial.

A proposta é dividida nos módulos de treinamento e codificação. No módulo de treinamento, todos os pixels de todas as imagens passam por um *thresholding* e são contabilizados em uma tabela de *hashing* de acordo com seu contexto. Cada classe de dígito é treinada separadamente, gerando 10 modelos diferentes. No processo de codificação, dada uma imagem de teste, para cada modelo gerado é calculada a probabilidade de ocorrência de cada contexto para estimar a quantidade de bits necessários para codificar a imagem em cada classe. Ao final, o modelo que fornece a menor codificação, ou seja, a menor representação em bits, determina a classe predita. Nos experimentos, os autores utilizaram os métodos de linearização *raster* horizontal e vertical e o zigue-zague vertical, diversos tamanhos de imagens e várias profundidades de contexto. Os resultados reportados mostram que, para qualquer estratégia de linearização, independentemente do tamanho da imagem, à medida que a profundidade do contexto aumenta, a taxa de erro diminui, chegando a 0,0267 de erro.

3.6 Outras Aplicações de Medidas de Similaridade por Compressão

A NCD foi proposta inicialmente para aplicação direta em problemas de agrupamento de objetos de diversos tipos, caracterizando uma métrica de propósito geral. Entretanto, alguns trabalhos têm tentado fazer uma conexão da abordagem de similaridade por compressão com outros algoritmos de aprendizagem de máquina. Essa seção discute algumas dessas tentativas.

Em [34], a NCD é utilizada como técnica de extração de características. A estratégia é baseada na escolha de T objetos de treinamento rotulados. A partir desses objetos, chamados de *âncora*, para cada amostra de teste é criada uma representação U -dimensional. Essas representações, na forma de vetores de características $\mathbf{x} = \{x_1, x_2, \dots, x_U\}$, são obtidas calculando $x_i = \text{NCD}(o, a_i)$ para todos os objetos desconhecidos o , onde a_i é o i -ésimo objeto-âncora. A partir disso, os autores discutem que existem, pelo menos, duas boas escolhas de algoritmos de aprendizagem de máquina para utilização desses vetores de características: Redes Neurais Artificiais (ANN - *Artificial Neural Networks*) e SVMs. Em relação às ANNs, é discutido que a intensa parametrização característica desse método, o torna difícil de ser integrado à abordagens de aprendizagem livre de parâmetros. Então, utilizam o SVM para um problema de classificação de dígitos numéricos manuscritos (0 a 9) com 0,85 de acurácia.

No trabalho de [31], os autores apresentam duas abordagens de reconhecimento

de padrões. Na primeira é investigado um problema de agrupamento de imagens em escala de cinza utilizando a NCD. Na segunda abordagem, os autores investigam a classificação multi-classe de imagens de satélite. No primeiro experimento, as imagens em escala de cinza foram codificadas utilizando os 256 caracteres da tabela ASCII após serem quantizadas para duas cores (preto 1 e branco 0). Essa codificação é obtida a partir de uma janela deslizante 8x2 pixels que, à medida que percorre a imagem, extrai palavras-código de 2 bytes de cada pixel. Essas palavras-código são representadas em vetores bidimensionais obtidos, a partir da quantização em 256 protótipos, cada um associado a um símbolo ANSI. Em seguida, são calculadas as complexidades LZ das sequências geradas, em substituição aos compressores da NCD, e determinado o agrupamento das imagens. A eficiência do método é descrita de forma subjetiva, apresentando os resultados dos agrupamentos de desenhos de frutas e veículos. Na fase de pré-processamento das imagens de satélite, os pixels foram quantizados individualmente utilizando os símbolos UTF-8 para representar 1000 protótipos (10 níveis de quantização para cada um dos canais RGB). As amostras de treinamento de cada classe - mar, árvores e cidade - foram recortadas em tamanho de 16x16 pixels. As imagens de teste, de 272x192 pixels, foram divididas em janelas de mesmo tamanho dos protótipos. Para cada amostra de treinamento, foi calculada a distância para essas janelas. Para cada classe foram computadas as ocorrências de acordo com as menores distâncias e determinado o percentual de área mar, cidade e árvores em cada imagem. Embora tenham empregado abordagens inovadoras utilizando compressão de dados, o trabalho não descreveu ou discutiu as taxas de acerto de forma objetiva ou comparou com outros métodos, deixando uma lacuna para prosseguimento da investigação.

Em uma pesquisa recente, [38] utilizam a NCD para calcular a distância entre imagens originais e sequências transladadas e rotacionadas dessas imagens. Foram utilizados compressores comerciais de propósito geral, específicos para imagens e experimentais. As distâncias foram organizadas em vetores de características e calculado o coeficiente de correlação de Pearson entre esses vetores. Os experimentos realizados foram de detecção de similaridade entre as imagens originais e versões modificadas com processamento de: suavização, adição de ruído, compressão com perda, equalização de histograma, recorte e transformações de escala e rotação. Os resultados apontam que o cálculo direto da NCD falha ao encontrar similaridade entre as imagens originais e processadas. No entanto, obtém bons resultados com as técnicas de compressão FPAQ0f nas transformações de rotação e JPEG2000 nas translações verticais, ambas concatenando as imagens horizontalmente para a compressão conjunta.

3.7 Contribuições e Lacunas de Pesquisa

Nessa seção, apresentamos na Figura 3.2 os pontos de saturação e as lacunas deixadas pelos trabalhos anteriores. Relacionamos as pesquisas mais relevantes dos últimos anos com as problematizações discutidas pelos respectivos autores e as contribuições deixadas por estes. A seguir, elucidamos os critérios utilizados para cada um dos pontos avaliados.

As linhas da Figura 3.2 representam as dimensões das contribuições em relação às técnicas de reconhecimento de padrões por compressão de dados. Essas dimensões são sumarizadas na Figura 3.1 e discutidas de forma contextualizada ao longo da Seção 3.4. Para cada dimensão de contribuição, foram correlacionadas as problematizações discutidas nos resultados, nas reflexões levantadas e nas dificuldades e restrições relatadas pelos autores. Os trabalhos são colorizados conforme as aplicações propostas nos experimentos como, por exemplo, avaliação de qualidade de imagens, similaridade de imagens ou propósito geral. Também foram incluídos o propósito da aplicação (classificação e/ou clusterização) e se as imagens são pré-processadas (quantização de cores e/ou conversão para tons de cinza). Em destaque, as células em vermelho indicam os pontos de interseção entre as contribuições e as problematizações propostas no presente trabalho. As colunas da Figura 3.2 organizam cada uma dessas questões:

- *Aplicabilidade de compressor normal*: Um compressor é dito normal se satisfaz as propriedades de idempotência, monotonicidade, simetria e distributividade [34]. Entretanto, os algoritmos dedicados para imagens, apesar de obterem ótimas taxas de compressão, não preenchem as características necessárias para serem considerados compressores normais [130]. A aplicabilidade de compressores normais é discutida para compreender as restrições quando não há disponibilidade das propriedades definidas por [34]. Nessa problemática, os esforços se concentram na proposição de métricas, medidas e *toolkits*. As pesquisas que tratam dessa temática tendem a não investigar os efeitos do uso de compressores, normais ou não-normais, em relação aos demais parâmetros como, por exemplo, técnicas de linearização e concatenação, formatos de arquivos ou os efeitos das transformações e processamentos de imagem. Assim, os autores partem do princípio que apenas os compressores normais são capazes de criar modelos dos dados, coletando estatísticas e dependências, à medida que avançam no processo de compressão e acumulam o conhecimento [130]. Entretanto, acreditamos que trabalhos que investiguem a não-normalidade dos compressores, de propósito geral e específicos para imagens, sob os mais diversos delineamentos experimentais, poderiam esclarecer o impacto dessa restrição na extração de semântica;

PROBLEMATIZAÇÃO											
	Aplicabilidade de compressor normal	Desempenho dos compressores	Relação taxa de compressão x extração de semântica	Tamanho dos arquivos ou volume de dados	Princípio da idem potência	Tempo de compressão	Compressão com perdas ou supressão de detalhes	Desempenho em transformações geométricas	Exploração dos efeitos da aleatoriedade	Aspectos da percepção visual humana e suas abstrações	Comparativo com métricas objetivas de qualidade
Métricas e medidas	Gilbride & Wilányi (2005) Pinho & Ferreira (2017) Pinho et al. (2014)		Campana & Koepfl (2010) Guba & Vetro (2014) Guba & Vetro (2014)	Cerra & Dutoit (2011) Chester & Itskhanyan (2012)	Gilbride & Wilányi (2005) Campana & Koepfl (2010) Guba & Vetro (2014) Pinho et al. (2014)	Cerra & Dutoit (2011) Nank & Achary (2016)	Chua & Tschier (2014) Tran (2007) Cerra & Dutoit (2011) Chester & Itskhanyan (2012)	Tran (2007) Chua & Tschier (2014) Nank & Achary (2016) Tran (2019) Pinho & Ferreira (2017) Guba & Vetro (2014) Nank & Achary (2016)	Guba & Vetro (2014)	Tran (2007) Chua & Tschier (2014) Nank & Achary (2016) Tran (2019) Pinho & Ferreira (2017) Guba & Vetro (2014) Nank & Achary (2016)	Milward & Wang (2010) Guba & Vetro (2014) Nank & Achary (2016)
Toolkits	Gilbride & Wilányi (2005) Sambur et al. (2017)										
Algoritmos de compressão	Gilbride & Wilányi (2005)	Tran (2007) Heidemann & Ritter (2008) Heidemann & Ritter (2009) Pinho & Ferreira (2017) Vazquez & Marco (2012) Coltur et al. (2019)	Pinho & Ferreira (2011) Vazquez & Marco (2012)	Vazquez & Marco (2012)	Vazquez & Marco (2012)		Heidemann & Ritter (2008) Montemsen et al. (2009) Heidemann & Ritter (2009) Vazquez & Marco (2012) Coltur et al. (2019)	Tran (2007) Montemsen et al. (2009) Vazquez & Marco (2012) Coltur et al. (2019)			
Formatos de imagens		Vazquez & Marco (2012) Coltur et al. (2019)	Vazquez & Marco (2012)		Vazquez & Marco (2012)		Vazquez & Marco (2012)				
Transformações geométricas		Heidemann & Ritter (2008) Montemsen et al. (2009) Vazquez & Marco (2012)	Heidemann & Ritter (2008)				Montemsen et al. (2009)	Heidemann & Ritter (2008) Heidemann & Ritter (2009) Campana & Koepfl (2010) Coltur et al. (2019)		Nank & Wang (2010) Tran (2007) Tran (2019)	Nank & Wang (2010)
Processamentos de imagem		Tran (2007) Coltur et al. (2019)					Chua & Tschier (2014) Cerra & Dutoit (2011)		Cebalre et al. (2007)	Tran (2007) Chua & Tschier (2014) Nank & Wang (2010) Tran (2019) Cerra & Dutoit (2011)	Milward & Wang (2010) Nank & Wang (2010)
Linearização		Montemsen et al. (2009)		Correa (2017)				Montemsen et al. (2009)			
Concatenação	Pinho et al. (2014)	Montemsen et al. (2009) Vazquez & Marco (2012) Coltur et al. (2019)						Montemsen et al. (2009)			
Métodos de compressão		Chester & Itskhanyan (2012)	Campana & Koepfl (2010) Pinho et al. (2014)	Correa (2017)	Campana & Koepfl (2010) Pinho et al. (2014)		Campana & Koepfl (2010) Pinho & Ferreira (2017) Correa (2017)				Nank & Wang (2010)

Figura 3.2: Contribuições e problematizações sobre similaridade de imagens utilizado compressão de dados.

APLICABILIDADES

- Qualidade de imagens
- Propósito geral
- Similaridade de imagens
- Análise por específica
- Extensão de características

PRE-PROCESSAMENTO

- Quantização e escalas
- Escola de cor

PROPOSTO

- Classificação
- Clustering

- *Desempenho dos compressores*: A temática mais observada pelos trabalhos é o desempenho de diversos tipos de compressores. As pesquisas tendem a se concentrar na investigação multifatorial do uso de compressores comerciais em diversos formatos, transformações e processamentos de imagens, bem como as técnicas de linearização e concatenação dos dados. Entretanto, os autores que propuseram novas métricas, medidas ou *toolkits* não comparam o desempenho de suas propostas utilizando outros compressores. Isso se deve, em grande parte, ao fato que as novas métricas e medidas são projetadas para uso de compressores específicos, inviabilizando a aplicação de outros métodos. Portanto, é desejável a proposição de novas medidas de similaridade, aptas a explorar uma maior variabilidade de técnicas de compressão, que não fiquem restritas a determinadas aplicações e possam ser expandidas para outros contextos;
- *Relação taxa de compressão vs. extração de semântica*: Os trabalhos que tratam dessa problematização discutem, direta ou indiretamente, o *tradeoff* entre os objetivos da compressão para redução do volume de dados e para reconhecimento de padrões. Embora esse tema não seja o ponto focal dos trabalhos relacionados, os autores desenvolvem os experimentos e discutem os resultados buscando o equilíbrio entre quais dados são relevantes e quais podem ser descartados. Em geral, essa problematização está relacionada a dois principais fatores: a escolha do esquema de compressão e a emulação do sistema visual humano. Consideramos esse tema como aberto visto que as discussões são permeadas por duas premissas complementares, mas ainda não abordadas conjuntamente: 1) altas taxas de compressão sem perda favorecem a aproximação da informação mutuamente compartilhada entre um par de objetos e 2) a qualidade dessa aproximação é condicionada ao algoritmo de compressão utilizado, desde que este tenha as propriedades de um compressor normal;
- *Tamanho dos arquivos ou volume de dados*: Até o presente momento, as propostas encontradas na literatura são superficiais em relação ao desempenho computacional dos métodos ou a restrição de tamanho dos dados. Os trabalhos são essencialmente dedicados a investigar as limitações das técnicas ou as taxas de acurácia. Embora o uso da compressão de dados para reconhecimento de padrões tenha a vantagem de ser eficiente na representação e armazenamento de dados, ainda carece de intensivas rotinas de processamento que poderiam inviabilizar sua aplicação em larga escala. A pouca atenção dada a essa questão nos leva a concluir que se trata de um tema com grandes perspectivas de avanço e potencial de investigação. Como discutido por [25],

a construção da matriz de distâncias entre os pares de objetos é uma tarefa computacionalmente dispendiosa e necessária a cada inclusão de uma nova amostra de predição. Essa limitação é ainda mais acentuada quando envolve processamento de imagens. Citamos como tema relevante de investigação o estudo de *pipelines* de paralelização dessa abordagem com o objetivo de viabilizar experimentos com grandes quantidades de imagens;

- *Princípio da idempotência*: A propriedade da idempotência de um compressor normal é definido por [34] como a capacidade de identificar repetições exatas. Assim, espera-se que o tamanho da compressão de um objeto individual seja igual ao tamanho da compressão resultante da concatenação desse mesmo objeto e sua cópia. É uma tendência dos trabalhos a investigação dessa propriedade nos experimentos preliminares. O objetivo é definir quais algoritmos de compressão ou formatos de imagens, por exemplo, devem ser considerados relevantes;
- *Tempo de compressão*: Como discutido anteriormente, a maioria dos trabalhos relatados não focam no desempenho dos modelos. A atenção é prioritária para as taxas de acurácia das técnicas e não as relacionam ao custo computacional necessário para obtê-las. Consideramos que a abordagem dessa temática deve ser realizada em conjunto com a limitação do volume de dados, visto que o esforço computacional aumenta exponencialmente na etapa de compressão conjunta dos objetos;
- *Compressão com perdas ou supressão de detalhes*: Destacamos nessa revisão a grande atenção dada às técnicas de compressão com perdas e a supressão de informação irrelevante nos dados. Os trabalhos voltados para aplicação em imagens partem do pressuposto que, para esse tipo de dado, o modelo de detecção de similaridade deve estar apto a abstrair detalhes. Somente assim, as técnicas podem ser resistentes a ruídos e pequenas variações dos dados. Para isso, os autores tendem a utilizar a quantização para forçar a recorrência de padrões e favorecer os métodos de compressão. Entretanto, destacamos que os algoritmos de compressão com perda não se enquadram como compressores normais [130] como estabelecido por [34];
- *Desempenho em transformações geométricas*: Observa-se recorrência nas discussões que relatam grandes variações de desempenho em relação às transformações geométricas nas imagens. Os autores reportam irregularidades dessas operações relacionadas aos algoritmos e também relatam que estas são sensíveis às técnicas de linearização e concatenação;
- *Exploração dos efeitos da aleatoriedade*: Embora sejam baseados na Teoria de Kolmogorov, que traz o conceito de aleatoriedade intrínseca, a problema-

tização com menor atenção é o impacto da complexidade na descrição dos objetos. Destaca-se que a definição de aleatoriedade de Kolmogorov é própria de qualquer objeto que possa ser descrito algorítmicamente por uma máquina de Turing. Assim, dizer que dados correlacionados possuem aleatoriedade é diferente de dizer que são estatisticamente aleatórios. Nesse sentido, acreditamos que o pré-processamento dos dados, visando ampliar ou reduzir sua complexidade, pode contribuir para o realce das principais características e favorecer a extração de semântica;

- *Aspectos da percepção visual humana e suas abstrações:* A aproximação da forma como o sistema visual humano percebe os estímulos externos é requisito para elaboração de qualquer medida ou métrica de similaridade de imagens. Essa afirmação pode ser observada na Figura 3.2, que relaciona grande quantidade de trabalhos sobre as relações entre os aspectos biológicos da visão e a modelagem computacional. De acordo com [59], a psicologia relaciona nossa percepção natural de aleatoriedade com a teoria matemática da complexidade algorítmica de Kolmogorov: quanto mais complexo o estímulo, mais aleatório será percebido. Em [64], os autores discutem a relação entre a esparsidade, considerada uma medida direta da aleatoriedade de um dado, e a quantidade de informação visual em cenas. Eles afirmam que a redundância presente nesse tipo de imagem permite que a informação relevante seja representada por meio de uma decomposição de funções-base. Em contraposição, imagens aleatórias não podem ser representadas com um pequeno número de elementos-base. Adicionalmente, [97] afirma que é amplamente aceita a ideia de que os sistemas sensoriais biológicos são naturalmente capazes de identificar propriedades estatísticas em sinais provenientes do ambiente natural. Em [7] e [9], os autores discutem essa capacidade biológica como a *hipótese da codificação eficiente*. Segundo essa teoria, o sistema sensorial é entendido como uma transformada que elimina redundâncias dos estímulos sensoriais de entrada [97]. Essa habilidade natural resulta em um conjunto de respostas neurais estatisticamente independentes [116];
- *Comparativo com métricas objetivas de qualidade:* Como último ponto focal dos trabalhos analisados, destacamos o uso de métricas objetivas de qualidade de imagens. Dentro dessa problematização, pode-se observar na literatura que as pesquisas sobre medidas e métricas de similaridade de imagens baseadas em compressão de dados se concentram em 2 frentes: 1) para fins de avaliação de qualidade visual ou 2) para comparativo de desempenho de novas medidas baseadas em compressão. Justificamos essa atenção especial ao tema em razão da avaliação de qualidade ser um subproblema de similaridade de imagens.

Método de classificação de cenas utilizando compressão de dados

Neste capítulo é proposto um método de classificação de cenas utilizando compressão, denominado *Reconhecimento de Padrões por Aleatoriedade* (RPA). A base de dados utilizada nos experimentos é apresentada e a arquitetura do PRA é detalhada.

4.1 Preliminares

Em [59], os autores afirmam que a forma de percepção natural da aleatoriedade está correlacionada com a teoria matemática da complexidade de Kolmogorov. Essa relação é ligada ao conceito de complexidade de um estímulo visual, ou seja, a quantidade de informações relevantes contidas em uma cena. Segundo o autor, quanto mais complexo o estímulo, mais aleatoriedade será percebida. Portanto, quanto mais informações significantes presentes em uma cena, maior será a complexidade de Kolmogorov. Para preservar ou potencializar as informações relevantes de uma cena, para sua classe verdadeira, são necessárias transformações. As transformações devem aumentar a complexidade intraclasse e simplificar a descrição extraclasse. Essas transformações são realizadas utilizando compressão de dados com perda.

Tanto a extração e seleção de características quanto a classificação são dependentes das semelhanças entre os objetos de uma mesma classe e do quanto eles se distinguem de outras classes. Acredita-se que é possível explorar a aleatoriedade desses objetos para o reconhecimento de padrões. Um exemplo clássico de aleatoriedade são os geradores de números aleatórios. Eles são projetados para produzir sequências de números reais, que simulam variáveis aleatórias independentes [85]. No entanto, embora possam ser validados por métodos estatísticos, eles são determinísticos, baseados em padrões repetidos de números, tornando esses geradores apenas pseudoaleatórios [56, 55]. Com testes adequados e volume suficiente de dados é possível identificar padrões nessas sequências pseudoaleatórias [102].

Se padrões podem ser encontrados em sequências pseudoaleatórias, a aleatoriedade da Complexidade de Kolmogorov pode ser utilizada como discriminante para agrupar cenas similares. Isso depende do nível de abstração das classes do problema. A princípio, sendo a imprevisibilidade dos acontecimentos característica fundamental do caos, não se pode afirmar que existe padrão na aleatoriedade de processos intrinsecamente indeterminados. Entretanto, acredita-se que é possível extrair informações e reconhecer padrões a partir da aleatoriedade de dados correlacionados espacialmente. Vale destacar que a definição de aleatoriedade de Kolmogorov é própria de qualquer objeto que possa ser descrito algorítmicamente por uma máquina de Turing. Assim, dizer que dados correlacionados possuem aleatoriedade é diferente de dizer que são estatisticamente aleatórios.

4.2 Base de Dados

A base de dados utilizada foi a *GHIM-10k*, organizada por [92], originalmente composta por 20 categorias de cenas, cada uma com 500 imagens de dimensões 400x300 ou 300x400 pixels no formato JPEG. Em razão da capacidade do hardware de processamento disponível, foram selecionadas 12 classes, com 100 imagens cada, a partir do *dataset* original: *árvore*, *avião*, *campo*, *carro*, *construção*, *corrida de motos*, *flor*, *fogos de artifício*, *montanha*, *moto*, *pôr do sol* e *praia*. As classes foram escolhidas de modo que, na ausência de cores, após todas as amostras serem convertidas para escala de cinza, algumas regiões das imagens, quando observadas isoladamente, pudessem ser confundidas com outras. Por exemplo, o céu de uma imagem da classe *praia* se assemelha ao fundo de imagens da classe *avião*; ou as árvores da classe *árvore* serem similares às árvores de uma imagem da classe *montanha*; regiões escuras presentes nas imagens tanto da classe *pôr do sol* quanto da classe *fogos de artifício*; ou partes das imagens da classe *campo* podendo também serem observadas na classe *montanha*. As 100 amostras de cada classe foram selecionadas aleatoriamente para evitar modelos de treinamento enviesados. A Figura 4.1 apresenta uma amostra de cada classe utilizada nos experimentos descritos neste capítulo.

Os experimentos foram realizados com as imagens convertidas em escala de cinza, utilizando a equação

$$I = 0,2126 \times R + 0,7152 \times G + 0,0722 \times B, \quad (4-1)$$

em conformidade com a especificação ITU-R BT.709 [72] que, de acordo com [124], representa a forma como olho humano percebe de maneira diferente o brilho entre as cores. Na Equação 4-1, o canal vermelho é representado por R e o canal de cor azul B recebe menos peso na conversão para o nível de intensidade I em relação ao canal verde



Figura 4.1: Exemplos de cenas utilizadas nos experimentos das classes, da esquerda para a direita e de cima para baixo: avião, praia, construção, carro, campo, fogos de artifício, flor, corrida de moto, moto, montanha, pôr do sol e árvore.

G. Isso se deve em razão da visão humana perceber os tons verdes com mais brilho em comparação com os tons azuis. As imagens foram pré-processadas para remoção das cores por 2 principais motivos:

- Segundo [71], a cor é um componente com grande potencial discriminante em problemas de similaridade de imagens. Portanto, optou-se por suprimi-la para avaliar a proposição do RPA sob condições de invariância dessa característica;
- Em condições normais, a visão humana é capaz de reconhecer objetos, cenas e pessoas utilizando apenas os níveis de intensidade captados pelos bastonetes. Segundo [66], é um processo que, predominantemente, utiliza informações acromáticas. A presente proposta é inspirada no sistema visual humano e procura explorar de forma simplificada e objetiva a maneira como as cenas são percebidas e processadas.

Em [35] são reportados resultados que demonstram correlação significativa, entre dados psicofísicos de imagens coloridas e em escala de cinza, em experimentos de avaliação subjetiva de complexidade visual. Esses resultados sugerem que a cor não influencia significativamente a percepção da complexidade de uma cena, considerando que o componente de luminosidade fornece informações suficientes sobre seu conteúdo semântico [36].

4.3 Arquitetura do Método Proposto

Nas abordagens anteriores, discutidas no Capítulo 3, 2 objetos são considerados similares se um pode ser significativamente comprimido utilizando a informação do outro [64]. Em outras palavras, recorrendo à informação adicional de um objeto, outro objeto pode ser descrito algorítmicamente de forma mais simples se estes compartilharem informações relevantes que os classifiquem dentro de um mesmo grupo. Seguindo essa

premissa, o RPA também investiga a compressibilidade de imagens, porém sob uma perspectiva de aproximação de uma cena desconhecida e protótipos treinados. O RPA utiliza, de forma implícita, como informação adicional, múltiplos *codebooks* gerados a partir de amostras de treinamento. O pré-processamento da imagem de teste com os *codebooks*, denominado *quantização*, pode ser caracterizado como uma agregação de informação adicional para descrição do objeto. Entretanto, a complexidade condicional não é aplicada de forma explícita. Portanto, utiliza-se a complexidade de um objeto único para classificá-lo. Outra diferença em relação às técnicas anteriores é a independência dos formatos de arquivo, pois atua no nível da matriz das intensidades dos pixels.

Segundo [97], o sistema sensorial é adaptado para encontrar propriedades estatísticas nos sinais provenientes de ambientes naturais. Isso permite que a redundância dos estímulos seja removida, resultando em um conjunto de respostas neurais estatisticamente independentes. Em [130], os autores afirmam que o desafio de medidas de similaridade aplicadas às imagens é duplo: avaliar similaridade sem a necessidade de um perfeito alinhamento espacial, mas que corresponda aos resultados fornecidos por nossa percepção visual. Tratando a complexidade como uma medida qualitativa da informação, pode-se afirmar que a quantização de uma imagem com protótipos de outra classe promove a degradação da informação, simplificação da descrição algorítmica e redução da complexidade. Ao contrário, uma imagem quantizada com protótipos de sua classe mantém ou amplia sua complexidade em relação às demais classes. Assim, uma imagem de teste tende a ser menos compressível quando quantizada com o *codebook* de sua classe verdadeira. Nesse caso, a opção pela maior complexidade na classificação é feita sob a premissa: *quanto maior a aleatoriedade de uma cena quantizada, mais informação, ou seja, mais elementos descritores do contexto foram preservados*.

A Figura 4.2 descreve as principais etapas do RPA, divididas entre os *Módulos de Treinamento* e de *Classificação*. A base do *Módulo de Treinamento* é a *etapa #1*, de *Prototipação*. Nela as imagens de treinamento fornecem amostras para uma heurística responsável por obter os protótipos das classes (*codevectors*) e organizá-los em *codebooks*. Para cada classe, o *codebook* é formado por diversos tamanhos de janelas e quantidade de protótipos, definidos em parâmetro. Na *etapa #2*, de *Quantização*, as imagens de trei-

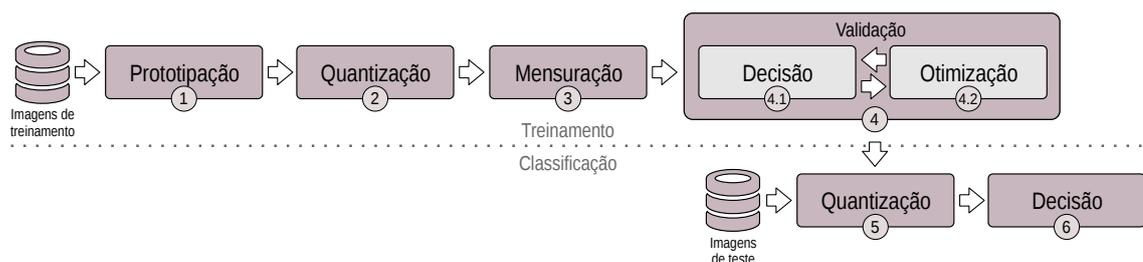


Figura 4.2: Principais etapas do modelo de treinamento e classificação proposto.

namento são reconstruídas com os *codevectors* obtidos na etapa anterior. Uma medida de complexidade da informação é calculada na *etapa #3, Mensuração*. Os melhores parâmetros de treinamento de cada classe são selecionados na *etapa #4 Validação*. Esses parâmetros validados são utilizados no *Módulo de Classificação* na *etapa #5 de Quantização* das imagens de teste. Por fim, na *etapa #6* é realizado o processo de tomada de *Decisão* para as amostras de teste. Cada uma dessas etapas serão detalhadas nas próximas seções.

4.4 Módulo de Treinamento

Esta seção descreve todas as etapas envolvidas no processo de treinamento do modelo, desde a preparação das amostras até a busca pelos melhores parâmetros para maximizar as taxas de acurácia da classificação. Como a classificação realizada pelo RPA não é fundamentada em fronteiras de decisão explícitas, ou seja, hiperplanos de separação extraclasses, isso permite que o treinamento seja realizado de forma independente para cada classe. Assim, o RPA possui uma arquitetura de aprendizagem escalável, possibilitando o aprimoramento do modelo de uma classe específica ou adição e remoção de classes sem a necessidade de retreinamento.

4.4.1 Prototipação

Na etapa de *Prototipação* apresentada na Figura 4.3, a partir das amostras de treinamento, são obtidos protótipos que descrevem suas respectivas classes. Inicialmente, para cada imagem das W classes do problema, no *passo 1.1* é realizada uma quantização de cores para redução da informação e simplificação das amostras de treinamento. Esse pré-processamento é sugerido por [25] como uma estratégia para favorecer a identificação de padrões recorrentes. Esse passo é caracterizado como o primeiro componente de compressão com perda, incluído na arquitetura para eliminar detalhes e ruídos das amostras. O algoritmo *median cut* foi escolhido para esta tarefa e, em razão da conversão das imagens em escala de cinza, este parâmetro permite ser configurado entre 2 e 256 cores, sempre em potência de 2.

No *passo 1.2* é definido o conjunto Γ com J tamanhos de janelas das amostras que serão fornecidas como entrada para o treinamento na forma $\Gamma = \{\gamma_j\}_{j=1}^J$. A arquitetura do RPA não limita o número de valores para esse parâmetro mas, apesar de permitir a escalabilidade, impacta diretamente no tempo de execução de todas as etapas do módulo. Dadas a altura M e largura N das janelas, as imagens são divididas em M' janelas na vertical e N' na horizontal. A Figura 4.4 mostra o exemplo de conversão para escala de cinza de uma imagem da classe *praia* e divisão em janelas 15x15. Em seguida, no *passo*

1.3, essas janelas são linearizadas na ordem linha-a-linha (*row-major*) para se tornarem vetores U -dimensionais. Os vetores constituem representações das janelas, na forma de conjuntos de símbolos.

1. Prototipação

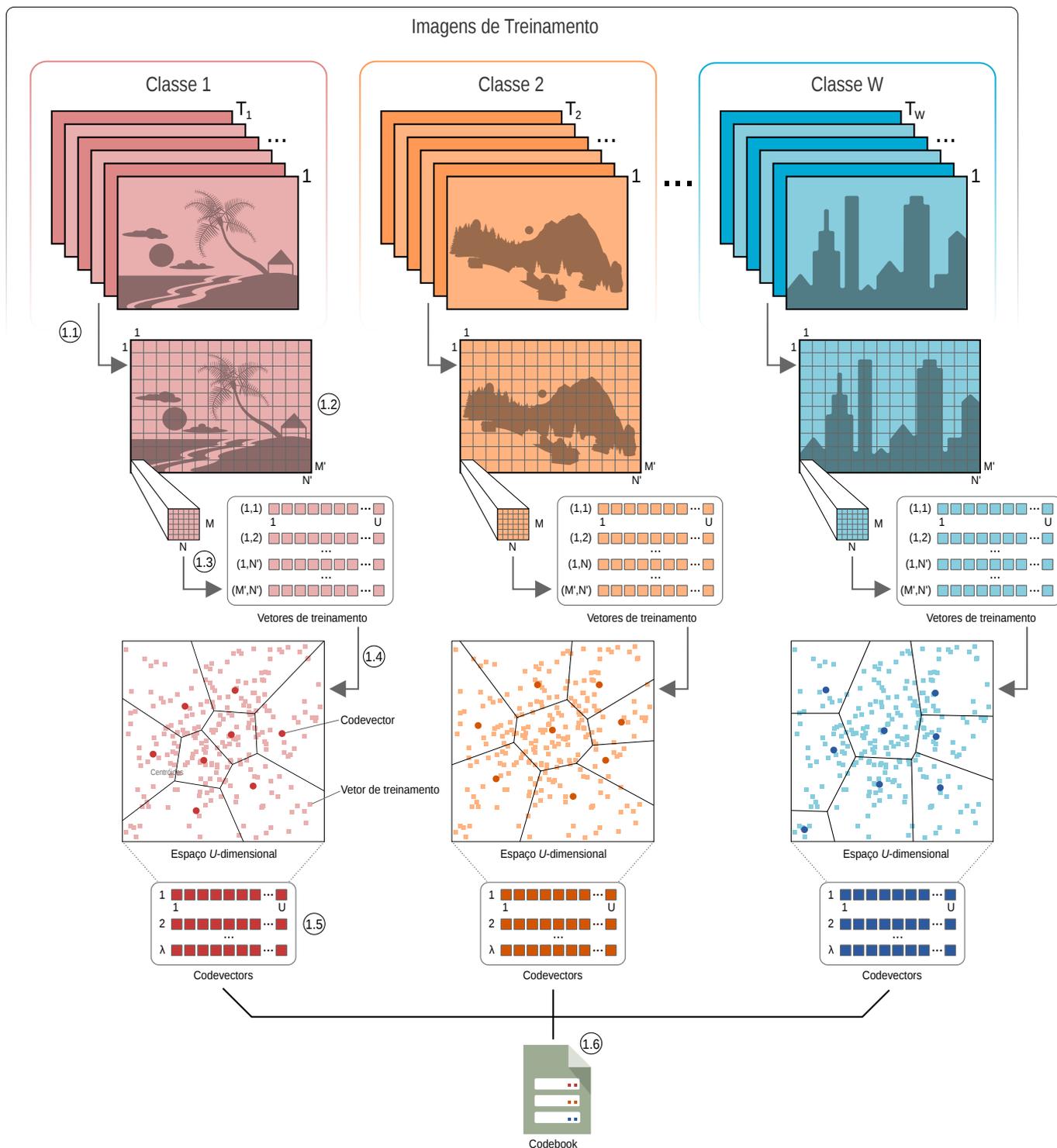


Figura 4.3: Etapa de prototipação, base do módulo de treinamento.

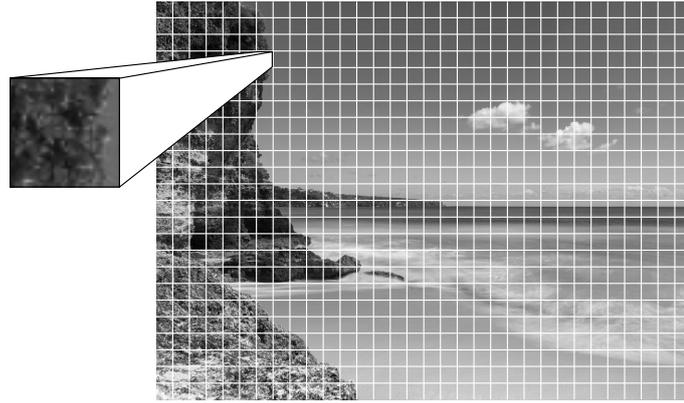


Figura 4.4: Exemplo de imagem, após a conversão para escala de cinza e quantização de cor, preparada para fornecer amostras de treinamento para a classe praia com janelamento 15×15 .

No *passo 1.4*, as amostras, agora denominadas *vetores de treinamento*, são projetadas em um espaço para aplicação do algoritmo LBG, que recebe como parâmetro o conjunto Λ com L tamanhos de *codebook* no qual espaço será dividido, na forma $\Lambda = \{\lambda_l\}_{l=1}^L$. A saída do algoritmo LBG no *passo 1.5* são os centroides das partições e são denominados *codevectors* ou *símbolos do dicionário*. Esse parâmetro deve obedecer à potência de 2 e também tem grande influência no tempo total de processamento do módulo. Entretanto, o cálculo dos *codevectors* λ_l é incremental. Isso permite que o particionamento do espaço seja aproveitado para os L *codevectors* do *codebook*. Porém, enquanto o tempo de execução da etapa de *Prototipação* permanece constante, o desempenho das demais etapas do módulo de treinamento sofre interferência. Por exemplo, deseja-se obter o conjunto de $L = 4$ *codevectors* $\Lambda = \{64, 128, 256, 512\}$ de uma mesma classe. O algoritmo particiona o espaço, sucessivamente, dobrando a quantidade de partições que inicia com 1, até chegar a 512. Portanto, a influência desse parâmetro no tempo da prototipação obedece ao limite do maior valor estabelecido para o conjunto Λ .

Ao final da etapa de *Prototipação*, todas as combinações de tamanhos de janelas Γ e quantidades de *codevectors* Λ , combinados em pares $[\gamma_j, \lambda_l]$ para todas as classes, por exemplo, $\{[10 \times 10, 64], [10 \times 10, 128], \dots, [15 \times 15, 64], \dots, [20 \times 20, 256], [20 \times 20, 512]\}$, são armazenadas em um único *codebook* no *passo 1.6*.

4.4.2 Quantização de Treinamento e Medidas de Aleatoriedade

Nas 2^a e 3^a etapas, as T imagens de treinamento das W classes são submetidas aos processos de *Quantização* e *Mensuração*, descritos na Figura 4.5. Na etapa de *Quantização* é empregada a quantização vetorial, responsável por realizar as transformações que ampliam as características mais discriminantes de cada classe. No *passo 2.1*, cada imagem de treinamento $\mathbf{X}^{(t)}$, onde $t = 1, 2, \dots, T$, é dividida em janelas, de acordo com os

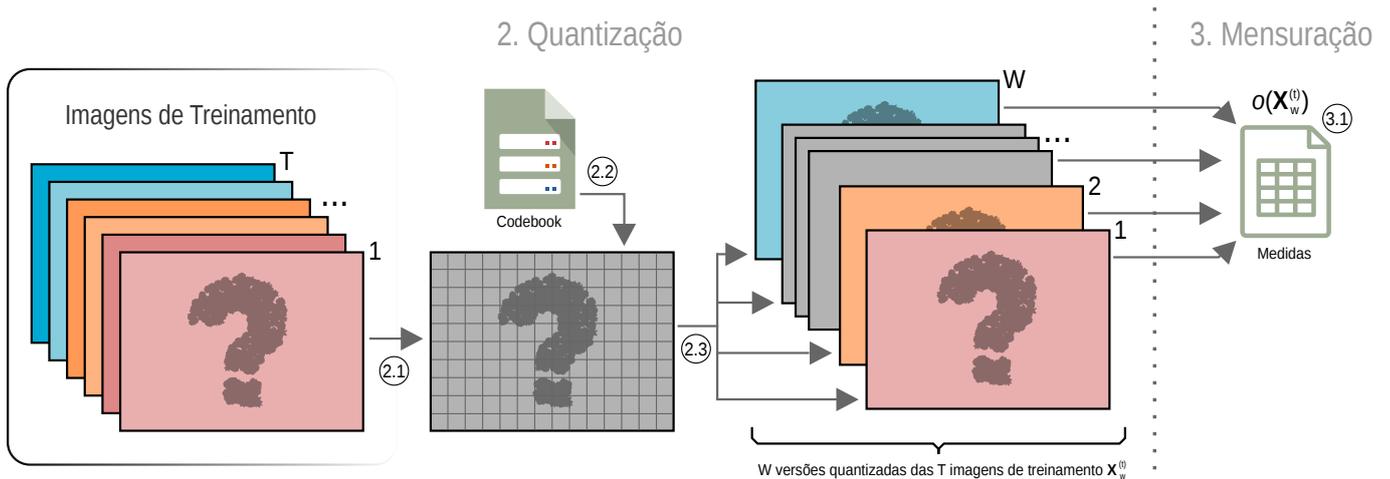


Figura 4.5: Etapas de quantização e cálculo da medida de classificação das imagens de treinamento.

tamanhos definidos no parâmetro Γ da *Prototipação* (passo 2.2). Para cada par $[\gamma_j, \lambda_l]$ do *codebook* obtido no passo 2.3, são geradas W versões quantizadas para as T imagens de treinamento, denotadas por $X_w^{(t)}$, onde $w = 1, 2, \dots, W$. O processo de quantização vetorial é realizado calculando a distância euclidiana entre cada janela e todos os *codevectors* do respectivo par $[\gamma_j, \lambda_l]$; a menor distância determina o *codevector* que substituirá os pixels da janela.

A etapa de *Quantização* pode ser considerada como a mais importante do RPA. Nessa etapa, as informações relevantes de uma cena, quantizada para uma classe divergente¹, são atenuadas em um processo de filtragem, em razão do distanciamento entre os símbolos do dicionário e as janelas da cena. Isso resulta em diferentes janelas da cena sendo representadas por um mesmo *codevector*. Em outras palavras, quando essa cena é reconstruída utilizando um dicionário divergente², a descrição das partes que a compõe é atenuada por meio de símbolos que se repetem. Ao contrário, quando o dicionário é convergente³, as partes são representadas por *codevectors* muitos próximos no espaço U -dimensional. Dessa forma, resulta em um casamento mais distribuído, portanto, mais aleatório, entre as janelas e os *codevectors*. Os detalhes de como a quantização convergente aumenta a aleatoriedade de uma cena são discutidos na Seção 5.2.4.

A Figura 4.6 mostra uma imagem original da classe *praia* (Figura 4.6(a)) após ser quantizada utilizando os *codebooks* obtidos no treinamento das classes *praia* (Figura 4.6(b)), *montanha* (Figura 4.6(c)) e *árvore* (Figura 4.6(d)) com janelas de tamanho 5x5 e

¹Termo utilizado para definir qualquer outra classe do problema que não seja a classe verdadeira da cena (classe convergente).

²*Codebook* formado por *codevectors* obtidos a partir de amostras de classes distintas da classe verdadeira da cena.

³*Codebook* formado por *codevectors* obtidos a partir de amostras de mesma classe da cena.

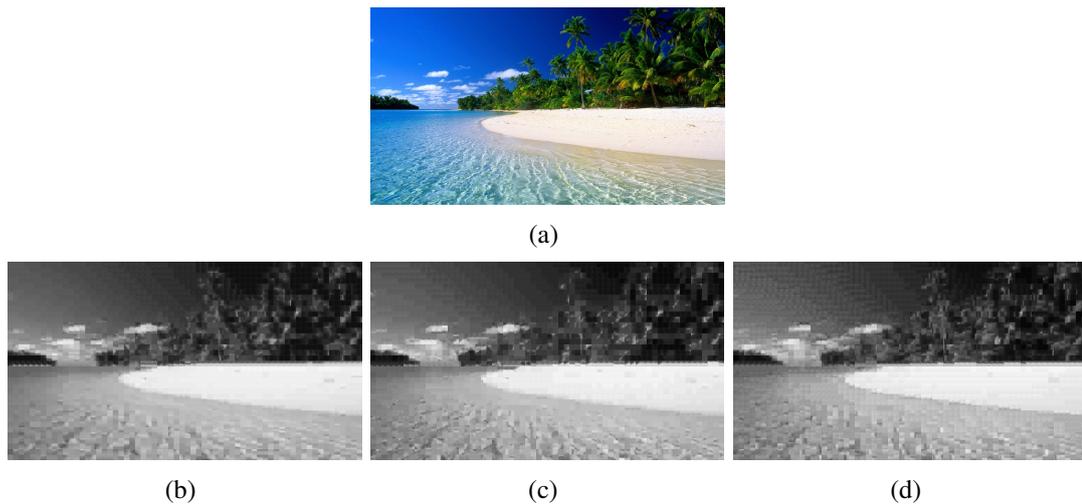


Figura 4.6: Exemplo de uma imagem original da classe praia (a) após as quantizações com janelas de tamanho 5x5 e dicionário com 64 símbolos para a classe praia (b), montanha (c) e árvore (d).

dicionário com 64 símbolos. Em uma rápida inspeção, visualmente não é notada nenhuma diferença entre as 3 versões quantizadas. Entretanto, após uma avaliação mais criteriosa, essas diferenças podem ser notadas. A distinção entre o grau de complexidade de cada uma das versões obtidas é realizada pelas medidas de aleatoriedade $o(\mathbf{X}_w^{(t)})$ na etapa *Mensuração*. $o(\mathbf{X}_w^{(t)})$ denota a medida da w -ésima versão quantizada da imagem de treinamento $\mathbf{X}^{(t)}$. No *passo 3.1*, essas medidas, descritas nas próximas subseções, são calculadas para todas as versões quantizadas das imagens e os resultados são armazenados em arquivo para posterior validação.

Entropia Algorítmica Aproximada por Compressão

A aleatoriedade das quantizações descritas anteriormente é calculada utilizando a compressão de dados como aproximação da complexidade de Kolmogorov. Essa medida quantifica a complexidade da descrição algorítmica da imagem. Assim, é possível determinar qual das versões quantizadas de uma cena possui mais informação visual relevante, dada pela maior complexidade. Dentro da arquitetura do RPA, essa medida pode ser calculado utilizando 2 estratégias:

- Tamanho da compressão: nessa estratégia as versões quantizadas das imagens são salvas em disco no formato BMP, sem nenhum tipo de compressão do formato. Em seguida, são aplicados algoritmos de compressão sem perdas de propósito geral - como ZIP, *gzip*, *bzip2* - para obter o tamanho do arquivo comprimido;

- Taxa de compressão: nessa estratégia, apresentada na Equação 2-7 da Seção 2.3, o comprimento da descrição da imagem é dividido pelo comprimento de sua respectiva descrição comprimida com os algoritmos LZW e *Huffman*. Quanto mais aleatoriedade presente na imagem menor é a taxa de compressão alcançada pelo compressor.

Na Figura 4.7 é detalhado o processo de quantização das imagens. No exemplo, a *imagem original* da classe *floresta* é quantizada em janelas 32x32 utilizando o dicionário treinado com 64 símbolos, também da classe *floresta*. Após o cálculo das distâncias entre todas as janelas e os *codevectors*, são obtidos os respectivos *índices dos codevectors* mais próximos. Como os *codevectors* são armazenados respeitando uma relação de ordem, esses índices são os identificadores de posição no intervalo $[0, \lambda-1]$, onde λ denota a quantidade de símbolos do dicionário. O cálculo da complexidade utilizando o *tamanho da compressão* é realizado na *imagem quantizada* e, portanto, atua em nível de byte como unidade de informação do arquivo. Já a medida de *taxa de compressão* é calculada diretamente a partir dos *índices dos codevectors*, linearizados linha-a-linha e comprimidos utilizando uma implementação de compressão por dicionário (LZW) e estatística (*Huffman*). A taxa de compressão T_C é calculada por meio da equação

$$T_C = \frac{M' \times N' \times n_{bits}}{|Z_{LZW+Huffman}|} \times (-1), \quad (4-2)$$

onde M' e N' denotam, respectivamente, a quantidade de janelas na vertical e horizontal; $|Z_{LZW+Huffman}|$ é o comprimento em bits da descrição comprimida dos *índices dos codevectors* e n_{bits} é obtido pela equação

$$n_{bits} = \left\lceil \frac{\log_{10}(\lambda - 1)}{\log_{10} 2} + 1 \right\rceil, \quad (4-3)$$

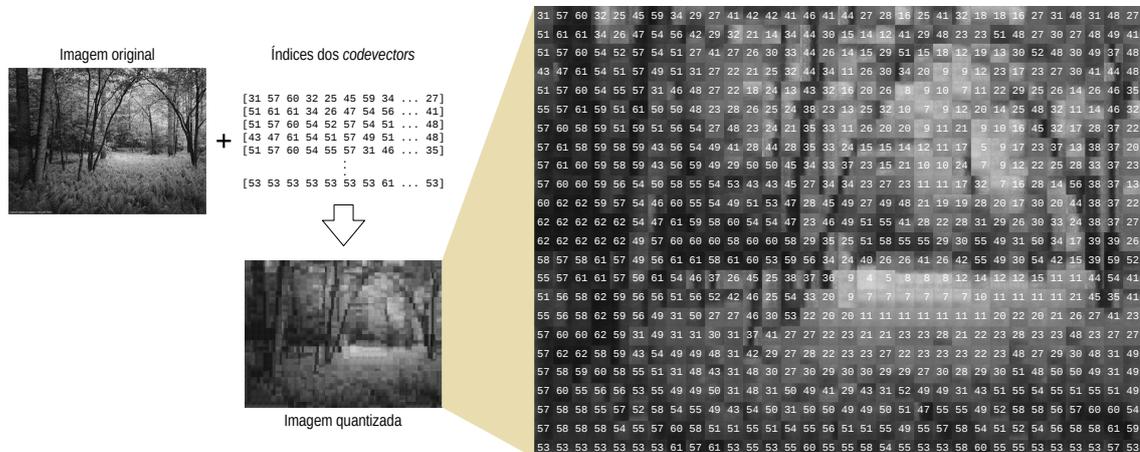


Figura 4.7: Imagem quantizada com os codevectors mais próximos de cada janela da imagem original.

onde λ representa a quantidade de *codevectors* do *codebook* utilizado na quantização. Para fins de padronização da regra de decisão, na qual quanto maior o valor da medida maior a aleatoriedade da descrição, a taxa de compressão T_C é multiplicada por (-1) na Equação 4-2. Como o processo de compressão não objetiva posterior reconstrução dos dados, tanto o dicionário do LZW quanto a árvore do *Huffman* são descartados para simplificação da descrição $Z_{LZW+Huffman}$. A escolha dos algoritmos foi realizada observando a predominância destes nos formatos de compressão de imagens sumarizados na Tabela 2.1 da Subseção 2.3.6. Entretanto, outros algoritmos podem ser explorados para identificação de diferentes padrões.

Entropia Média

A entropia de Shannon, da Teoria Clássica da Informação, também pode ser utilizada como medida de complexidade das versões quantizadas das imagens. Da mesma forma que na taxa de compressão com *LZW+Huffman*, o cálculo é realizado considerando apenas os *índices dos codevectors* e, quanto maior a entropia, maior a aleatoriedade da representação. A classe é predita pelo maior valor de entropia alcançado nas quantizações. Isso significa que a versão da imagem quantizada com dicionário convergente tende a manter maior equilíbrio quantitativo na escolha dos *codevectors*. As janelas de uma imagem de teste, quando projetadas no espaço de prototipação, ocupam, em grande maioria, a mesma região delimitada pelos *codevectors* de sua classe. Ao contrário, se comparadas com os *codevectors* de outras classes, são projetadas em outras regiões, com menos interseções. Essa observação, além de justificar o fato da entropia ser maior para quantizações convergentes, explica como as quantizações divergentes produzem sequências de *codevectors* menos aleatórias. As janelas de uma classe, quando quantizadas para outra, tendem a se aproximar de *codevectors* de fronteira, o que limita as opções de representação e ocasiona repetições.

A Figura 4.8 ilustra o processo de quantização vetorial de duas imagens A e B utilizando o *codebook* de uma classe qualquer. No espaço bidimensional, os vetores de treinamento da classe (círculos brancos) são particionados para obter 8 *codevectors* (círculos vermelhos). Ao projetar as janelas extraídas das imagens A (quadrados) e B (triângulos) no espaço, observa-se que as janelas da imagem A se distribuem mais uniformemente em torno dos *codevectors* da classe treinada. As janelas da imagem B, embora compartilhem parte do espaço ocupado pela classe, são predominantemente projetadas em uma região marginal. No caso da imagem A, por meio do cálculo da distância euclidiana, a nova representação será aproximada utilizando todo o *codebook* treinado para a classe (círculos vermelhos); gerando aleatoriedade e distribuição mais uniforme na escolha dos *codevectors*. Já para a imagem B, serão atribuídos como mais próximos apenas os *codevectors* marginais (círculos vermelhos marcados com x) na

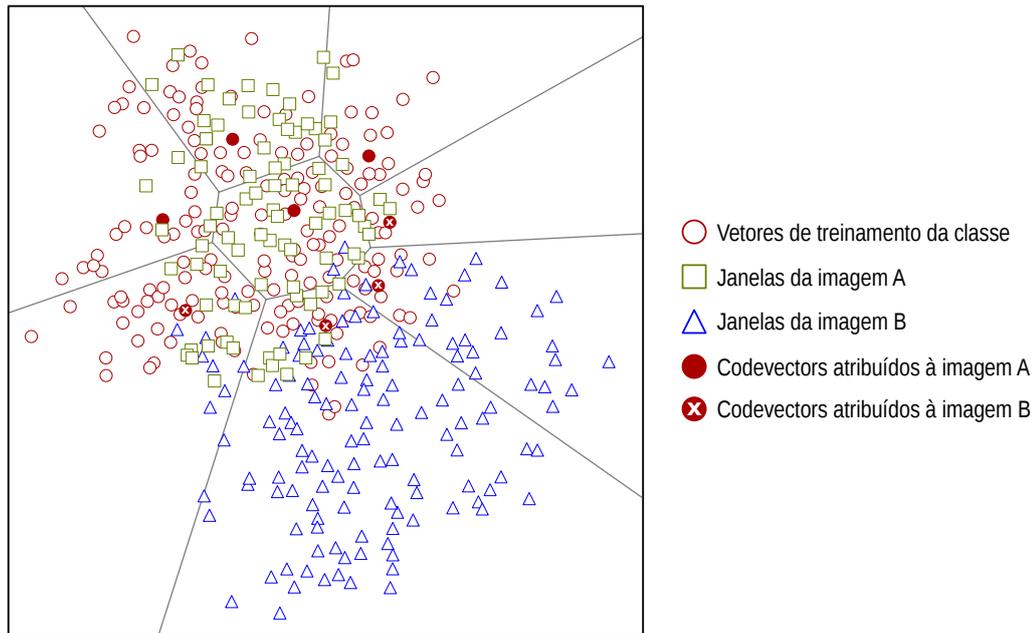


Figura 4.8: Exemplo das aproximações realizadas nas quantizações convergentes e divergentes visualizadas em um espaço bidimensional.

aproximação; promovendo repetições e maior probabilidade de uma distribuição menos equiprovável na escolha dos *codevectors*. Esse processo realiza quantizações, convergente para a imagem *A* e divergente para a imagem *B*. Então, a imagem *A* será classificada como pertencente à classe, decidindo pelo maior valor ou da entropia algorítmica ou da entropia média.

A Figura 4.9 apresenta 2 imagens distintas, formadas por pixels em 4 níveis de intensidade, distribuídos em quantidades semelhantes ao longo das imagens. Na Figura 4.9(a), os pixels de cores iguais estão agrupados continuamente e ordenados, do tom mais escuro para o tom mais claro. Na Figura 4.9(b), o mesmo conjunto de pixels foram dispostos de forma aleatória no espaço de mesmas dimensões. O princípio fundamental da compressão de imagens, discutido na Seção 1.1 do Capítulo 1, explora a alta probabilidade de, ao selecionar um pixel arbitrário em uma imagem, os pixels vizinhos serem da mesma cor ou muito similares. Nesse sentido, pode-se afirmar que a Figura 4.9(a) é altamente comprimível. A Figura 4.9(b), apesar de conter a mesma informação em termos quantitativos, não contém padrões que possam ser identificados pelos algoritmos de compressão. Entretanto, ao calcular a entropia de Shannon das 2 imagens, percebe-se que o valor da medida é idêntico para os dois casos, não sendo adequada para diferenciá-las.

Em contraponto à limitação da entropia discutida no exemplo acima, essa medida se mostra adequada para discriminar cenas no RPA. Seu comportamento se assemelha à capacidade cognitiva do ser humano em, apenas observando as peças de um quebra-cabeças dispostas desordenadamente sobre uma mesa, inferir o conteúdo da figura que tais

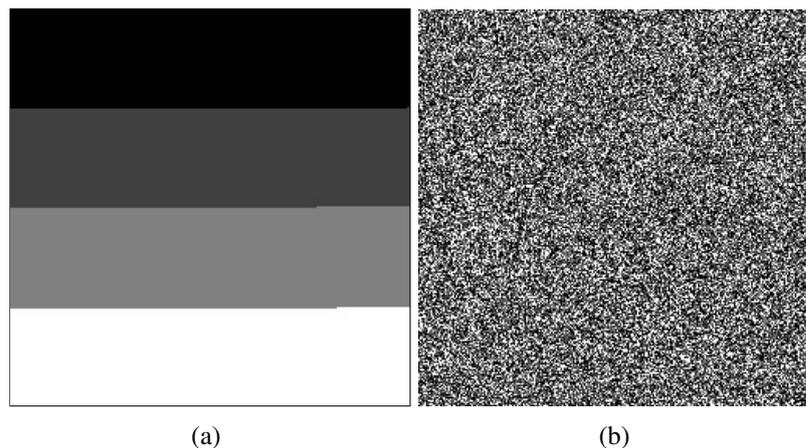


Figura 4.9: *Imagens formadas por 4 níveis de intensidade, divididos em proporções semelhantes, com os pixels agrupados em sequência (a) e distribuídos aleatoriamente (b).*

peças representariam após montadas. Obviamente, essa capacidade de percepção depende do tamanho, da quantidade de peças e da complexidade do conteúdo da figura. No caso da entropia isso não é diferente: da mesma forma que na compressão, a medida alcança diferentes acurácias para cada classe, sujeitas aos tamanhos da janela e do dicionário utilizados. Portanto, entende-se que o cálculo da entropia dos *índices dos codevectors* é capaz de classificar cenas com a informação fragmentada e desordenada.

4.4.3 Validação

Na *Validação*, são obtidos os melhores parâmetros $[\gamma_j, \lambda_l]$ de cada classe, ou seja, os melhores níveis de abstração. Na discussão dessa etapa, utiliza-se a seguinte terminologia:

- *Contexto*: universo do problema de classificação em questão, envolvendo todas as W classes;
- *Parâmetros*: par $[\gamma_j, \lambda_l]$, ou seja, a combinação entre um tamanho de janela e uma quantidade de símbolos utilizados no treinamento. Por exemplo, os parâmetros $[15 \times 15, 128]$ se referem ao treinamento de uma classe dividindo as imagens em janelas de tamanho 15×15 , submetidas à prototipação para obtenção de 128 *codevectors*;
- *Cenário* (δ): subconjunto de classes extraído do contexto, com seus respectivos parâmetros de treinamento. O tamanho do cenário é denotado por $|\delta|$. Por exemplo, um hipotético cenário com 3 classes seria $\{[\text{praia}, 8 \times 8, 256], [\text{floresta}, 8 \times 8, 128], [\text{montanha}, 32 \times 32, 64]\}$, com $|\delta| = 3$;

- *Rodada*: iteração do algoritmo de validação para encontrar os melhores cenários entre as combinações possíveis de classes. Por exemplo, para um contexto de 12 classes, pode-se analisar os melhores cenários para combinações de 2, 3, ..., 12 classes;
- *Acurácia mínima* (τ): valor de acurácia estabelecido para determinar se a discriminabilidade de um cenário é relevante para resolução do problema;
- *Passo de acurácia* (η): unidade de decréscimo da acurácia mínima (τ) para redução gradual. Em algumas situações nenhum cenário avaliado atinge a discriminabilidade mínima definida pela taxa de acurácia mínima. Nesse caso, o algoritmo decrementa τ para garantir que a avaliação do contexto não seja precocemente interrompida. Assim, até um tamanho de cenário a taxa de acurácia mínima é garantida, a partir disso o algoritmo continua a avaliação para as próximas rodadas, identificando os cenários com taxa de acurácia inferior.

Como pode ser observado na Figura 4.2, a etapa da *Validação*, detalhada na Figura 4.10, é dividida em 2 subetapas, *Decisão* e *Otimização*, que são executadas iterativamente a cada rodada até que um critério de parada. Para entendimento preliminar, na subetapa *Otimização* é realizada a otimização dos hiperparâmetros, em busca de quais cenários são melhores discriminados, dado um contexto. Na subetapa *Decisão*, os cenários avaliados são filtrados, preservando apenas os melhores (que atendem a taxa de acurácia mínima). Os cenários não validados são eliminados das próximas rodadas.

4. Validação

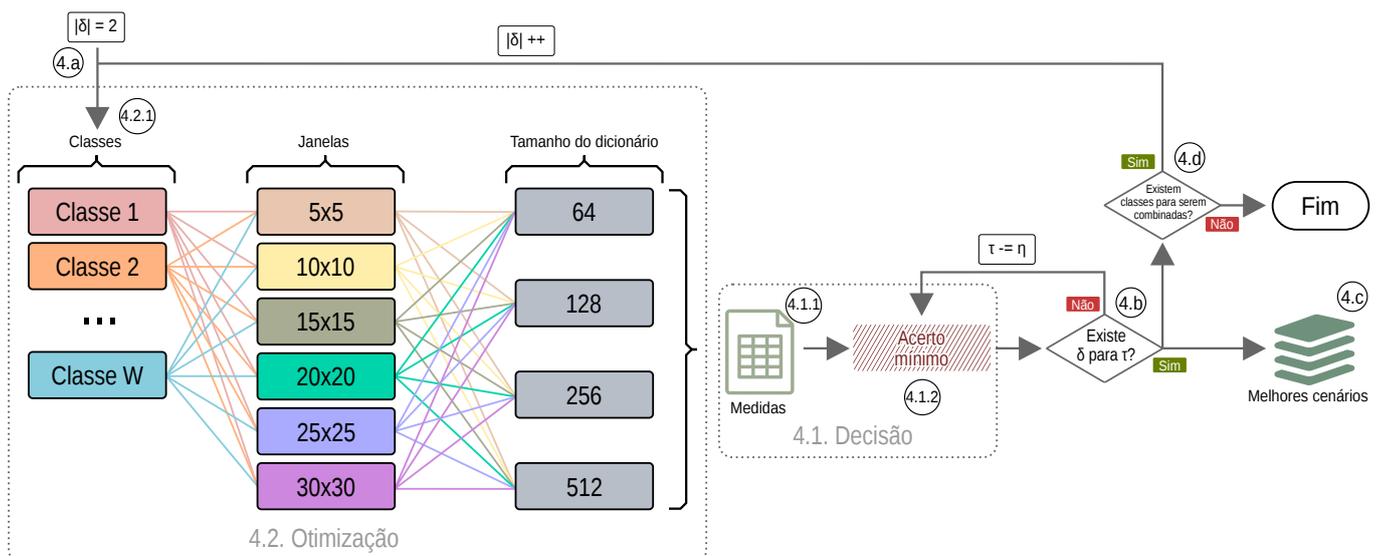


Figura 4.10: Etapa de validação, integrando as subetapas iterativas de decisão e otimização, para determinar os melhores parâmetros de cada classe.

O processo se inicia no *passo 4.a* da Figura 4.10, montando os cenários em pares de classes ($|\delta| = 2$), combinadas com todos os hiperparâmetros. No *passo 4.1.1*, as medidas de complexidade das versões quantizadas das imagens, calculadas no *passo 3.1* da etapa *Mensuração* (Figura 4.5), são carregadas para os cenários. No *passo 4.1.2*, as imagens rotuladas que pertencem às classes combinadas nos cenários são classificadas. No *passo 4.b*, caso sejam encontrados cenários válidos δ na rodada que satisfaçam τ , estes são armazenados em um arquivo externo (*passo 4.c*); caso contrário, subtrai η de τ ⁴ e os cenários são novamente avaliados até que a condição do *passo 4.b* seja verdadeira. Caso o *passo 4.b* seja verdadeiro, o algoritmo continua a execução para o *passo 4.d*.

No *passo 4.d*, se a quantidade de classes validadas na rodada for maior ou igual ao tamanho dos cenários da próxima rodada ($|\delta|$), é iniciada uma nova rodada para ampliação dos cenários ($|\delta|++$). Caso contrário, a validação é encerrada com cenários menores que o contexto ($|\delta| < W$). Assim, enquanto o tamanho dos cenários avaliados não atingir a quantidade de classes do contexto, as classes e parâmetros, previamente selecionados em rodadas anteriores, retroalimentam o processo de validação para a próxima rodada. Nessa retroalimentação, no *passo 4.2.1*, as classes e seus respectivos parâmetros não validados na rodada anterior são descartados das próximas iterações. Essa otimização permite o algoritmo ser executado em um tempo razoável, evitando processar todo o espaço de busca.

A etapa de *Validação* foi computacionalmente melhorada, observando que as classes e seus respectivos hiperparâmetros de uma rodada subsequente formam um subconjunto da rodada anterior nos termos de

$$\phi_w \subseteq \phi_{w-1}, \text{ em que } h \in \phi_w \rightarrow h \in \phi_{w-1} \quad (4-4)$$

e

$$\phi_{w-2} - \phi_{w-1} \cap \phi_w = \emptyset, \quad (4-5)$$

onde ϕ denota o conjunto de combinações de todas as classes e parâmetros válidos de uma rodada com w classes em tuplas h ; h é uma tupla composta pelo rótulo de uma classe e seus respectivos parâmetros de treinamento. Por exemplo, as classes e parâmetros descartados pelos cenários de 4 classes não têm possibilidade de comporem cenários válidos na próxima rodada com 5 classes. Os parâmetros eliminados nas rodadas com cenários de 2, 3, 4 e 5 classes não podem ser encontrados nos cenários de 6 classes,

⁴O passo de acurácia η estabelece a sensibilidade, tanto da taxa de acurácia quanto do tamanho máximo dos cenários válidos ao final da *Validação*. Passos de acurácia maiores reduzem rapidamente as taxas de acurácia a cada decréscimo e, portanto, propagam mais combinações de classes e parâmetros válidas para as próximas rodadas. Passos de acurácia menores permitem a manutenção da taxa de acurácia próxima do originalmente estabelecido, porém, forçam a convergência precoce da validação com cenários menores que o contexto analisado.

e assim sucessivamente. Essa estratégia permite que a cada nova rodada as classes e seus respectivos parâmetros não selecionados na anterior sejam descartados, evitando comparações desnecessárias.

Adotando essa estratégia de otimização, à medida que o RPA aumenta o tamanho do cenário avaliado, a cobertura do contexto diminui. A Figura 4.11 apresenta um comparativo entre o número de possíveis cenários para cada rodada de um contexto de 12 classes, 6 tamanhos de janelas e 4 tamanhos de dicionário, avaliados sem otimização e com otimização, em escala logarítmica. Com otimização, a partir dos cenários com 5 classes, o número de possíveis classes e parâmetros com taxa de discriminabilidade igual ou superior à taxa de acurácia. Isso faz com que algoritmo convirja para cenários maiores e mais restritos em razão da restrição da taxa mínima de acurácia. Apesar da otimização evitar a busca em todo o espaço, ela garante o ótimo global por meio da eliminação de classes e parâmetros com baixa acurácia.

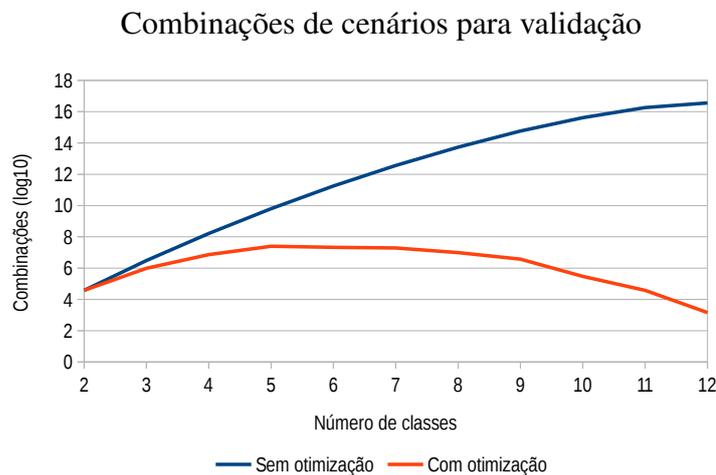


Figura 4.11: Número de cenários avaliados com otimização e sem otimização para o contexto de 12 classes.

4.5 Módulo de Classificação

Nessa seção é descrito o funcionamento do *Módulo de Classificação*. As etapas são análogas às anteriormente descritas, com exceção de utilizar apenas os melhores parâmetros obtidos na *Validação* para quantização das imagens de testes.

4.5.1 Quantização de Teste e Decisão

O *Módulo de Classificação* é formado pelas 2 últimas etapas do RPA, apresentadas na Figura 4.2, de *Quantização* e *Decisão*. A Figura 4.12 detalha essas etapas. Analogamente à etapa de *Quantização* do *Módulo de Treinamento*, descrita na Seção 4.4.2, as

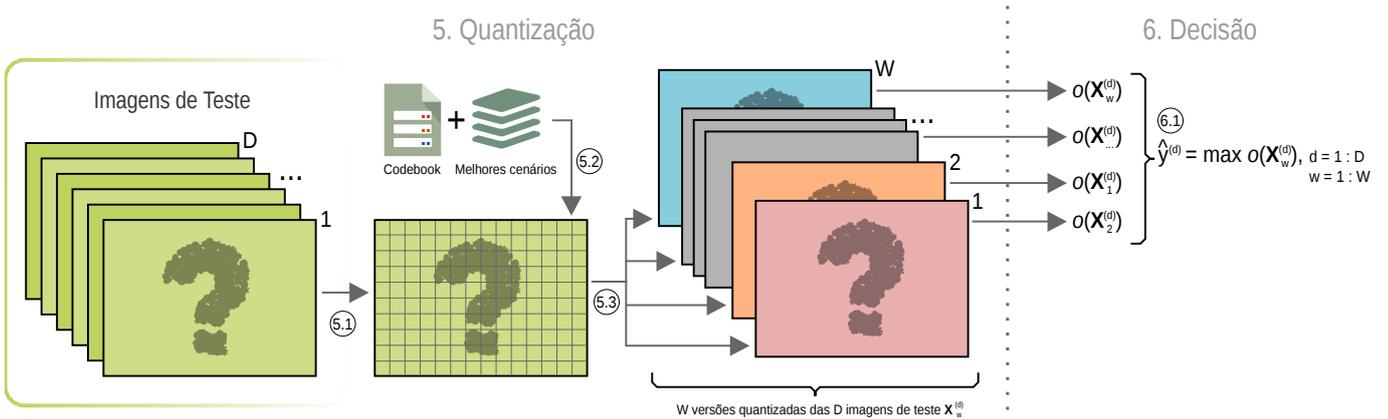


Figura 4.12: Modelo de classificação das imagens de teste descrito pelas etapas de quantização e decisão.

imagens passam pelo mesmo processo de quantização. Entretanto, no *passo 5.1*, as D imagens de entrada do algoritmo são do conjunto de teste e, no *passo 5.2*, os parâmetros do *codebook* são filtrados pelos melhores cenários identificados na validação. Nesse passo, o RPA utiliza as sugestões de melhores parâmetros que maximizam a acurácia da classificação, quantizando as imagens de teste no *passo 5.3*. Por fim, no *passo 6.1* a decisão da classificação é tomada utilizando a equação

$$\hat{y}^{(d)} = \max_{w=1:W} o(\mathbf{X}_w^{(d)}), \quad d = 1 : D, w = 1 : W \quad (4-6)$$

onde $\hat{y}^{(d)}$ denota a classe predita da d -ésima imagem de teste e $o(\mathbf{X}_w^{(d)})$ é o valor da medida de complexidade da w -ésima versão quantizada da d -ésima imagem de teste $\mathbf{X}^{(d)}$.

4.5.2 Implementação e Código-fonte

O PRA foi implementado na linguagem Python 2.7 e está disponível para download em <https://github.com/rdfffeitosa/prr>. Além do código-fonte, no repositório é possível acessar o *dataset* utilizado nos experimentos e modelos treinados nas 3 partições dos conjuntos de treinamento e teste.

Parametrização e desempenho do classificador de cenas

Esse capítulo descreve os experimentos realizados com método de reconhecimento de cenas descrito no Capítulo 4. Os aspectos quantitativos e qualitativos dos parâmetros do RPA e do experimento são discutidos preliminarmente. Em seguida, é analisada a influência de cada um desses parâmetros, tanto no treinamento quanto na classificação. Por fim, são reportados e discutidos dos resultados de desempenho do RPA sob diversos cenários experimentais utilizando as métricas de acurácia, *precision*, *recall* e *F1 score*.

5.1 Experimentos

Os resultados são apresentados e discutidos nessa seção seguindo as variações de parâmetros utilizadas nos experimentos. Foram testadas diferentes proporções de particionamento dos dados de treinamento e teste para avaliar a capacidade do RPA em generalizar a aprendizagem. Os treinamentos foram realizados com diversos valores de convergência, explorando a sensibilidade do método em relação à qualidade dos modelos gerados para representação de cada classe. Também foi investigada a relação entre os tamanhos das janelas e dos dicionários obtidos no treinamento e as taxas de acurácia alcançadas.

5.1.1 Recursos Computacionais

Os módulos de treinamento e classificação foram implementados na linguagem Python versão 2.7. Os experimentos foram executados em 2 computadores, ambos com o sistema operacional Linux Ubuntu 16.04 com as seguintes configurações: computador *Ryzen*, com processador AMD® Ryzen™ 7 1800X 3.6GHz de 8 núcleos (16 threads) e 32GB de memória RAM; computador *i7*, com processador Intel® Core™ i7-2600K 3.4GHz de 4 núcleos (8 threads) e 32GB de memória RAM.

5.1.2 Parametrização do Método

Nessa subseção são descritas as variações dos parâmetros configurados no RPA para execução dos experimentos. As discussões dos resultados serão realizadas de maneira integrada, sem prejuízo às observações pontuais sobre os efeitos de cada um dos parâmetros no desempenho do classificador.

Particionamento dos Dados

Para avaliar a capacidade o RPA de generalizar a aprendizagem do treinamento, sob diferentes disponibilidades de amostras, a base de dados utilizada foi dividida entre conjunto de treinamento e conjunto de teste em 3 proporções de partição. Na primeira partição (A), foram selecionadas aleatoriamente 50 imagens para treinamento e 50 imagens para teste de cada classe. Na segunda partição (B), o conjunto de treinamento foi reduzido para 25 imagens e o conjunto de teste ampliado para 75 imagens. Na terceira partição (C), apenas 15 imagens de cada classe foram utilizadas para treinar o modelo e as 85 restantes para teste. Utilizou-se o método *repeated holdout cross-validation* com $k = 10$ para geração dos conjuntos de treinamento e teste de cada partição.

Convergência de Treinamento ϵ

Na etapa de *Prototipação*, descrita na Subseção 4.4.1, as janelas das imagens de treinamento são submetidas ao algoritmo LBG, para calcular os *codevectors* que descrevem as classes. Como discutido na Subseção 2.3.5, essas amostras são projetadas em um espaço U -dimensional, onde $U = M \times N$, M e N são, respectivamente, altura e largura das janelas. Esse espaço é recursivamente segmentado em regiões, ilustradas pelo Diagrama de Voronoi apresentado na Figura 2.4, e os centroides dessas regiões são ajustados até que o erro seja menor ou igual a ϵ , definido como *taxa de convergência*. O erro é calculado pela diferença entre as distorções médias das sucessivas iterações, obtidas pelas distâncias entre os vetores de treinamento e os seus respectivos centroides. Os valores testados para ϵ foram [0,1; 0,01; 0,001; 0,0001; 0,00001]. Valores maiores de ϵ permitem convergências mais rápidas e maiores distorções entre as amostras de entrada e os *codevectors*. Valores menores aumentam consideravelmente o tempo de treinamento, porém fornecem centroides distribuídos mais homogeneamente no espaço.

Tamanhos da Janela e do Dicionário

A escolha dos tamanhos das janelas e dos dicionários está relacionada ao contexto do problema a ser resolvido. Esses parâmetros devem ser definidos observando o conteúdo da informação de cada classe. Classes compostas por cenas com informação

visual mais fina tendem a ser melhor representadas com janelas menores, pois causam menos perda de informação relevante no processo de quantização. Ao contrário, classes com menos informação quantitativa presente nas cenas ou regiões mais homogêneas podem ser representadas com janelas maiores, sem prejuízo ao conteúdo da informação relevante. Detalhes adicionais a respeito dessa escolha serão discutidos adiante.

Embora a etapa de validação confira ao RPA uma estratégia adaptativa de seleção dos melhores parâmetros, a definição dos conjuntos de janelas e símbolos deve ser realizada em função do tempo disponível para o módulo de treinamento. Muitos valores para cada um desses parâmetros implica o aumento expressivo de cenários para serem avaliados. Nos experimentos discutidos nesse capítulo, foram utilizadas janelas quadradas de 5, 10, 15, 20, 25 e 30 pixels e dicionários com 64, 128, 256 e 512 símbolos.

5.2 Resultados e Discussão

5.2.1 Particionamento dos Dados e Taxas de Convergência

A Figura 5.1 mostra o tempo decorrido (em minutos) para a etapa de prototipação das amostras de treinamento nas partições A (Figura 5.1(a)), B (Figura 5.1(a)) e C (Figura 5.1(a)) em todos os tamanhos de janelas e taxas de convergência ϵ . As partições A e C foram prototipadas no *Ryzen* e a partição B no *i7*, o que explica o maior tempo de prototipação mesmo com menos dados. Os resultados mostram diminuição no tempo de execução à medida que o tamanho da janela é aumentado. O aumento da dimensionalidade dos vetores de treinamento e, conseqüentemente, a diminuição da quantidade de amostras de treinamento, favorece tempos menores para a convergência da prototipação em todos os valores de ϵ . A maior amplitude de redução de tempo é observado para taxas de convergência ϵ 0,0001 e 0,00001, principalmente em relação às janelas 5x5, 10x10 e 15x15. Isso significa que o ajuste dos centroides é computacionalmente mais oneroso para janelas menores. À medida que o tamanho da janela aumenta, gradualmente, em

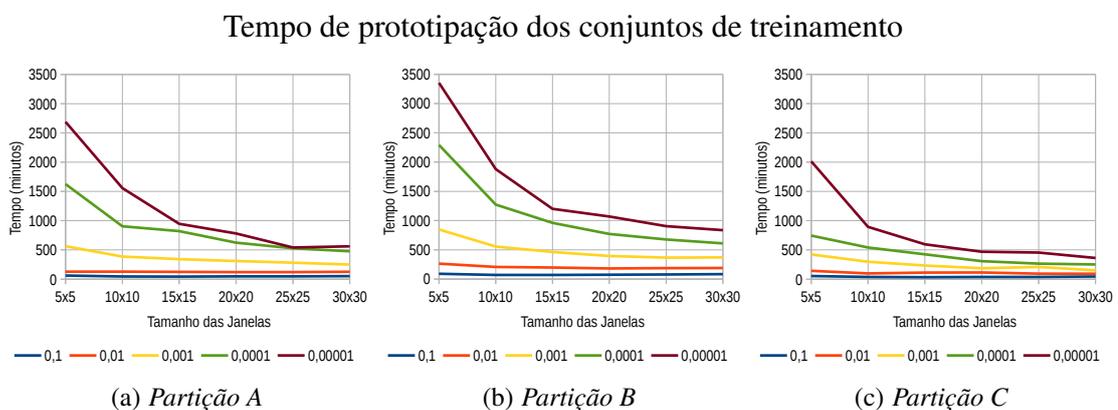


Figura 5.1: Tempo de prototipação das amostras de treinamento (em minutos) para as partições A (a), B (a), C (a) de acordo com as taxas de convergência ϵ .

5 pixels tanto na largura quanto na altura, a curva de decrescimento do tempo tende a suavizar. Assim, para a maior janela definida em parâmetro (30x30), a diferença entre o tempo de execução do algoritmo LBG para todos os valores de ϵ atinge o menor valor.

Para aferição do desempenho do RPA, cada partição dos dados (A, B e C) foi dividida, aleatoriamente, 10 vezes em conjuntos de treinamento e teste utilizando o método *repeated holdout cross-validation*. As Tabelas 5.1 e 5.2 resumizam os percentuais de interseção entre as amostras dos conjuntos de treinamento e teste, respectivamente, considerando a combinação de 2 a 10 *holdouts*. Na Tabela 5.1 pode ser observado que a interseção total entre as amostras foi de 0,67% para a partição A. Isso indica que o processo de seleção das amostras em 10 *holdouts* manteve a diversidade dos dados e a aleatoriedade na escolha das amostras. Observa-se a total separação das amostras na partição B a partir da combinação de 4 *holdouts* e na partição C a partir de 3 *holdouts*. Em comparação com as demais partições, os valores da Tabela 5.2 mostram grande interseção de amostras com todos os *holdouts* na partição C (23,53%).

Conforme discutido anteriormente, a taxa de convergência ϵ da prototipação dos *codevectors* estabelece o limite de ajuste das distorções nas partições. A cada iteração

Tabela 5.1: Percentuais de interseção entre os *holdouts* dos conjuntos de treinamento, considerando valores máximos, médios e mínimos.

#Holdout	Partição A			Partição B			Partição C		
	Máxima	Mínima	Média	Máxima	Mínima	Média	Máxima	Mínima	Média
2	53,50%	46,83%	50,30%	30,67%	20,67%	25,29%	21,67%	11,11%	15,05%
3	29,33%	22,33%	25,48%	10,33%	3,00%	6,48%	6,11%	0,00%	2,25%
4	16,50%	9,67%	13,04%	4,33%	0,00%	1,68%	1,67%	0,00%	0,34%
5	9,33%	4,67%	6,81%	2,00%	0,00%	0,44%	0,56%	0,00%	0,05%
6	5,67%	2,17%	3,68%	0,67%	0,00%	0,11%	0,56%	0,00%	0,01%
7	3,17%	1,17%	2,12%	0,33%	0,00%	0,02%	0,00%	0,00%	0,00%
8	2,00%	0,83%	1,33%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
9	1,17%	0,67%	0,92%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
10	0,67%	0,67%	0,67%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%

Tabela 5.2: Percentuais de interseção entre os *holdouts* dos conjuntos de teste, considerando valores máximos, médios e mínimos.

#Holdout	Partição A			Partição B			Partição C		
	Máxima	Mínima	Média	Máxima	Mínima	Média	Máxima	Mínima	Média
2	53,50%	46,83%	50,30%	76,89%	73,56%	75,10%	86,18%	84,31%	85,01%
3	29,67%	21,00%	25,41%	59,67%	53,67%	56,46%	73,73%	70,78%	72,28%
4	17,50%	9,83%	12,91%	46,00%	39,22%	42,50%	63,33%	59,80%	61,46%
5	10,33%	4,17%	6,60%	35,33%	28,44%	32,02%	54,02%	50,10%	52,28%
6	5,67%	1,67%	3,39%	27,33%	21,00%	24,15%	46,47%	42,16%	44,49%
7	3,17%	0,67%	1,74%	20,89%	15,89%	18,22%	39,41%	35,69%	37,89%
8	1,83%	0,17%	0,88%	16,00%	11,89%	13,74%	33,63%	30,69%	32,29%
9	0,83%	0,17%	0,42%	11,11%	9,11%	10,36%	28,43%	26,86%	27,55%
10	0,17%	0,17%	0,17%	7,78%	7,78%	7,78%	23,53%	23,53%	23,53%

do LBG, as distorções são calculadas por meio da distância euclidiana quadrática média, entre os vetores de treinamento e seus respectivos centroides. Quando a diferença entre a distorção média da iteração atual e da anterior é menor ou igual à taxa de convergência ϵ , o algoritmo interrompe a execução. A distorção média da última iteração, para cada combinação de parâmetros de treinamento e classes, são apresentadas em suas respectivas tabelas para os particionamentos A (Tabela B.1), B (Tabela B.2) e C (Tabela B.3). As colunas ψ_1 e ψ_2 dessas tabelas apresentam as médias das distorções entre as classes, por taxas de convergência ϵ (ψ_1) e por tamanho do dicionário (ψ_2). Pode-se observar, para todas as partições e todos os tamanhos de janelas, que o valor de ψ_2 decresce a medida que são criados novos *codevectors*. Entretanto, nota-se que em alguns casos ψ_1 aumenta com a diminuição da taxa de convergência de 0,0001 para 0,00001.

Esses valores aumentados são identificados nas Tabelas B.1, B.2 e B.3 com asterisco (*). Esses aumentos podem ser considerados irrelevantes pois estão entre 0,0028% e 1,1285% para a partição A (Tabela B.1), 0,0049% e 1,7764% para a partição B (Tabela B.2) e 0,0016% e 3,5269% para a partição C (Tabela B.3). Esse resultado indica que quanto menor o conjunto de treinamento, mais e maiores aumentos da distorção média ocorrem, mesmo diminuindo o limite de erro aceito entre as iterações.

A Figura 5.2 apresenta uma visão geral das distorções médias de cada conjunto de treinamento. Pode ser observada a diminuição, quase linear, da distorção média geral em função do tamanho do *codebook* (ψ_2) para todos tamanhos de janela. Como esperado, as distorções médias, para todas as dimensões de janelamento, se aproximam a medida que mais *codevectors* são criados. A cada aumento da janela, a taxa de escala de aumento diminui progressivamente. Por exemplo, a janela 10x10 é 4x maior que a 5x5; a 15x15 é 2,25x maior que a 10x10; a 20x20 é 1,77x maior que a 15x15, e assim por diante. A proporção entre as distorções médias gerais diminui de forma semelhante em todos os tamanhos de dicionário. Portanto, o tamanho das janelas é proporcional à distorção da prototipação. Quanto maior o janelamento das amostras, menos vetores de treinamento são projetados no espaço e maior sua dimensionalidade. Quanto maior a dimensionalidade

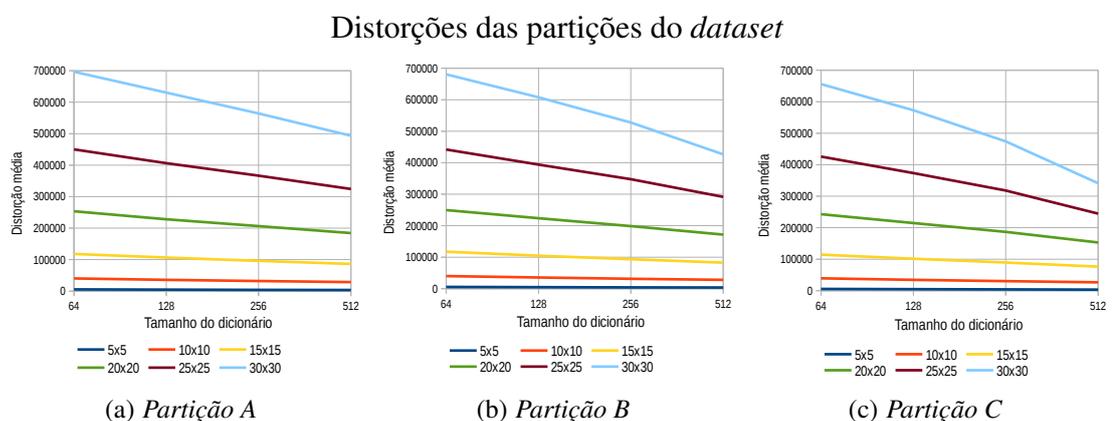


Figura 5.2: Distorção média entre os vetores de treinamento em os *codevectors* para dicionários com 64, 128, 256 e 512 símbolos para as partições A (a), B (b) e C (c).

do espaço, mais próximos os vetores de treinamento tendem a ser [13, 3, 46].

O volume de amostras é consideravelmente diferente em cada uma das 3 partições, com redução de 50% do particionamento A para o B e de 40% do B para C. Entretanto, observa-se na Figura 5.2 que as distorções médias são semelhantes em todos os conjuntos de treinamento. Então, pode-se inferir que, em termos qualitativos, a prototipação não é influenciada pela quantidade de amostras de treinamento. Considerando que o treinamento é realizado de forma independente para cada classe, esse resultado sugere que o RPA é capaz de lidar com conjuntos de treinamento desbalanceados. As Tabelas B.4, B.5, B.6, B.7 e B.8 detalham o distanciamento médio dos *codebooks* gerados na prototipação, comparando as partições A, B e C entre si. Nessas tabelas, a coluna *Absoluta* é calculada obtendo-se a média dos centroides dos *codebooks* de todas as classes, em seguida é determinada a distância euclidiana entre esses pontos. Essa escala absoluta não é adequada para determinar se os *codebooks* são próximos, pois a dimensionalidade do espaço influencia a noção de distância. Portanto, na coluna *Relativa* são apresentadas as distâncias normalizadas em função da amplitude do espaço. As distâncias relativas consideram a maior distância possível dentro do espaço e determinam, em termos percentuais, o quanto os *codebooks* se diferenciam, na média. Os resultados mostram que os *codebooks* se distanciam à medida que o tamanho da janela é aumentado, permanecendo estável para a quantidade de símbolos. As menores diferenças são observadas entre as partições A e B e as maiores entre as partições A e C, para todas as taxas de convergência ϵ . Os distanciamentos variam entre 0,6%, utilizando $\epsilon = 0,0001$ (Tabela B.7), e 2,15% utilizando $\epsilon = 0,00001$ (Tabela B.8). Essas pequenas variações percentuais corroboram os resultados apresentados na Figura 5.2. Indicam que a quantidade de imagens de treinamento não influencia decisivamente a qualidade dos *codebooks* gerados na etapa de *Prototipação* (Figura 4.2).

Outra perspectiva de análise da influência do tamanho do conjunto de treinamento, considera métricas objetivas de qualidade, que avaliam a similaridade de imagens reconstruídas após algum processo de compressão. Nesse caso a compressão é realizada pela quantização vetorial aplicada nas etapas de *Quantização* (Figura 4.2). As medições foram realizadas para cada taxa de convergência ϵ e são apresentadas nas Tabelas B.9 ($\epsilon = 0,1$), B.10 ($\epsilon = 0,01$), B.11 ($\epsilon = 0,001$), B.12 ($\epsilon = 0,0001$) e B.13 ($\epsilon = 0,00001$). As medidas objetivas de qualidade utilizadas foram: MSE^1 e $PSNR^2$ por serem escolhas populares na literatura [148], $SSIM^3$ devido à sua performance [148] e a Informação Vi-

¹Medida não normalizada. Valores maiores indicam baixa qualidade na reconstrução.

²Medida não normalizada e em escala logarítmica. Reconstruções de maior qualidade apresentam valores maiores.

³Medida normalizada. Quanto mais próximo da imagem de referência, maior o valor.

sual Fidelidade (VIF^4 - Visual Information Fidelity) [145] se baseia em estatísticas de cenas naturais e a noção de interpretação da informação pelo sistema visual humano. As imagens foram reconstruídas utilizando a quantização vetorial e separadas em 2 grupos: *quantização convergente* se quantizadas com o *codebook* correspondente à sua própria classe e *quantização divergente* se quantizadas com os *codebooks* das demais classes.

Como esperado, em todos os casos, as imagens quantizadas com classe divergente apresentaram menor qualidade. Tanto o *PSNR* quanto a *VIF* se mostraram muito sensíveis à variação do tamanho do janelamento. A *VIF* apresentou valores se aproximando de 0 para janelas 30x30 com 64 símbolos em todas as taxas de convergência ϵ . As melhores condições de qualidade são observadas para janelas 5x5 com 512 símbolos. As maiores diferenças entre os valores da *quantização convergente* e da *quantização divergente* foram obtidas com a *VIF*. As diferenças médias entre os tipos de quantização de cada medida de qualidade, em função das taxas de convergência ϵ , são sumarizadas na Figura 5.3. Pode-se observar que a maior diferença, à medida que a taxa de convergência ϵ diminui, foi obtida pelo *MSE* (Figura 5.3(a)) e a menor pela *SSIM* (Figura 5.3(b)).

As medidas *MSE*, *SSIM* e *VIF* tendem a distanciar as imagens da *quantização convergente* e da *quantização divergente* à medida que a taxa de convergência ϵ diminui. Entretanto, o *PSNR* para $\epsilon = 0,1$ apresentou a maior diferença, o menor valor para $\epsilon = 0,01$

Diferenças entre as reconstruções das quantizações convergente e divergente

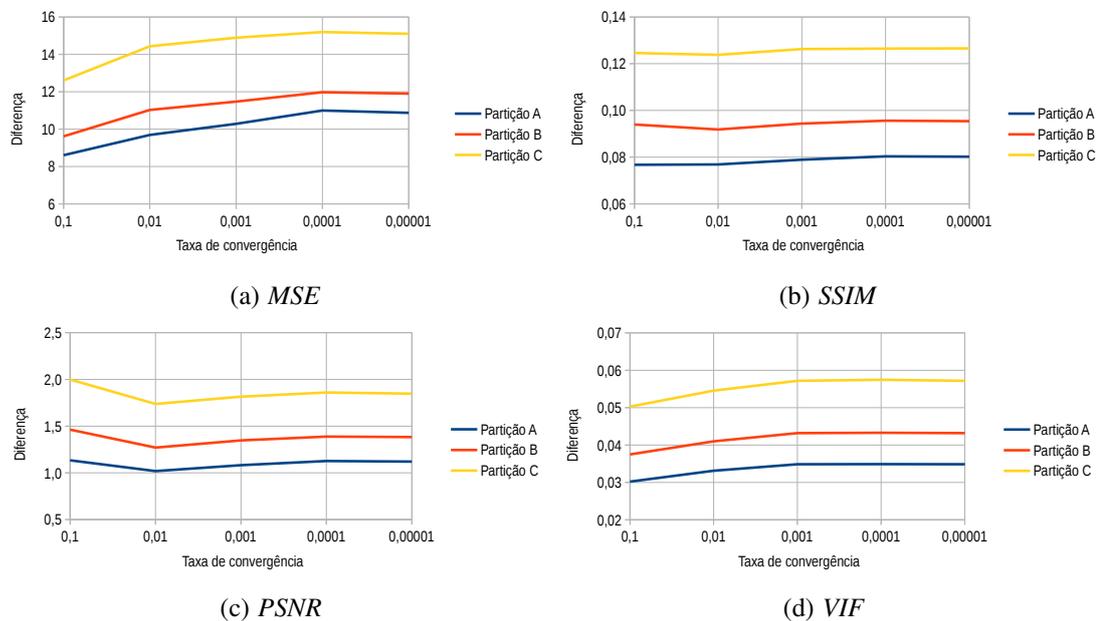


Figura 5.3: Variação média da diferença das medidas de qualidade *MSE* (a), *SSIM* (b), *PSNR* (c) e *VIF* (d) entre as imagens quantizadas para as classes verdadeiras e para as classes falsas.

⁴Medida normalizada. Imagens muito semelhantes tendem a fornecer valores próximos a 1 e imagens diferentes próximos a 0.

e voltou a crescer para as demais taxas de convergência (Figura 5.3(c)). Sob a perspectiva das partições de treinamento e teste, na Figura 5.3 pode ser observado que as partições A e B apresentam maior proximidade, com as maiores diferenças obtidas entre partições B e C. Sabendo que o tamanho do conjunto de treinamento tem pouca influência na qualidade da prototipação e, conseqüentemente, das reconstruções na quantização, atribui-se esse resultado à escolha arbitrária das imagens.

A Figura 5.4 apresenta o resultado das quantizações de uma cena da classe *praia*, utilizando *codebooks* convergentes com taxas de convergência $\epsilon = 0,1$ (Figura 5.4(a)), $0,001$ (Figura 5.4(b)) e $0,00001$ (Figura 5.4(c)). Os parâmetros de tamanho da janela e quantidade de símbolos foram 5×5 e 512, respectivamente, foram escolhidos para maximizar possíveis diferenças na quantização utilizando diferentes valores de ϵ . As imagens apresentam cada ponto mapeado em *colormap* tipo *viridis* da biblioteca *Matplotlib* com a escala de referência de 0 a 511 na Figura 5.4(c). Pode-se observar valores mais altos para os pixels e menos contraste à medida que a taxa de convergência ϵ é diminuída. Entretanto, como o RPA não está preocupado com valores isolados da quantização, mas com a seqüência ou a distribuição dos *codevectors* na cena, isso não influencia na classificação.

Considerando os resultados apresentados nas Figuras 5.1 e 5.3, no caso de pouca disponibilidade de recursos computacionais, avalia-se que a taxa de convergência $\epsilon = 0,001$ é adequada, tanto para redução do tempo de treinamento quanto para manutenção da qualidade da prototipação. A Figura 5.5 detalha as diferenças, entre as quantizações convergente e divergente, das medidas de qualidade para $\epsilon = 0,001$. O *MSE* (Figuras 5.5(a), 5.5(b) e 5.5(c)) para dicionários com 64 símbolos apresenta crescimento irregular da diferença. À medida que os dicionários aumentam de tamanho, o crescimento da diferença se regulariza em função do aumento da janela. Na *SSIM* (Figuras 5.5(d), 5.5(e) e 5.5(f)) a diferença cresce para janelas maiores com mesmo tamanho de dicionário. O crescimento da diferença tende a ser maior com o aumento da quantidade de símbolos. No *PSNR* (Figuras 5.5(g), 5.5(h) e 5.5(i)) a diferença média de qualidade entre os tipos de quantização para $[5 \times 5, 512]$ e para $[30 \times 30, 64]$ apresenta valores semelhantes, que tendem a aumentar para janelas e dicionários maiores. Na *VIF* (Figuras 5.5(j), 5.5(k) e

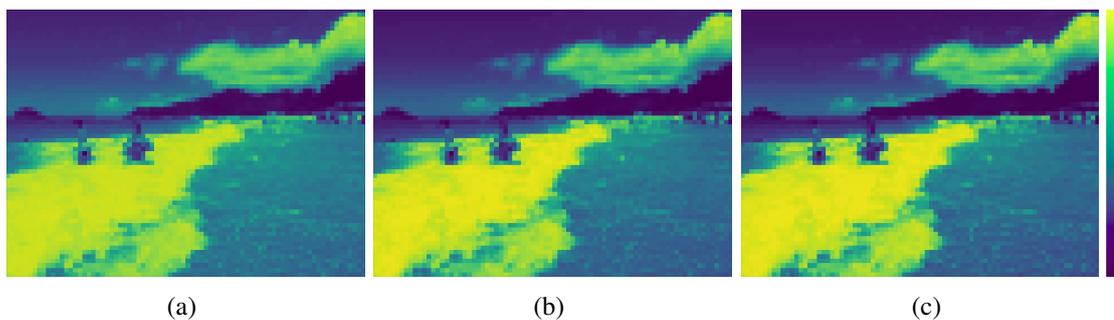


Figura 5.4: Exemplo de uma cena da classe *praia* quantizada com *codebook* convergente, janela 5×5 e 512 símbolos, da partição A para $\epsilon = 0,1$ ((a)), $\epsilon = 0,001$ ((b)) e $\epsilon = 0,00001$ ((c)).

5.5(I)) a diferença diminui para dicionários menores à medida que as janelas aumentam. Entretanto, em dicionários maiores, esse comportamento inverte e a diferença passa a crescer para janelas maiores. Portanto, em razão das maiores diferenças, a medida *VIF* se mostra capaz de determinar de forma mais precisa a variação da qualidade entre as cenas da *quantização convergente* e da *quantização divergente*. Esse resultado é compatível com a característica da *VIF* apresenta em [145], que se baseia em estatísticas de cenas naturais e a noção de interpretação da informação pelo sistema visual humano [145]. Assim, o processo de quantização simula a visão humana na percepção das características

Diferenças das reconstruções das quantizações convergente e divergente calculadas pelas medidas objetivas de qualidade

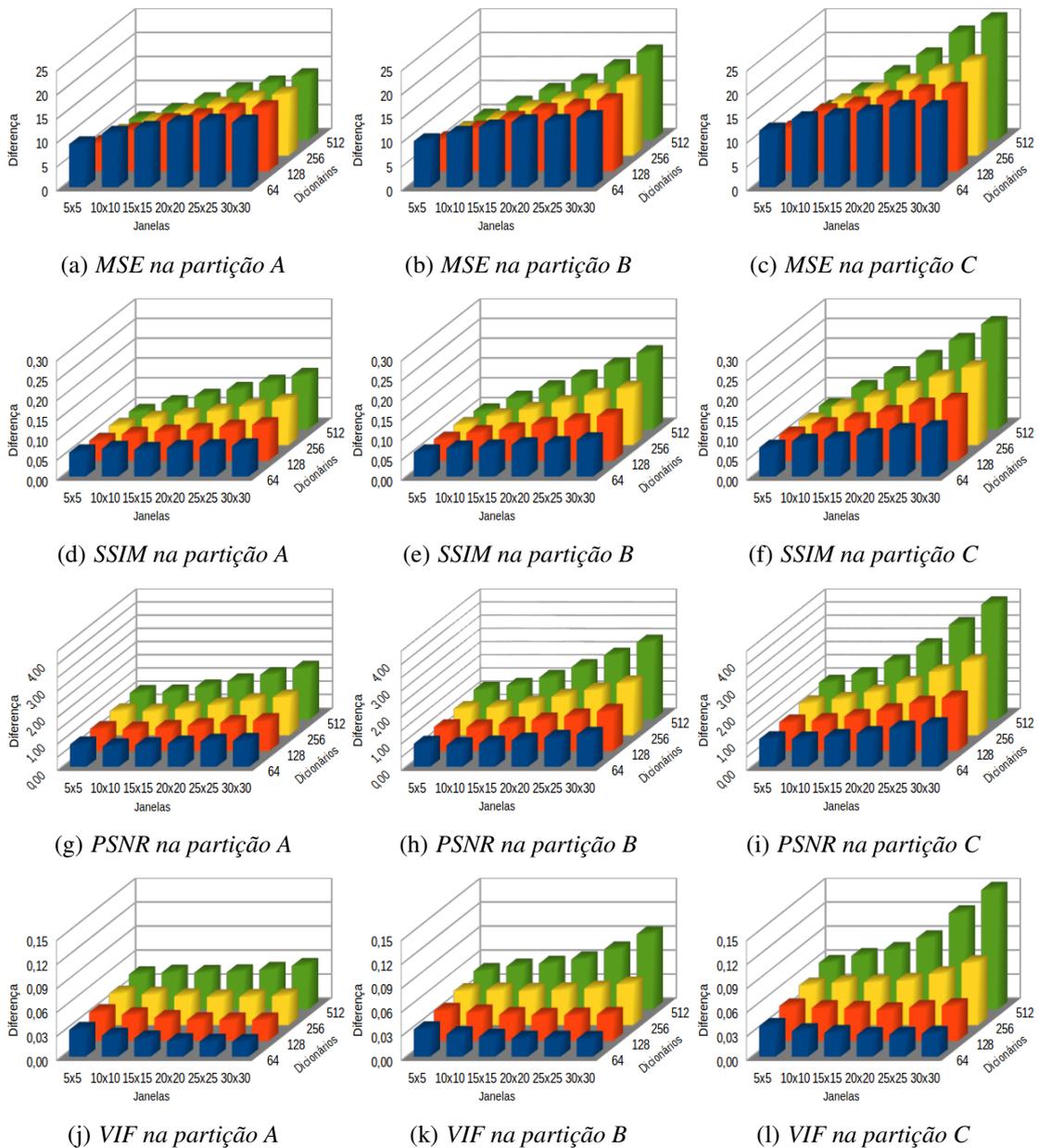
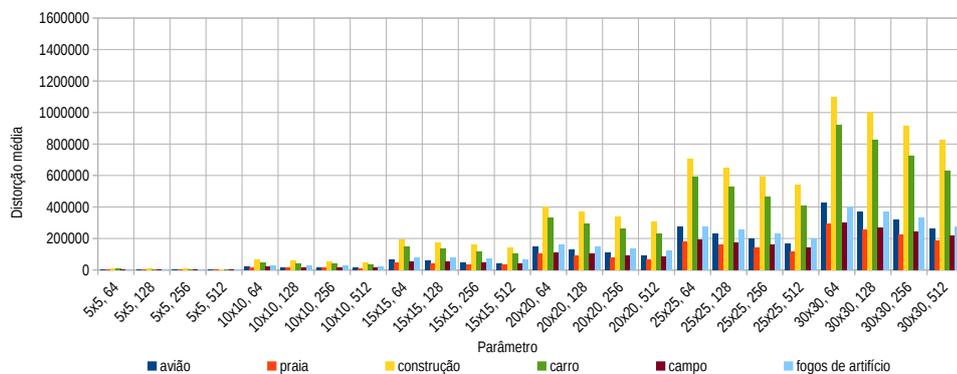


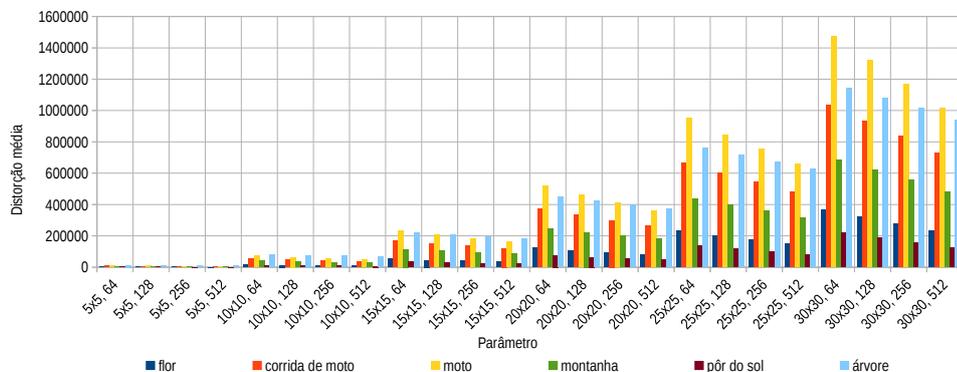
Figura 5.5: Detalhamento das diferenças das medidas de qualidade entre as imagens de treinamento quantizadas com suas respectivas classes convergentes e as imagens quantizadas com classes divergentes com $\epsilon = 0,001$.

da cena. Em termos qualitativos, o distanciamento extraclasse é maximizado com janelas e dicionário maiores e os resultados de classificação tendem a ser melhores.

As Figuras 5.6 (partição A), 5.7 (partição B), 5.8 (partição C) apresentam as distorções de cada classe, para todos os tamanhos de janelas e dicionário. As classes que tiveram as maiores distorções foram *árvore* e *moto* e a menor distorção *pôr do sol*. Maior distorção indica que os vetores de treinamento ficaram mais dispersos nas partições, e menor distorção que os mesmos ficaram mais concentrados, na média. Portanto, entendendo uma janela como uma unidade básica de informação visual, pode-se afirmar que cenas com regiões semelhantes tendem a transmitir menos informação visual. Nesse caso, as janelas não se diferenciam tanto umas das outras e, conseqüentemente, fornecem amostras para vetores de treinamento localizados em porções concentradas do espaço. Portanto, os *codevectors* são posicionados mais próximos dos vetores de treinamento, compondo *codebooks* com menor distorção. Um exemplo desse efeito é a classe *pôr do sol*. Cenas com detalhamento visual fino, como da classe *árvore*, fornecem amostras para treinamento que se distribuem de maneira mais heterogênea por todo o espaço de prototipação. Dessa forma, os *codevectors* obtidos se distanciam mais dos vetores de sua respectiva partição, na média, ocasionando mais distorção. O caso de alta distorção da classe *moto* será discutido adiante considerando outros fatores.



(a) Distorção média por classe da partição A



(b) Distorção média por classe da partição A (continuação)

Figura 5.6: Distorções médias das classes de acordo com os parâmetros de treinamento da partição A.

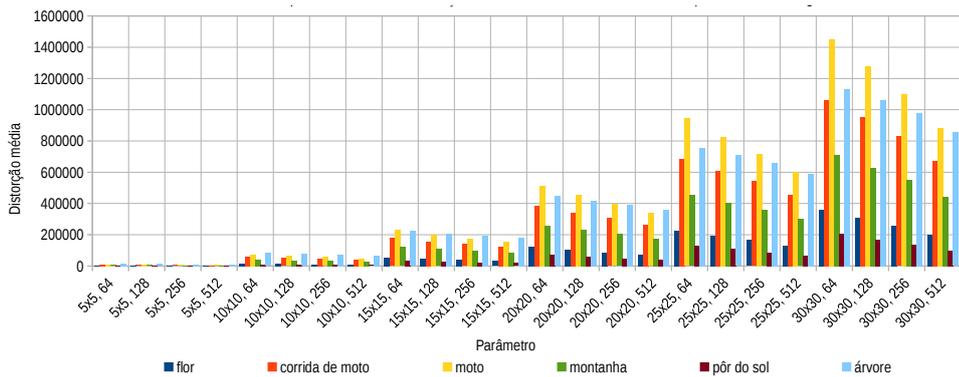
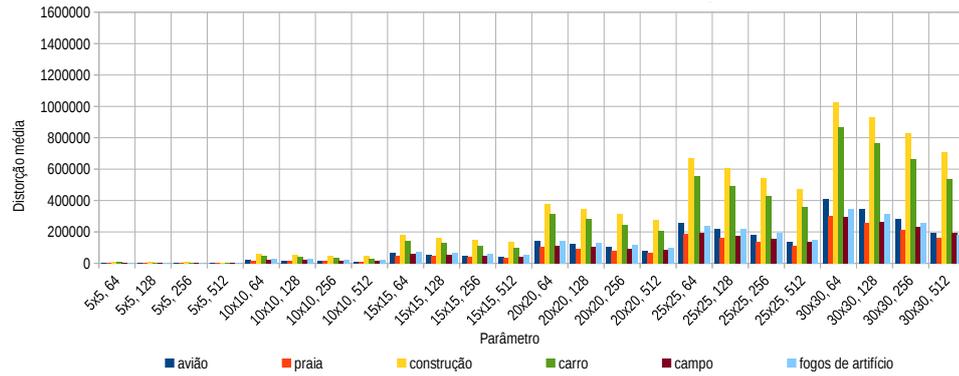


Figura 5.7: *Distorções médias das classes de acordo com os parâmetros de treinamento da partição B.*

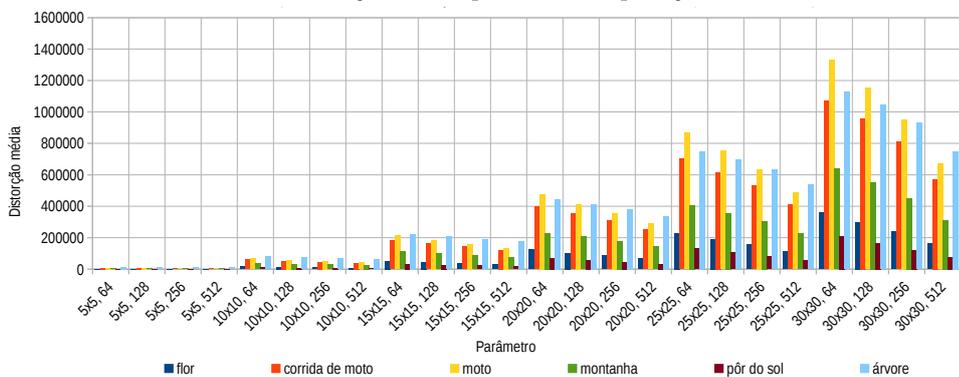
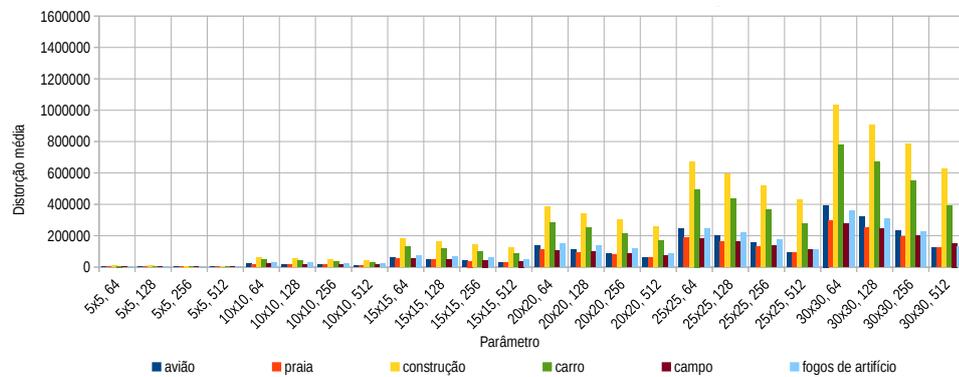


Figura 5.8: *Distorções médias das classes de acordo com os parâmetros de treinamento da partição C.*

5.2.2 Tamanho da Janela e do Dicionário

Analisadas as questões referentes às partições e taxas de convergência ϵ do treinamento, os resultados seguintes são apresentados utilizando a partição A. A escolha é justificada em razão do volume de amostras de treinamento ter representatividade para os próximos experimentos. Nas discussões seguintes utiliza-se as denominações de *melhores cenários* e *cenários máximos*. Como introduzido na Subseção 4.4.3, os *melhores cenários* representam o conjunto de combinações entre classes e parâmetros, que se discriminam mantendo a taxa mínima de acurácia na validação. Dos *melhores cenários*, pode-se extrair o subconjunto dos *cenários máximos*, que se restringe apenas aos que atingiram a maior taxa de acurácia.

A frequência dos parâmetros dos melhores cenários da partição A, são apresentados na Figura 5.9 para as medidas de compressão, a partir dos dados da Tabela 5.3, e na Figura 5.10 para a medida de entropia, a partir dos dados da Tabela 5.4 em escala reversa (valores centrais são maiores que os marginais). A escolha da partição A foi realizada em razão de ser observada maior estabilidade nas generalizações dos treinamentos, que será

Ocorrências dos parâmetros nos melhores cenários da partição A de acordo com as medidas de compressão

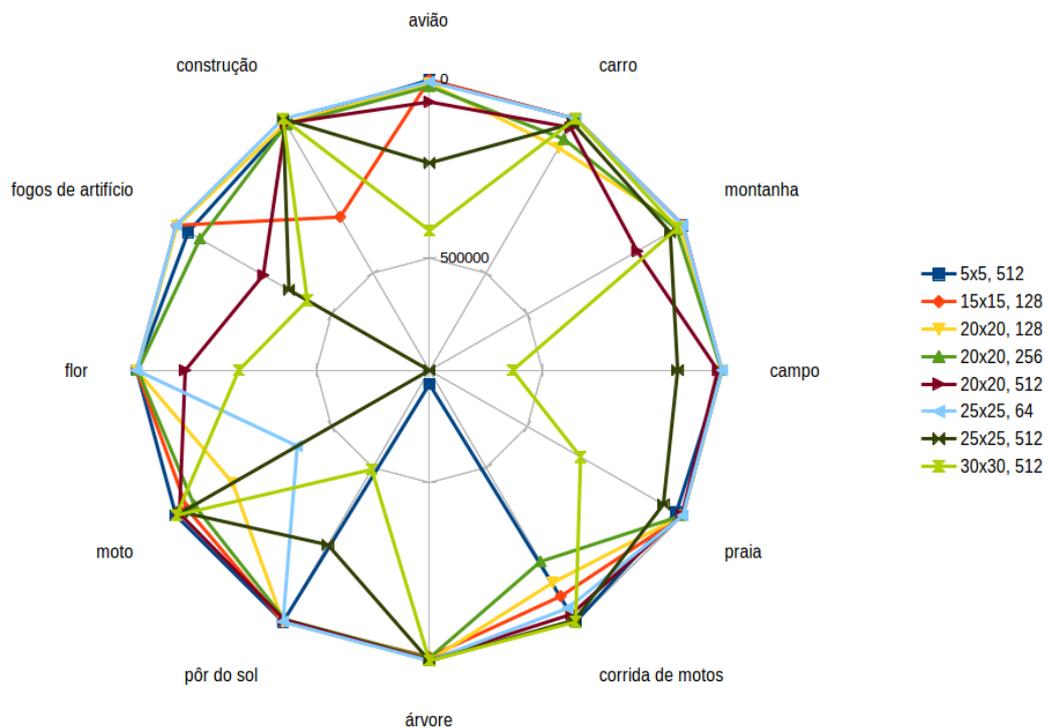


Figura 5.9: Frequência dos parâmetros de tamanhos da janela e do dicionário de cada classe, obtidos a partir dos melhores cenários na partição A e medidas de compressão.

Tabela 5.3: Ocorrências dos parâmetros observados nos melhores cenários, obtidos na validação do treinamento da partição A e medidas de compressão.

Janela	Dicionário	avião	construção	fogos de artifício	flor	moto	pôr do sol	árvore	corrida de motos	praia	campo	montanha	carro
5x5	64	0	0	3919	0	0	0	1640	0	0	0	0	0
	128	0	0	17953	0	0	0	76087	0	112	0	0	0
	256	0	4905	28164	0	0	0	296610	0	17958	0	0	0
	512	0	17198	39237	0	0	0	776164*	0	20274	2075	0	0
10x10	64	0	116714	4251	520	543	0	17563	13530	0	0	0	0
	128	0	133935	15117	33	159	41	77657	17419	0	6	0	0
	256	0	169220	41813	412	90	125	199011	38180	764	0	36	0
	512	1940	289988	86620	1368	61	1028	247931	49824	6299	3778	3015	2348
15x15	64	3122	56577	234	0	95245	0	3079	67951	0	0	0	0
	128	20	318207*	1012	46	36124	6	11116	82939	0	2	22	0
	256	5843	216143	12768	16824	33976	1650	17272	136254	416	10666	29651	35701
	512	6679	56578	137138	6131	3647	1168	127349	54463	8233	8384	59524	26655
20x20	64	5311	0	18	0	280914	0	0	62073	0	0	0	13059
	128	9331	14567	3827	0	182479	60	469	127135	0	0	8599	94271*
	256	18723	18803	77088	2834	58454	8262	7808	195381*	574	1084	20542	65596
	512	61792	14221	280753	134966	11311	6857	2676	24447	3178	11488	146423*	28157
25x25	64	5343	0	210	0	390163*	0	0	44149	0	0	0	0
	128	60309	0	6755	670	263472	2904	0	49643	0	3255	9110	18908
	256	70178	5018	101018	34682	63191	2087	0	124625	654	0	47793	46044
	512	233285	0	363525	814885*	19024	249643	0	5500	62293	123448	40270	14875
30x30	64	6685	0	130	0	270011	82	0	21815	0	0	0	0
	128	69043	0	8785	1148	161432	5305	0	106122	0	983	13943	43964
	256	308570	0	240888	40245	21765	10452	0	17017	0	46750	82542	39503
	512	422643*	0	422500*	284076	1548	493673*	0	8	328527*	583184*	17720	0

discutido adiante.

O critério utilizado para a escolha dos melhores cenários considerou a taxa mínima de acurácia da partição A, $\tau = 0,9$. Os maiores cenários com discriminabilidade igual ou acima desse limite e os cenários obtidos nas rodadas sucessoras foram incluídos na avaliação. Por exemplo, o algoritmo de validação encontrou combinações de até 7 classes ($|\delta| = 7$) e seus respectivos parâmetros com taxa de acurácia $\tau \geq 0,9$. Na rodada seguinte, a taxa de acurácia foi reduzida para aceitar cenários maiores, utilizando o passo de acurácia $\eta = 0,1$. Essa redução possibilitou validar cenários com 8, 9, 10, 11 e 12 classes. Nesse caso, a contagem das ocorrências dos melhores parâmetros considera apenas os cenários $|\delta| = [7, 8, 9, 10, 11 \text{ e } 12]$. Cenários menores, que fornecem um grande número de possibilidades, foram descartados para não enviesar a contagem total.

Para otimização dos gráficos das Figuras 5.9 e 5.10, são exibidas apenas as combinações de parâmetros que alcançaram o máximo de ocorrências para, pelo menos, uma classe, identificadas com asterisco (*) nas respectivas Tabelas 5.3 e 5.4. Nos experimentos com medidas de compressão (Figura 5.9), as maiores ocorrências foram obtidas pelas classes e seus respectivos pares de parâmetros $[\gamma, \lambda]$, em ordem decrescente: [flor, 25x25, 512], [árvore, 5x5, 512], [campo, 30x30, 512], [pôr do sol, 30x30, 512], [avião, 30x30, 512], [fogos de artifício, 30x30, 512], [moto, 25x25, 64], [praia, 30x30, 512], [construção, 15x15, 128], [corrida de motos, 20x20, 256], [montanha, 20x20, 512] e [carro, 20x20, 128]. Já nos experimentos com a medida de entropia (Figura 5.10), as maiores ocorrências foram, em ordem decrescente: [árvore, 5x5, 256], [corrida de motos, 10x10, 256], [praia, 5x5, 128], [flor, 25x25, 256], [avião, 30x30, 512], [moto, 10x10, 512], [construção, 5x5, 256], [pôr do sol, 10x10, 128], [campo, 5x5, 128], [fogos de artifício, 30x30, 512], [carro, 10x10, 256] e [montanha, 15x15, 256]. Os resultados mostram predominân-

cia de janelas maiores e dicionários com 512 símbolos; mas também incluem cenários com todos os tamanhos de dicionário e janela, considerando as duas medidas de aleatoriedade utilizadas.

Conforme discutido preliminarmente, a definição do tamanho das janelas remete à noção de granularidade da informação, ou seja, à unidade adequada de informação com a presença de semântica da classe. Se delineada de maneira apropriada, essa unidade é capaz de representar os componentes da cena, mantendo o mínimo de informação útil. Outra perspectiva de entendimento das janelas está relacionada ao conteúdo da informação que se deseja extrair da cena, ou seja, o nível de abstração. Dentro de uma mesma classe, diversos tamanhos de janelamento permitem extrair diferentes interpretações dos dados. Por exemplo, uma imagem de satélite pode ser utilizada tanto para cálculo da área de edificações quanto para determinar o percentual de vegetação na cobertura do solo. Esse nível de abstração pode ser ajustado para detectar objetos ou partes específicas de uma cena. Entretanto, nesta tese é discutida apenas a similaridade de cenas.

Nesse contexto, propõe-se expandir os conceitos da área de reconhecimento clássico de padrões para incluir a definição de *componentes da cena*, que representam

Ocorrências dos parâmetros nos melhores cenários da partição A de acordo com a medida de entropia

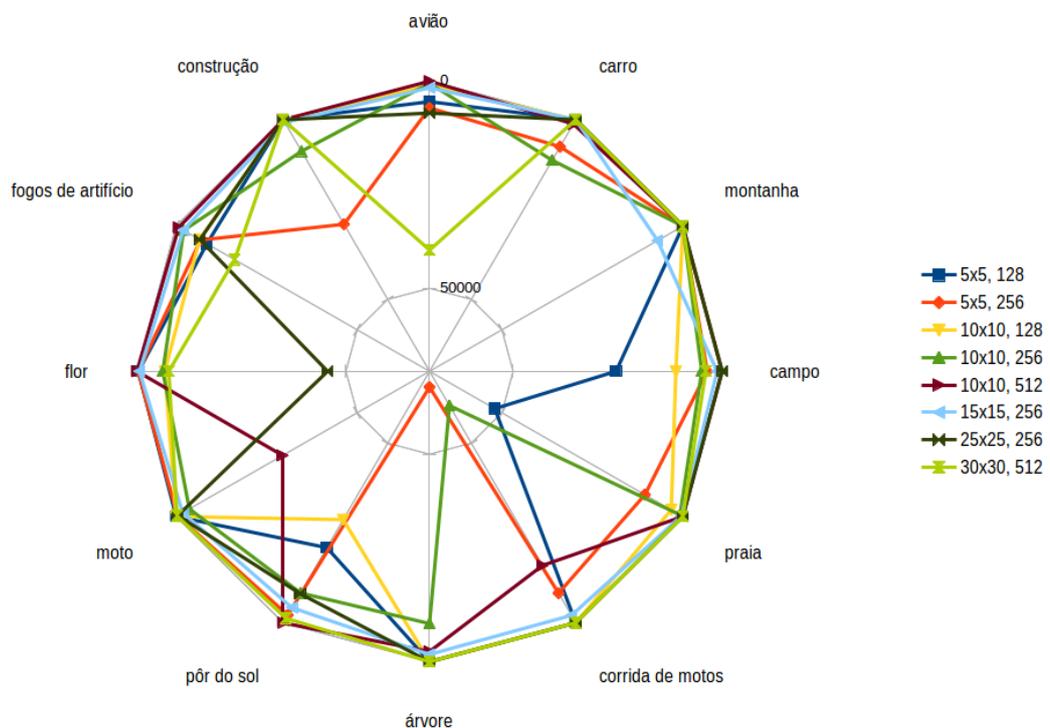


Figura 5.10: *Frequência dos parâmetros de tamanhos da janela e do dicionário de cada classe, obtidos a partir dos melhores cenários na partição A e medida de entropia.*

Tabela 5.4: Ocorrências dos parâmetros observados nos melhores cenários, obtidos na validação do treinamento da partição A e medida de entropia.

Janela	Dicionário	avião	construção	fogos de artifício	flor	moto	pôr do sol	árvore	corrida de motos	praia	campo	montanha	carro
5x5	64	0	0	106	0	0	0	0	0	0	0	0	0
	128	5023	0	8510	371	0	20902	0	0	51972*	25367*	0	0
	256	6522	29086*	6753	321	353	2152	66183*	8236	10494	3564	0	7511
	512	0	10099	895	0	7332	0	15911	1605	0	0	0	465
64	0	0	177	0	0	0	0	0	0	0	0	0	
10x10	128	1098	0	6490	6669	0	28695*	0	0	3207	11054	0	0
	256	620	8765	2087	6202	3655	8298	9185	60452*	798	4867	0	11285*
	512	87	0	702	0	29405*	0	2439	15944	0	0	0	1175
	64	0	0	56	0	0	0	0	0	0	0	0	0
15x15	128	0	0	8758	1146	0	2065	0	0	0	4199	0	0
	256	1680	741	1971	629	1806	4200	1748	2069	362	1193	7003*	0
	512	2160	234	734	642	27786	0	78	213	0	0	0	0
	64	0	0	0	0	0	0	0	0	0	0	0	0
20x20	128	0	0	6827	0	0	0	0	0	0	0	0	0
	256	10690	0	4021	10620	1377	3937	0	0	0	347	0	0
	512	5355	0	2528	8727	18825	0	0	5364	0	1592	0	6309
	64	0	0	0	0	0	0	0	0	0	0	0	0
25x25	128	0	0	5460	0	0	0	0	0	0	0	0	0
	256	7787	0	6550	45623*	451	8105	0	0	0	0	0	0
	512	11570	0	6311	2208	7957	0	0	0	0	0	0	5755
	64	0	0	11	0	0	0	0	0	0	0	0	0
30x30	128	0	0	5921	0	0	0	0	0	0	0	0	0
	256	2093	0	10592	7014	0	1857	0	0	0	0	0	0
	512	40788*	0	16170*	7624	255	1269	0	0	0	3880	0	0

a granularidade da informação visual. Segundo [141]⁵, granularidade da informação está relacionada ao volume de atributos (características) que compõem um conjunto de dados (imagem) e à diversidade de seus conteúdos (complexidade). Granularidade fina (janelas menores) é caracterizada pela presença de maior detalhamento, enquanto no menor nível de detalhamento a granularidade é dita grossa (janelas maiores) [141]. Considerando que a granularidade atua em conjunto com a Teoria da Informação na concepção das técnicas de compressão de dados, a proposição do conceito de *componentes*, como partes de uma cena, se faz necessária na discussão do RPA.

As abordagens clássicas de reconhecimento de padrões⁶, não incluem a Teoria da Informação como componente-chave para descrever as classes de um problema. Nessas abordagens, o processo de decisão é efetuado a partir de características já estruturadas e validadas. Assim, as partes que compõem os objetos são previamente delineadas em etapas auxiliares de extração e seleção de características. Sob o enfoque do RPA, as cenas são entendidas como um conjunto de componentes integrados, que a estruturam visualmente, e esses componentes, por sua vez, são formados por características. As cenas, de acordo com sua classe, possuem diferentes características que as descrevem, assim como seus componentes apresentam maior ou menor grau de complexidade para representá-los.

Segundo [65], a grande maioria das teorias de representação de objetos natu-

⁵Observações, entre parênteses, incluídas para alinhar o conteúdo da citação com o contexto do presente trabalho.

⁶Segundo [134], as técnicas de reconhecimento de padrões em imagens podem ser divididas, principalmente, mas não exclusivamente, em três abordagens: estatística, estrutural e neural. Em [73] os autores também consideram a abordagem de casamento de modelos (*template matching*).

rais assumem que estes não podem ser representados apenas pelas abordagens clássicas⁷. Essas abordagens se limitam a mensurar a similaridade a partir de dois tipos de representação: pontos no espaço ou conjuntos de características. Em contraponto, os autores hipotetizam que a modelagem da complexidade dos estímulos exige representações estruturadas. Essa abordagem é capaz de descrever, explicitamente, os objetos e suas inter-relações, bem como a dinâmica de decomposição do todo em componentes e suas respectivas sub-partes. Nessa perspectiva, de diversidade das representações dos objetos, o RPA atua no sentido de identificar a complexidade visual, por meio do ajuste dos melhores tamanhos de janelas e quantidades de símbolos por dicionário. Considerando que: 1) as dimensões da janela de uma classe determina o tamanho dos componentes que a estruturam e 2) os pixels dessas janelas são as características do componente; portanto, uma mesma característica pode transitar e atuar com diferentes graus de importância entre os diversos janelamentos definidos em parâmetro.

A distribuição das amostras de treinamento de uma classe, pelo espaço de prototipação, determina a quantidade necessária de símbolos para representá-las de forma adequada. Vetores de treinamento muito próximos no espaço de prototipação não exigem muitos *codevectors* para uma reconstrução de qualidade da imagem. Entretanto, a caracterização dos componentes de uma cena em um problema de classificação exige *codebooks* maiores. Essa dicotomia ocorre em razão da tarefa principal ser de discriminação de classes, e não reconstrução de imagens. Portanto, primando apenas pela qualidade das imagens, deduz-se que quanto maior o *codebook* melhor a aproximação entre essa representação e os dados originais. Seguindo essa lógica, todas as classes de um problema deveriam ser prototipadas com o máximo possível de símbolos. Entretanto, 2 fatores contradizem essa afirmação:

1. Ao aproximar uma representação comprimida (*codebook*) dos dados originais de uma imagem (processo de quantização), aumentando a quantidade de símbolos, a reconstrução tende a não eliminar ruídos ou detalhes da informação visual. Esses ruídos ou detalhes, eventualmente, podem influenciar de maneira negativa a identificação das características mais discriminantes;
2. Em se tratando de um problema de classificação, a calibração do nível de abstração das cenas é a etapa inicial para promover a maximização das distâncias extraclasse. Em outras palavras, as características que fomentam o processo de decisão devem ser definidas para não causar ambiguidade. Prototipar todas as classes de um problema com a mesma quantidade de símbolos, implica colocar todos os seus respectivos componentes no mesmo

⁷Abordagem espacial [146], que representa a similaridade em termos de distância em um espaço psicológico; e abordagem de modelo de contraste [159], que considera a similaridade como uma função de características comuns e distintas de entidades sob comparação.

patamar de complexidade visual. Na prática isso é realizado pelo sistema visual atento humano no reconhecimento de cenas.

A medida da distorção da prototipação, apresentada nas Figuras 5.6, 5.7 e 5.8, utilizada para demonstrar equivalência entre as partições, não é adequada para estabelecer uma relação com o número ideal de símbolos para representar cada classe. Independente do distanciamento das amostras no espaço, dicionários menores fornecerão distorções maiores. Portanto, utiliza-se uma medida de dispersão que representa a distância média entre as amostras e o ponto central do conjunto. A Figura 5.11 mostra as dispersões de cada classe, calculadas a partir das amostras da partição A, proporcionais à classe de referência *fogos de artifício*, com menor dispersão. Essas mesmas proporções foram observadas em todos os tamanhos de janelas, com pequenas variações. O menor valor atribuído à classe *fogos de artifício* se deve à predominância de janelas escuras, que tendenciam as amostras da classe a se localizarem em uma região muito específica do espaço de prototipação. Mesmo assim, classes com maior diversidade de componentes visuais como *praia*, *avião* e *flor* apresentam dispersão apenas cerca de 50% superior em relação à referência. A maior dispersão foi observada para a classe *moto*, considerada um *outlier* nesse quesito, em razão das amostras utilizadas possuírem fundo branco. Assim, grande parte dos vetores de treinamento coincidiram no ponto máximo do espaço, enquanto os demais se dispersaram de acordo com os componentes visuais das motos. Nesse caso, apesar de promover grande concentração das janelas em um único ponto, isso não enviesou o treinamento, pois o algoritmo LBG não particiona o espaço de acordo com a quantidade de elementos.

A Tabela 5.5 apresenta as classes e seus respectivos parâmetros de janela e tamanho do dicionário, obtidos nos experimentos com as medidas de compressão (Figura 5.9) e entropia (Figura 5.10). As colunas *Distorção* e *Dispersão* trazem as respectivas po-

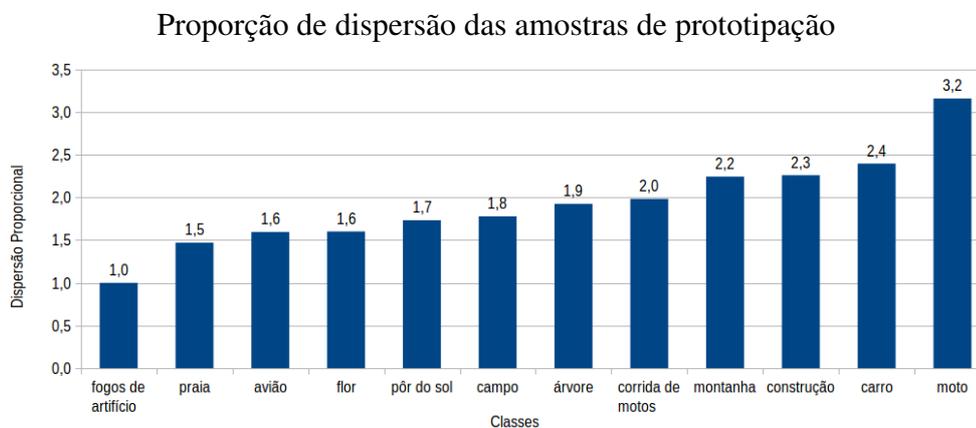


Figura 5.11: Dispersões dos vetores de treinamento de cada classe, proporcionais à classe de menor distorção *fogos de artifício*.

Tabela 5.5: *Ordenação crescente das medidas de distorção e dispersão das classes e suas relações com as maiores ocorrências de parâmetros de janela e dicionário, de acordo com as medidas de compressão e entropia.*

	Ordem		Compressão		Entropia	
	Distorção	Dispersão	Janela	Dicionário	Janela	Dicionário
árvore	1 ^a	7 ^a	5x5	512	5x5	256
construção	3 ^a	10 ^a	15x15	128	5x5	256
carro	5 ^a	11 ^a	20x20	128	10x10	256
corrida de motos	4 ^a	8 ^a	20x20	256	10x10	256
montanha	6 ^a	9 ^a	20x20	512	15x15	256
moto	2 ^a	12 ^a	25x25	64	10x10	512
flor	10 ^a	4 ^a	25x25	512	25x25	256
avião	8 ^a	3 ^a	30x30	512	30x30	512
campo	9 ^a	6 ^a	30x30	512	5x5	128
fogos de artifício	7 ^a	1 ^a	30x30	512	30x30	512
pôr do sol	12 ^a	5 ^a	30x30	512	10x10	128
praia	11 ^a	2 ^a	30x30	512	5x5	128

sições das classes referentes a essas medidas; na ordem, 1^a indica menor valor e 12^a maior valor. Avaliando os parâmetros obtidos pelas medidas de compressão, observa-se que as classes com 512 símbolos apresentaram menor dispersão. Esses resultados corroboram a argumentação de que, quanto menor a dispersão da classe, menor a discriminabilidade dos componentes da cena e, portanto, são necessários mais símbolos para diferenciá-los sem perder informações sensíveis.

Também pode ser observado na Tabela 5.5 que, utilizando as medidas de compressão, classes com maior valor de distorção na prototipação tendem a apresentar melhor desempenho com janelas maiores. Nesse caso, a medida de distorção está relacionada à granularidade dos componentes, na qual distorções menores indicam maior proximidade entre os vetores de treinamento representados pelo mesmo *codevector*. Portanto, quanto menor a distorção de uma classe, mais semelhantes visualmente são seus componentes e, conseqüentemente, são necessárias janelas menores para extração das informações e cálculo da complexidade algorítmica. Em relação a entropia, embora classes com mesmo tamanho de janela sejam posicionadas próximas pela distorção, não são observadas relações diretas com as medidas de distorção e dispersão. Atribui-se esse resultado à natureza da entropia de Shannon que trata a complexidade de um objeto na média; assim, é incapaz de descrever sua aleatoriedade intrínseca. Desse modo, a entropia consegue captar a complexidade média da cena, mas não a complexidade na relação entre os componentes.

A Figura 5.12 ilustra 4 possíveis distribuições das amostras de treinamento no plano bidimensional, em função da distorção e dispersão. Na ocorrência de alta distorção e dispersão, os vetores de treinamento se encontram distribuídos por todo o espaço,

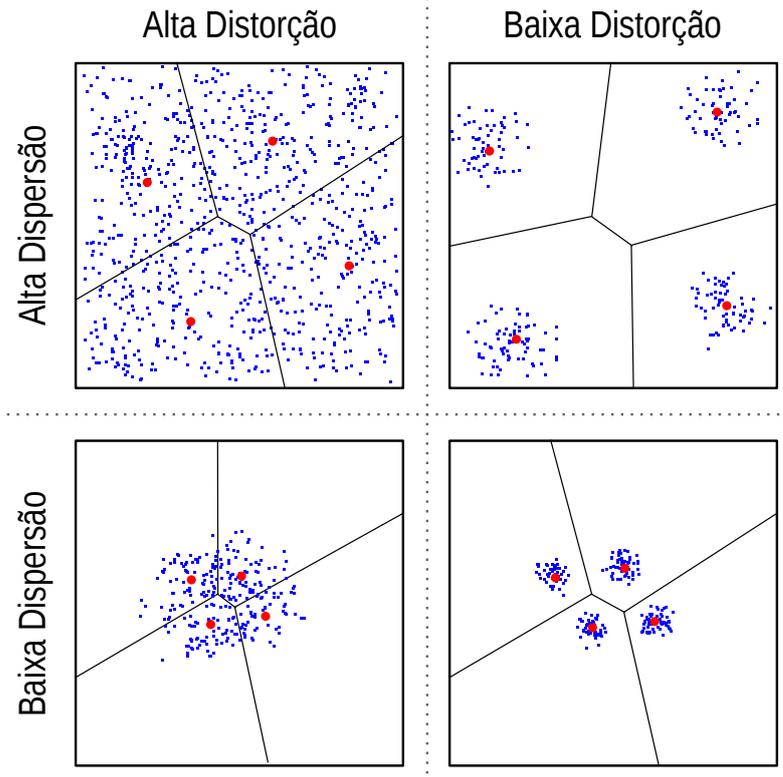


Figura 5.12: Possíveis casos de distorção e dispersão de uma distribuição hipotética de amostras de treinamento no espaço bidimensional.

inclusive dentro das próprias partições. No caso de baixa distorção e alta dispersão, os vetores de treinamento se distribuem por todo espaço, porém agrupados nas partições. A alta distorção em conjunto com a baixa dispersão pode ser observada quando as amostras são projetadas, de forma concentrada, em uma região. A distribuição dos vetores de treinamento em uma região limitada do espaço, porém com agrupamentos definidos, ocasiona tanto baixa distorção quanto dispersão. Portanto, dispersão é uma medida extra-partição e distorção é uma medida intra-partição, do Diagrama de Voronoi.

5.2.3 Avaliação de Desempenho

A Tabela A.1 lista os cenários avaliados nos experimentos de desempenho do RPA, nomeados com códigos de identificação (*ID*) e o detalhamento de cada conjunto de classes e seus respectivos parâmetros (*Cenário*). Os resultados de classificação, respeitando as taxas de acurácia mínimas τ definidas para cada partição, são sumarizados nas Tabelas B.14 e B.15 para as medidas de compressão, e na Tabela B.16 para a medida de entropia. Os maiores e menores valores de cada medida de desempenho são identificados com dois asteriscos (**) e um asterisco (*), respectivamente. Durante a etapa de *Validação*, notou-se que a quantidade de cenários validados com as poucas amostras

de treinamento da partição C era muito superior em relação às partições A e B. Como consequência, mais classes e seus respectivos parâmetros eram aceitos para as rodadas subsequentes, tornando a busca muito ampla e computacionalmente dispendiosa. Então, optou-se por fixar a taxa de acurácia mínima de $\tau = 0,9$ para as partições A e B e de $\tau = 0,95$ para a partição C. Assim, nas Tabelas B.14, B.15 e B.16 estão relacionados apenas os resultados da primeira ocorrência de cenário máximo para todas as combinações de medidas, hiperparâmetros e tamanhos de cenários. O intuito não é apresentar todos os possíveis cenários, mas uma visão geral do desempenho do RPA em diversas configurações do experimento. A coluna *ID* faz referência aos códigos de identificação da Tabela A.1; *W* indica a quantidade de classes; as colunas *Validação* e *Generalização* representam, respectivamente, a acurácia obtida na etapa de validação e sua diferença entre a acurácia média da classificação.

Analisando as taxas de generalização nas Tabelas B.14, B.15 e B.16, pode-se observar diferenças expressivas entre as taxas de acurácia dos conjuntos de teste e validação, variando entre -0,5865 (*ID* 215 da Tabela B.15) e 0,0038 (*ID* 220 da Tabela B.15). Para as medidas de compressão, as menores generalizações ocorreram na partição A (Tabela B.14) em cenários maiores com os compressores ZIP, *gzip* e *bzip2*. Considerando uma variação de até -0,05 na taxa de generalização, a compressão *LZW+Huffman* apresenta melhor desempenho em relação aos demais métodos. Esse desempenho também é observado quando se estabelece o limite mínimo de 0,80 de acurácia no conjunto de teste. Em valores nominais, os cenários com 11 classes tiveram maior variação de generalização e em valores médios, cenários com 7 classes tiveram a menor variação.

No geral, ao comparar as taxas de acurácia entre a etapa de *Validação* e a classificação do conjunto de testes, podem ser encontrados resultados muito próximos. Porém, há casos expressivamente negativos, com menos da metade da acurácia obtida na validação. Deve-se enfatizar que a etapa de *Validação* não necessariamente garante cenários com boa generalização, entretanto, é uma estratégia válida para encontrar bons resultados dentro de um grande conjunto de possibilidades. Dentro dessa perspectiva, embora a prototipação não seja sensível ao tamanho do conjunto de treinamento, o processo de validação depende de mais amostras para obter cenários de classificação com melhor generalização. Avalia-se que um dos componentes mais promissores para aperfeiçoar o desempenho do RPA é a etapa de *Validação*. O intuito é encontrar os parâmetros que maximizam a complexidade na quantização convergente e minimizam na quantização divergente. Dessa forma, os resultados podem ser aprimorados e o método estendido para outras aplicações como, por exemplo, detecção de objetos.

Nas colunas *Desvio* das Tabelas B.14, B.15 e B.16, estão relacionados os desvios-padrão observados entre os 10 *holdouts* para todas as métricas de avaliação de

desempenho: acurácia, *precision*, *recall* e *F1 score*. Considera-se os valores baixos, com desvio-padrão médio em torno de 0,02 para todas as métricas. A Tabela 5.6 sumariza os valores de desvio-padrão máximos, mínimos e médios. Considerando as interseções entre os *holdouts*, descritas na Tabela 5.2, observa-se que, mesmo garantido a diversidade entre os *holdouts*, o RPA se mostra estável e independente em relação à composição dos conjuntos de treinamento e teste.

Também pode ser verificado que os resultados das classificações utilizando os compressores *ZIP* e *gzip* foram idênticos. Apesar de compartilharem o mesmo núcleo de compressão⁸, o *ZIP* e o *gzip* possuem diferenças na forma de armazenar os metadados no cabeçalho e combinar os arquivos a serem comprimidos. A maior delas é que o *ZIP* possui suporte para a modalidade de arquivamento, quando vários arquivos e diretórios podem ser comprimidos em um único pacote. O *gzip* permite apenas a compressão de arquivos individuais. Mesmo comprimindo nativamente múltiplos arquivos, o *ZIP* não explora a redundância entre eles pois a compressão é realizada de forma individual, para depois empacotá-los [81]. Já o *gzip* pode explorar essa redundância utilizando o recurso de *tarball*, que concatena vários arquivos e os comprime (compressão sólida). Embora as imagens comprimidas por esses algoritmos tenham tamanhos diferentes, os comprimentos das descrições não foram suficientes para discriminá-las. Portanto, nas discussões seguintes eles serão agrupados para representar o mesmo resultado. Por outro lado, vale ressaltar que a implementação *LZW+Huffman*, similar ao DEFLATE, ao desconsiderar o dicionário e a árvore gerados no processo de compressão foi mais eficiente.

Em relação aos resultados obtidos pela medida de entropia (Tabela B.16), ao contrário das medidas de compressão, as melhores taxas de generalização foram alcançadas pelos cenários validados na partição A. Além disso, de modo geral, as diferenças entre a acurácia média dos conjuntos de teste e de validação da partição A são muito pequenas em relação às demais, chegando no máximo a 0,0995. Porém, a acurácia média dos conjuntos de teste foi inferior em relação às medidas de compressão para cenários de

Tabela 5.6: Limites máximos e mínimos e valores médios de desvio-padrão observados entre as reamostragens dos dados de validação e teste para as métricas de avaliação de desempenho.

	Acurácia	Precision	Recall	F1 score
Máximo	0,0441	0,0496	0,0441	0,0456
Mínimo	0,0081	0,0074	0,0081	0,0086
Médio	0,0224	0,0210	0,0224	0,0232

⁸Algoritmo DEFLATE, que combina *LZ77* + *Huffman*.

até 8 classes. As Tabelas 5.7 e 5.8 relacionam as taxas de acurácia máximas dos conjuntos de teste utilizando compressão e entropia, respectivamente, em cada um dos tamanhos de cenários avaliados. Na Tabela 5.7 nota-se que a maioria dos melhores resultados foram obtidos pela compressão *bzip2* para cenários de 7 (0,9286) a 11 classes (0,7473), considerando a acurácia. Já os resultados das classificações utilizando entropia são exibidos na Tabela 5.8, para cenários de 7 (0,9029) a 12 classes (0,6967). Recorrendo aos dados da Tabela B.16, constatou-se que, apesar da predominância de taxas de convergência e menores nos melhores resultados da entropia, não há relação direta no aumento da acurácia do RPA em função da diminuição de ϵ . A Figura 5.13 apresenta um comparativo entre as acurácias das medidas de compressão e entropia. Pode-se observar que para cenários de até 8 classes a compressão é mais eficiente e de 9 a 12 classes o desempenho da entropia é superior.

Algumas configurações de classificação serão discutidas de maneira mais específica, incluindo variações de parâmetros e apresentação das respectivas matrizes confusão. A Figura 5.14 explica de forma estruturada o delineamento utilizado para a escolha dessas configurações. Esse delineamento foi igualmente aplicado nos experimentos utilizando compressão e entropia. Os tamanhos de cenários escolhidos para discussão foram de 12 classes (*item 1*), por cobrir todas classes do problema, e de 7 classes (*item 2*), por ser o menor cenário validado. Os *itens 1.1* e *2.1* se referem ao melhor resultado de acurácia do conjunto de teste para 12 e 7 classes, respectivamente; e os *itens 1.2* e *2.2* os piores resultados de acurácia. Para o caso do melhor resultado (*item 1.1*) foram aplicadas 2 variações, mantendo as classes e parâmetros do cenário: variações entre as partições A, B e C e as taxas de convergência ϵ , totalizando 15 combinações (*item 1.a*); variação das medidas de compressão ZIP/gzip e *bzip2* (*item 1.b*). Os melhores e piores resultados observados nessas variações são identificados pelos *itens 1.1.1* e *1.1.3* (melhores) e *itens 1.1.2* e *1.1.4* (piores). Dessa forma, foi possível verificar se outro cenário máximo poderia ser encontrado, além daquele já avaliado no conjunto de teste. Para organização do texto, a partir daqui, as referências ao termo *item* remetem sempre à Figura 5.14.

Tabela 5.7: Detalhes das condições de testes das maiores acurácias obtidas em cenários de 6 a 12 classes utilizando compressão.

ID	# Classes	Acurácia	Precision	Recall	F1 score	Convergência	Compressor
106	12 classes	0,6350	0,6575	0,6350	0,6337	0,00001	LZW+Huffman
28	11 classes	0,7473	0,7777	0,7473	0,7510	0,01	<i>bzip2</i>
27	10 classes	0,7820	0,8032	0,7820	0,7767	0,01	<i>bzip2</i>
26	9 classes	0,8311	0,8428	0,8311	0,8310	0,01	<i>bzip2</i>
25	8 classes	0,9000	0,9172	0,9000	0,9012	0,01	<i>bzip2</i>
24	7 classes	0,9286	0,9369	0,9286	0,9294	0,01	<i>bzip2</i>
100	6 classes	0,9200	0,9245	0,9200	0,9207	0,00001	LZW+Huffman

Tabela 5.8: *Detalhes das condições de testes das maiores taxas de acurácia obtidas em cenários de 7 a 12 classes utilizando entropia.*

ID	# Classes	Acurácia	Precision	Recall	F1 score	Convergência
93	12 classes	0,6967	0,7260	0,6967	0,6912	0,00001
78	11 classes	0,7727	0,7863	0,7727	0,7738	0,0001
77	10 classes	0,8220	0,8303	0,8220	0,8212	0,0001
6	9 classes	0,8511	0,8597	0,8511	0,8504	0,1
89	8 classes	0,8825	0,8886	0,8825	0,8827	0,00001
4	7 classes	0,9029	0,9040	0,9029	0,9030	0,1

O melhor resultado de 12 classes (*item 1.1*), utilizando compressão e observado entre as Tabelas B.14 e B.15, obteve 0,6350 de taxa de acurácia, treinado e testado na partição A com $\epsilon = 0,00001$. Sua respectiva matriz confusão é especificada na Figura 5.15. O resultado foi alcançado utilizando a compressão *LZW+Huffman* no cenário {[avião, 30x30, 256], [praia, 30x30, 512], [construção, 15x15, 128], [carro, 20x20, 128], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corridas de moto, 20x20, 64], [moto, 25x25, 64], [montanha, 25x25, 256], [pôr do sol, 30x30, 512], [árvore, 5x5, 512]}. Para demonstrar experimentalmente a equivalência entre os particionamentos e taxas de convergência ϵ , esse mesmo cenário foi utilizado para classificação das imagens de teste da partição A em diferentes configurações. Das partições B e C foram utilizados apenas os *codebooks* treinados para garantir que não ocorresse nenhuma intersecção de amostras entre eles. Considerando a compressão como medida de complexidade da cena, o objetivo desse cruzamento de resultados é duplo: demonstrar que quantidades menores de amostras de treinamento são suficientes para prototipar as classes; e que a taxa de convergência $\epsilon = 0,001$ é adequada para o ajuste dos *codevectors*.

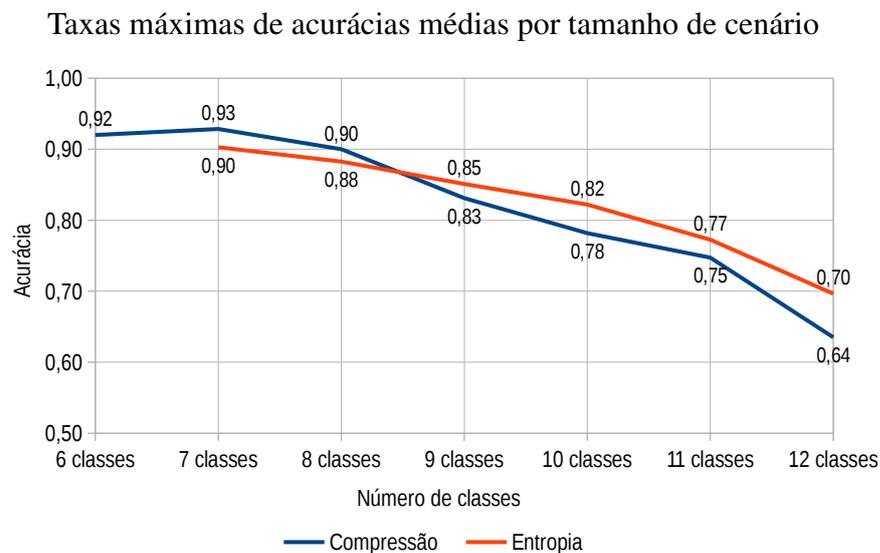


Figura 5.13: *Maiores taxas de acurácia médias obtidas nos conjuntos de teste para cenários de 6 a 12 classes (detalhes nas Tabelas 5.7 e 5.8).*

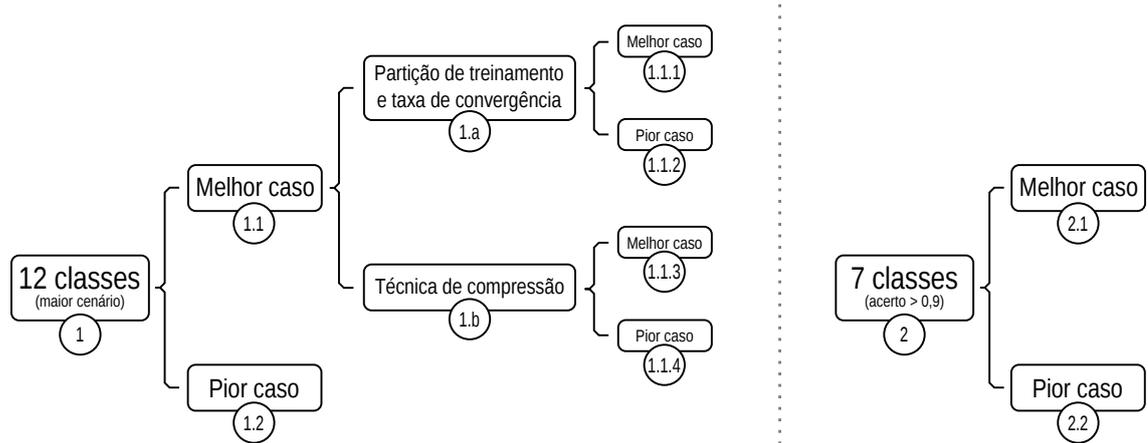


Figura 5.14: Delineamento experimental utilizado como critério para discussão da acurácia do RPA.

A Figura 5.16 apresenta os resultados comparativos do item 1.a utilizando a compressão. Pode ser observado que a variação para as maiores taxas de acurácia, em função do tamanho conjunto de treinamento, é máxima em $\epsilon = 0,0001$ e mínima, na média entre as partições, em $\epsilon = 0,1$. Apesar de treinados com apenas 50% das amostras da partição A, os *codebooks* obtidos da partição B foram capazes de alcançar a taxa de acurácia de 0,6070 em $\epsilon = 0,001$. Comparando com o resultado de 0,6350, da partição A, a diferença é de apenas 0,0280. Essa variação é semelhante entre as partições B e C (0,0320). Dessa maneira, as constatações baseadas na distorção da prototipação são demonstradas experimentalmente nos testes de acurácia da classificação. Portanto, sugere-se, para esse *dataset*, que treinamentos com 15 amostras de cada classe e taxa de convergência $\epsilon = 0,001$ são adequados para manter a eficiência do RPA,

		Classe Predita											
		Avião	Praia	Construção	Carro	Campo	Fogos de Artifício	Flor	Corrida de Moto	Moto	Montanha	Pôr do Sol	Árvore
Classe Verdadeira	Avião	0,58	0,16	0	0	0,18	0	0,04	0,02	0	0,02	0	0
	Praia	0,40	0,38	0	0	0,18	0	0,02	0	0	0	0,02	0
	Construção	0,02	0	0,68	0,02	0	0	0,04	0,14	0	0,06	0	0,04
	Carro	0,14	0	0	0,34	0	0,04	0,10	0,18	0	0,16	0,04	0
	Campo	0,04	0,12	0	0	0,50	0	0,02	0,02	0	0,04	0,18	0,08
	Fogos de Artifício	0	0	0	0	0	0,92	0	0	0	0	0,08	0
	Flor	0,02	0	0,02	0,02	0,04	0,02	0,74	0,08	0	0,02	0,04	0
	Corrida de Moto	0,04	0,02	0,02	0,18	0	0	0,06	0,64	0	0,02	0,02	0
	Moto	0	0	0	0,02	0	0	0	0	0,96	0,02	0	0
	Montanha	0,30	0,06	0,06	0,04	0	0	0	0,12	0	0,34	0,08	0
	Pôr do Sol	0,08	0,02	0	0	0,06	0	0,10	0	0	0,02	0,72	0
	Árvore	0	0	0,04	0	0	0	0,12	0,02	0	0	0	0,82

Figura 5.15: Matriz confusão do melhor resultado (itens 1.1, 1.1.1 e 1.1.3) obtido pelas medidas de compressão em cenários com 12 classes - incluindo variações de partição de treinamento, taxa de convergência ϵ e compressor - treinado e testado na partição A com $\epsilon = 0,00001$ e LZW+Huffman, conforme Tabela B.14 (ID: 106, acurácia: 0,6350, precision: 0,6487, recall: 0,6350, F1 score: 0,6337).

Taxa de acurácia do melhor cenário utilizando compressão para as partições e taxas de convergência ϵ

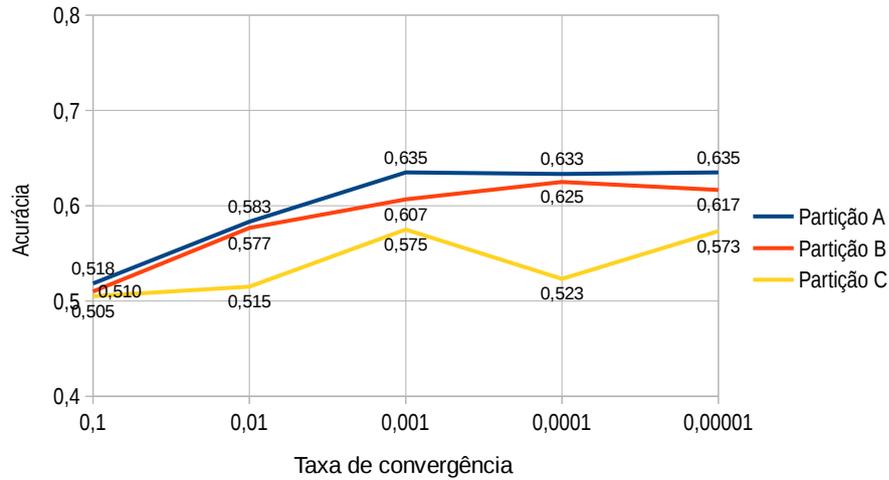


Figura 5.16: Taxas de acurácia da compressão *LZW+Huffman* comparada entre os treinamentos obtidos a partir das partições A, B e C e $\epsilon = \{0,1; 0,01; 0,001; 0,0001 \text{ e } 0,00001\}$ classificando o conjunto de teste da partição A.

utilizando a compressão *LZW+Huffman* como medida de complexidade. Os resultados da comparação mostram que a melhor acurácia, variando as partições de treinamento e a taxa de convergência ϵ (item 1.1.1), se mantém para o melhor caso observado (0,6350 com $\epsilon = 0,001$), segundo o critério do item 1.1. O pior resultado foi causado pelo treinamento da partição C com $\epsilon = 0,1$ (item 1.1.2), detalhado na matriz confusão da Figura 5.17 com 0,5050 de acurácia.

Em relação à variação da técnica de compressão no melhor cenário com 12 classes (item 1.b), foi observado que o melhor caso se mantém utilizando o *LZW+Huffman*

		Classe Predita											
		Avião	Praia	Construção	Carro	Campo	Fogos de Artifício	Flor	Corrida de Moto	Moto	Montanha	Pôr do Sol	Árvore
Classe Verdadeira	Avião	0,56	0,20	0	0,02	0	0	0,04	0,02	0,02	0,04	0,10	0
	Praia	0,40	0,36	0	0	0	0	0	0,02	0	0,12	0,10	0
	Construção	0	0	0,30	0,02	0	0	0	0,32	0	0,04	0	0,32
	Carro	0,10	0	0,04	0,30	0	0	0	0,48	0	0,08	0	0
	Campo	0,08	0,14	0,04	0,04	0,20	0	0,02	0	0	0,14	0,10	0,24
	Fogos de Artifício	0	0	0	0,08	0	0,58	0,12	0,08	0	0	0,14	0
	Flor	0	0,04	0	0,04	0	0	0,26	0,50	0	0,08	0,06	0,02
	Corrida de Moto	0,06	0,02	0,02	0,16	0,02	0	0	0,64	0,02	0,06	0	0
	Moto	0	0	0	0	0	0	0	0	1,00	0	0	0
	Montanha	0,16	0	0,02	0,04	0	0	0,02	0,16	0	0,48	0,04	0,08
	Pôr do Sol	0,02	0,04	0	0,08	0,08	0,02	0,10	0,04	0	0,20	0,40	0,02
	Árvore	0	0	0	0,02	0	0	0	0	0	0	0	0,98

Figura 5.17: Matriz confusão do pior resultado (item 1.1.2) obtido no conjunto de teste A pela compressão *LZW+Huffman* em cenários com 12 classes, variando as partições de treinamento e as taxas de convergência ϵ , treinado na partição C com $\epsilon = 0,1$ conforme Figura 5.16 (Acurácia = 0,5050).

(item 1.1.3) e o pior caso (item 1.1.4), coincidentemente com acurácia de 0,17, foi obtido pelos compressores *bzip2* e *ZIP/gzip*. A matriz confusão do *bzip2*, detalhada na Figura 5.18, mostra que todas as imagens foram classificadas entre as classes *praia*, *campo* e *fogos de artifício*. Já o pior resultado geral para cenários de 12 classes (item 1.2) observado entre Tabelas B.14 e B.15 foi para a partição C, treinada com $\epsilon = 0,001$ e testada com ZIP; a taxa de acurácia foi de 0,0,2931 e a matriz confusão é apresentada na Figura 5.19.

Para cenários de 7 classes, a partir da Tabela 5.7, pode ser verificado que o melhor resultado (item 2.1), utilizando compressão, obteve 0,9286 de acurácia treinado em $\epsilon = 0,01$ e testado com o *bzip2* na partição A. A matriz confusão desse resultado é apresentada na Figura 5.20 para o cenários {[*praia*, 5x5, 512], [*construção*, 10x10, 512], [*fogos de artifício*, 5x5, 512], [*flor*, 10x10, 512], [*corrida de motos*, 10x10, 512], [*moto*, 25x25, 128], [*árvore*, 5x5, 512]}. A matriz confusão do pior caso (item 2.2) é apresentada na Figura 5.21, com acurácia de 0,7200 na partição A com $\epsilon = 0,00001$ e ZIP.

A mesma metodologia descrita na Figura 5.14 foi aplicada à medida de entropia, a única exceção o item 1.b. De acordo com a Tabela 5.8, o melhor cenário de 12 classes (item 1.1) obteve acurácia de 0,6967 no conjunto de teste da partição A, treinado em $\epsilon = 0,00001$ para o cenário {[*avião*, 30x30, 512], [*praia*, 5x5, 128], [*construção*, 5x5, 256], [*carro*, 25x25, 512], [*campo*, 5x5, 128], [*fogos de artifício*, 30x30, 256], [*flor*, 25x25, 256], [*corrida de motos*, 10x10, 256], [*moto*, 15x15, 512], [*montanha*, 15x15, 256], [*pôr do sol*, 5x5, 128], [*árvore*, 5x5, 256]}. A matriz confusão desse cenário é apresentado na Figura 5.22. Mesmo variando as partições de treinamento e as taxas de convergência ϵ , não houve nenhuma outra configuração de experimento que melhorasse a acurácia (item 1.1.1). Entretanto, utilizando a partição C como treinamento com $\epsilon = 0,1$, a acurácia, detalhada na Figura 5.23, caiu para 0,52 (item 1.1.2).

A Figura 5.24 mostra um comparativo das taxas de acurácia alcançadas nas variações do item 1.a. Ao contrário dos resultados apresentados na Figura 5.16, as

		Classe Predita											
		Avião	Praia	Construção	Carro	Campo	Fogos de Artifício	Flor	Corrida de Moto	Moto	Montanha	Pôr do Sol	Árvore
Classe Verdadeira	Avião	0	0,34	0	0	0,52	0,14	0	0	0	0	0	0
	Praia	0	0,10	0	0	0,88	0,02	0	0	0	0	0	0
	Construção	0	0,04	0	0	0,64	0,32	0	0	0	0	0	0
	Carro	0	0,12	0	0	0,24	0,64	0	0	0	0	0	0
	Campo	0	0,04	0	0	0,96	0	0	0	0	0	0	0
	Fogos de Artifício	0	0	0	0	0,02	0,98	0	0	0	0	0	0
	Flor	0	0	0	0	0,76	0,24	0	0	0	0	0	0
	Corrida de Moto	0	0,10	0	0	0,52	0,38	0	0	0	0	0	0
	Moto	0	0,10	0	0	0	0,90	0	0	0	0	0	0
	Montanha	0	0,20	0	0	0,56	0,24	0	0	0	0	0	0
	Pôr do Sol	0	0,02	0	0	0,70	0,28	0	0	0	0	0	0
	Árvore	0	0,02	0	0	0,78	0,20	0	0	0	0	0	0

Figura 5.18: Matriz confusão do pior resultado (item 1.1.4) obtido, variando os compressores, pelo melhor cenário de 12 classes, treinado e testado na partição A com $\epsilon = 0,00001$, quando utilizada a compressão *bzip2* (Acurácia = 0,17).

		Classe Preditá											
		Avião	Praia	Construção	Carro	Campo	Fogos de Artifício	Flor	Corrida de Moto	Moto	Montanha	Pôr do Sol	Árvore
Classe Verdadeira	Avião	0,06	0,29	0,06	0,04	0,05	0	0,22	0,02	0,01	0,14	0,01	0,09
	Praia	0,02	0,40	0,04	0	0,15	0	0,26	0,01	0	0	0,08	0,04
	Construção	0	0	0,66	0,06	0	0	0,13	0,01	0	0,02	0	0,12
	Carro	0,02	0,02	0,27	0,35	0,01	0,01	0,04	0,12	0,02	0,06	0,01	0,06
	Campo	0	0,19	0,05	0	0,28	0	0,32	0	0	0,01	0,14	0,01
	Fogos de Artifício	0	0	0	0,24	0	0,48	0,07	0,02	0	0,06	0,13	0
	Flor	0	0	0,09	0,02	0	0	0,59	0,05	0	0,01	0,02	0,21
	Corrida de Moto	0,01	0,01	0,36	0,22	0,02	0	0,09	0,08	0,01	0,04	0	0,14
	Moto	0	0,02	0,27	0,15	0	0	0	0,19	0,02	0,33	0	0,01
	Montanha	0,01	0,16	0,16	0,11	0	0	0,33	0,01	0	0,16	0	0,05
	Pôr do Sol	0	0,05	0,11	0,12	0,09	0	0,27	0,01	0	0	0,27	0,08
	Árvore	0	0	0,35	0,05	0,01	0	0,44	0	0	0	0	0,15

Figura 5.19: Matriz confusão do pior resultado (item 1.2) obtido pelas medidas de compressão em cenários com 12 classes, treinado e testado na partição C com $\epsilon = 0,001$ e ZIP, conforme Tabela B.15 (ID: 221, acurácia: 0,2931, precision: 0,3521, recall: 0,2931, F1 score: 0,2732).

		Classe Preditá						
		Praia	Construção	Fogos de Artifício	Flor	Corrida de Motos	Moto	Árvore
Classe Verdadeira	Praia	0,98	0	0	0	0,02	0	0
	Construção	0	0,82	0	0,08	0	0	0,10
	Fogos de Artifício	0	0	0,92	0,06	0	0	0,02
	Flor	0	0	0	1,00	0	0	0
	Corrida de Motos	0	0	0	0,06	0,90	0,04	0
	Moto	0	0	0	0	0,06	0,94	0
	Árvore	0	0	0	0,06	0	0	0,94

Figura 5.20: Matriz confusão do melhor resultado (item 2.1) obtido pelas medidas de compressão em cenários com 7 classes, treinado e testado na partição A com $\epsilon = 0,01$ e bzip2, conforme Tabela B.14 (ID: 24, acurácia: 0,9286, precision: 0,9369, recall: 0,9286, F1 score: 0,9294).

		Classe Preditá						
		Avião	Campo	Fogos de Artifício	Flor	Moto	Pôr do Sol	Árvore
Classe Verdadeira	Avião	0,60	0,08	0,02	0,16	0,08	0,04	0,02
	Campo	0,08	0,58	0	0,16	0,02	0,02	0,14
	Fogos de Artifício	0	0	0,86	0,06	0	0	0,08
	Flor	0,02	0	0,02	0,62	0,02	0,04	0,28
	Moto	0,14	0	0	0	0,86	0	0
	Pôr do Sol	0	0	0	0,34	0	0,62	0,04
	Árvore	0	0	0	0,06	0,04	0	0,90

Figura 5.21: Matriz confusão do pior resultado (item 2.2) obtido pelas medidas de compressão em cenários com 7 classes, treinado e testado na partição A com $\epsilon = 0,00001$ e ZIP, conforme Tabela B.14 (ID: 108, acurácia: 0,7200, precision: 0,7589, recall: 0,7200, F1 score: 0,7253).

		Classe Predita											
		Avião	Praia	Construção	Carro	Campo	Fogos de Artifício	Flor	Corrida de Moto	Moto	Montanha	Pôr do Sol	Árvore
Classe Verdadeira	Avião	0,66	0,08	0	0	0,02	0,02	0,04	0,14	0	0,04	0	0
	Praia	0,30	0,54	0	0	0,02	0	0	0,04	0	0,04	0,06	0
	Construção	0	0	0,86	0,02	0	0	0	0,06	0	0,04	0	0,02
	Carro	0,06	0,02	0,02	0,20	0	0,04	0,06	0,24	0	0,30	0,04	0,02
	Campo	0	0,24	0	0	0,44	0	0	0	0	0,10	0,16	0,06
	Fogos de Artifício	0	0	0	0,02	0	0,80	0,02	0	0	0	0,16	0
	Flor	0,02	0	0	0,04	0	0	0,72	0,04	0	0,18	0	0
	Corrida de Moto	0	0	0	0,02	0	0	0	0,82	0	0,16	0	0
	Moto	0	0	0,04	0,02	0	0	0	0	0,94	0	0	0
	Montanha	0,08	0,06	0	0	0	0	0,02	0,10	0	0,68	0,04	0,02
	Pôr do Sol	0	0	0	0	0	0	0,12	0,04	0	0,08	0,76	0
	Árvore	0	0	0	0,04	0	0	0	0,02	0	0	0	0,94

Figura 5.22: Matriz confusão do melhor resultado (itens 1.1 e 1.1.1) obtido pela medida de entropia em cenários com 12 classes - incluindo variações de partição de treinamento e taxa de convergência ϵ - treinado e testado na partição A com $\epsilon = 0,00001$, conforme Tabela B.16 (ID: 93, acurácia: 0,6967, precision: 0,7260, recall: 0,6967, F1 score: 0,6911).

acurácias mostram crescimento em função da diminuição da taxa de convergência ϵ para as partições A e B, a única exceção para a partição C com $\epsilon = 0,00001$. Porém, esse comportamento não se repete em todos os casos individuais. Embora não possa ser sugerida uma taxa de convergência ϵ para entropia, como nas medidas de compressão, os dados da Tabela 5.8 e a Figura 5.24 indicam que, na média, os melhores resultados foram alcançados em $\epsilon = 0,0001$. Outra diferença é a taxa de acurácia maior ($\epsilon = 0,1$ e $\epsilon = 0,0001$) nos resultados utilizando o treinamento da partição B em relação as demais, não observada nos testes com os compressores. O pior resultado em cenários de 12 classes (item 1.2), utilizando a entropia, é reportado em amostras da partição C, treinada com $\epsilon = 0,00001$ e taxa de acurácia de 0,4049 (Figura 5.25).

		Classe Predita											
		Avião	Praia	Construção	Carro	Campo	Fogos de Artifício	Flor	Corrida de Moto	Moto	Montanha	Pôr do Sol	Árvore
Classe Verdadeira	Avião	0,54	0,30	0	0	0,04	0	0	0,02	0	0,04	0,06	0
	Praia	0,14	0,78	0	0	0,06	0	0	0	0	0	0	0,02
	Construção	0	0	0,20	0	0	0	0	0	0	0	0	0,80
	Carro	0	0,06	0,18	0	0	0	0	0,40	0	0,22	0,02	0,12
	Campo	0,02	0,28	0,02	0	0,42	0	0,02	0	0	0,06	0	0,18
	Fogos de Artifício	0	0	0	0,02	0	0,52	0	0,08	0	0	0,30	0,08
	Flor	0	0,06	0	0	0,06	0	0,24	0,10	0	0,36	0,10	0,08
	Corrida de Moto	0,02	0,18	0,02	0	0	0	0	0,56	0	0,16	0	0,06
	Moto	0,04	0	0,06	0	0	0	0	0	0,90	0	0	0
	Montanha	0,06	0,16	0,10	0	0	0	0	0,02	0	0,40	0,06	0,20
	Pôr do Sol	0	0,06	0,04	0,02	0,04	0	0	0	0	0,10	0,72	0,02
	Árvore	0	0	0,02	0	0	0	0	0,02	0	0	0	0,96

Figura 5.23: Matriz confusão do pior resultado (item 1.1.2) obtido no conjunto de teste A pela medida de entropia em cenários com 12 classes, variando as partições de treinamento e as taxas de convergência ϵ , treinado na partição C com $\epsilon = 0,1$ conforme Figura 5.24 (Acurácia = 0,52).

Taxa de acurácia do melhor cenário utilizando entropia para as partições e taxas de convergência ϵ

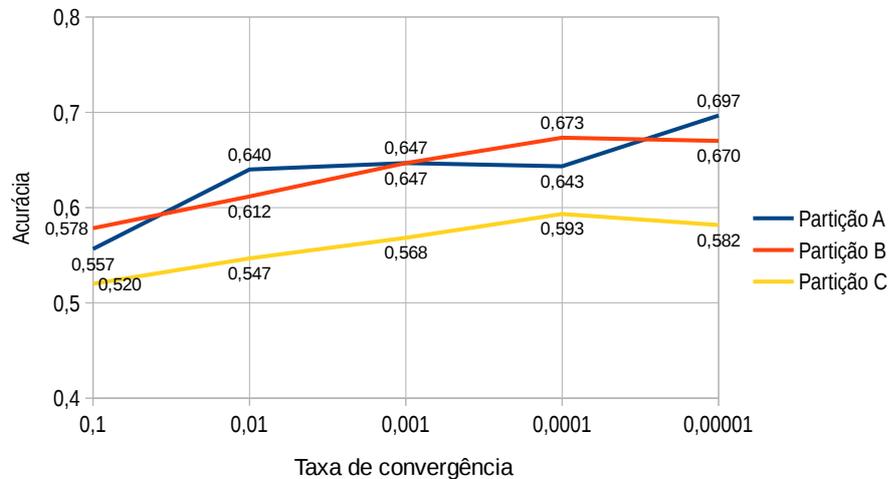


Figura 5.24: Taxa de acurácia da entropia comparada entre os treinamentos obtidos a partir das partições A, B e C e $\epsilon = \{0,1; 0,01; 0,001; 0,0001 \text{ e } 0,00001\}$ classificando o conjunto de teste da partição A.

A partir da Tabela 5.8 pode ser verificado que o melhor resultado para cenários de 7 classes (item 2.1) utilizando entropia, obteve 0,9029 de acurácia, treinado com $\epsilon = 0,1$ e testado na partição A. A matriz confusão desse resultado é apresentada na Figura 5.26 para o cenário $\{[avião, 5 \times 5, 256], [campo, 5 \times 5, 256], [fogos \ de \ artifício, 10 \times 10, 512], [flor, 15 \times 15, 512], [corrida \ de \ moto, 10 \times 10, 512], [moto, 10 \times 10, 512], [árvore, 5 \times 5, 512]\}$. Já no pior caso (item 2.2), apenas substituindo a classe *campo* pela classe *pôr do sol* e alterando a taxa de convergência $\epsilon = 0,001$, a acurácia diminuiu para 0,8029 e é detalhada na Figura 5.27.

		Classe Predita											
		Avião	Praia	Construção	Carro	Campo	Fogos de Artifício	Flor	Corrida de Moto	Moto	Montanha	Pôr do Sol	Árvore
Classe Verdadeira	Avião	0,53	0,16	0	0	0,08	0	0,02	0	0	0,12	0,08	0
	Praia	0,25	0,33	0	0	0,28	0	0	0	0	0,01	0,13	0
	Construção	0,02	0,04	0,06	0,16	0,16	0	0,05	0,20	0	0,04	0,05	0,22
	Carro	0,09	0,05	0	0,35	0,02	0,02	0,06	0,11	0	0,06	0,20	0,04
	Campo	0,04	0,13	0	0	0,56	0	0,01	0	0	0	0,26	0
	Fogos de Artifício	0	0	0	0,02	0	0,62	0,01	0	0	0	0,33	0,01
	Flor	0,01	0,05	0	0	0,02	0	0,35	0,04	0	0,01	0,34	0,18
	Corrida de Moto	0,09	0,12	0	0,15	0,01	0	0,01	0,44	0	0,04	0,08	0,06
	Moto	0	0	0,01	0,82	0	0	0	0	0,16	0	0	0
	Montanha	0,14	0,18	0	0,07	0,16	0	0,02	0,06	0	0,09	0,24	0,04
	Pôr do Sol	0	0,05	0	0,02	0,01	0	0,01	0,01	0	0,01	0,88	0
	Árvore	0,01	0,07	0,01	0,05	0,06	0	0,15	0,09	0	0	0,08	0,47

Figura 5.25: Matriz confusão do pior resultado (item 1.2) obtido pela medida de entropia em cenários com 12 classes, treinado e testado na partição C com $\epsilon = 0,00001$, conforme Tabela B.16 (ID: 242, acurácia: 0,4049, precision: 0,5028, recall: 0,4049, F1 score: 0,3852).

		Classe Predita						
		Avião	Campo	Fogos de Artifício	Flor	Corrida de Motos	Moto	Árvore
Classe Verdadeira	Avião	0,86	0,04	0,02	0,02	0,06	0	0
	Campo	0,06	0,84	0	0,04	0	0	0,06
	Fogos de Artifício	0	0	0,96	0,04	0	0	0
	Flor	0,02	0,04	0,02	0,88	0,04	0	0
	Corrida de Motos	0,06	0	0	0,08	0,84	0	0,02
	Moto	0,02	0	0	0	0	0,98	0
	Árvore	0	0,02	0	0,02	0	0	0,96

Figura 5.26: Matriz confusão do melhor resultado (item 2.1) obtido pela medida de entropia em cenários com 7 classes, treinado e testado na partição A com $\varepsilon = 0,1$, conforme Tabela B.16 (ID: 4, acurácia: 0,9029, precision: 0,9040, recall: 0,9029, F1 score: 0,9030).

		Classe Predita						
		Avião	Fogos de Artifício	Flor	Corrida de Motos	Moto	Pôr do Sol	Árvore
Classe Verdadeira	Avião	0,66	0,02	0,06	0,24	0	0,02	0
	Fogos de Artifício	0	0,82	0,08	0,02	0	0,04	0,04
	Flor	0,08	0,04	0,84	0,04	0	0	0
	Corrida de Motos	0	0	0,02	0,98	0	0	0
	Moto	0,02	0	0	0,02	0,96	0	0
	Pôr do Sol	0,02	0	0,32	0,24	0	0,42	0
	Árvore	0	0	0	0,06	0	0	0,94

Figura 5.27: Matriz confusão do pior resultado (item 2.2) obtido pela medida de entropia em cenários com 7 classes, treinado e testado na partição A com $\varepsilon = 0,001$, conforme Tabela B.16 (ID: 59, acurácia: 0,8029, precision: 0,8373, recall: 0,8029, F1 score: 0,7984).

Transformações e Processamentos de Imagens

A influência de transformações geométricas e de processamento em imagens em aplicações de visão computacional é amplamente investigada. As pesquisas envolvem o desenvolvimento de novos algoritmos e métodos de detecção e reconhecimento de objetos, extração de descritores e classificação de texturas e similaridade de imagens [98, 87, 54, 166]. Mesmo em técnicas de reconhecimento de padrões baseado em compressão de dados, é desafiador desenvolver métodos invariantes a essas influências que, em geral, representam as condições reais de funcionamento de uma aplicação [157, 23, 70, 71, 112, 19, 32, 117, 158, 25, 162, 38].

Para avaliar a resposta do RPA a essas influências, as imagens de teste da partição A foram classificadas nos melhores cenários com 12 classes discutidos na Seção 5.2.3. Utilizou-se as medidas de compressão *LZW+Huffman* (ID 106 da Tabela B.14) e entropia (ID 93 da Tabela B.16), após as imagens serem submetidas a:

- Ajuste de contraste: equalização de histograma e reescala de intensidade;
- Reflexão: eixos x e y ;
- Suavização linear: filtros 5×5 , 7×7 e 11×11 ;

- Suavização não-linear: filtros 5x5, 7x7 e 11x11;
- Ruído sal e pimenta: cobertura da imagem em 5%, 10% e 20%;
- Ruído Gaussiano: com $\mu = 0$ e variância (θ^2) em 0,01, 0,001 e 0,0001;
- Rotação: ângulos de 45°, 90°, 135°, 225°, 270° e 315°; e
- Escala: 50%, 75%, 125%, 150%, 200% e 300%.

A Figura 5.28 apresenta as taxas de acurácia obtidas para cada uma das transformações e suas variações. Observa-se na Figura 5.28(a) que, no ajuste de contraste com reescala de intensidade, a compressão e a entropia apresentaram reduções proporcionais de acurácia, mantendo acima de 0,5. Isso indica que ambas medidas de aleatoriedade se comportam de maneira similar sob esse processamento de realce. Na equalização de histograma, observou-se que, visualmente, as imagens perderam qualidade e ficaram mais escuras que as originais. Essa observação explica a baixa taxa de acurácia após esse processamento, em torno de 0,3. Na transformação de reflexão das imagens, os resultados apresentados na Figura 5.28(b) para o espelhamento horizontal (eixo x) são equivalentes às originais. Por se tratar de um problema de similaridade de cenas, em que a reflexão horizontal não ocasiona mudanças significativas no conteúdo visual, esse resultado é desejável e esperado. Entretanto, observa-se que no espelhamento vertical (eixo y), ambas medidas de compressão e entropia também mantiveram a taxa de acurácia. Mesmo no caso da medida de compressão, que é sensível à ordem da informação, o RPA é invariante à transformação de reflexão, bem como à rotação de 180° que é equivalente às operações de reflexões horizontal e vertical em conjunto.

O processamento de suavização linear (Figura 5.28(c)) foi realizado com filtros de coeficiente 1 e *bias* 0. O desempenho da classificação sofre influência para ambas medidas de aleatoriedade, já a partir do menor filtro (5x5). Entretanto, no caso da compressão, a taxa da acurácia se mantém estável para todos os filtros avaliados. Ao contrário, para a medida de entropia, a degradação progressiva da imagem, com *kernels* de convolução maiores, reduz o desempenho do RPA progressivamente. Na suavização linear (Figura 5.28(d)), o filtro da mediana com coeficiente 1 reduziu o desempenho para ambas medidas,. Entretanto, observa-se prevalência de melhores resultados utilizando a compressão *LZW+Huffman*.

Os piores resultados ocorreram na adição de ruído do tipo *sal e pimenta* (Figura 5.28(e)). Mesmo com o menor percentual de ruído (5%), houve grande queda na taxa de acurácia, que se manteve estável com o aumento gradual do ruído utilizando a medida de compressão. Em contraponto, a adição de ruído gaussiano com $\mu = 0$ e variância (θ^2) em 0,0001 e 0,001 não influenciou na acurácia do RPA em relação às imagens originais. Porém, diminuiu consideravelmente em $\theta^2 = 0,01$ de forma equivalente, tanto para a entropia quanto para a compressão.

Desempenho do RPA sob transformações geométricas e de processamento de imagem

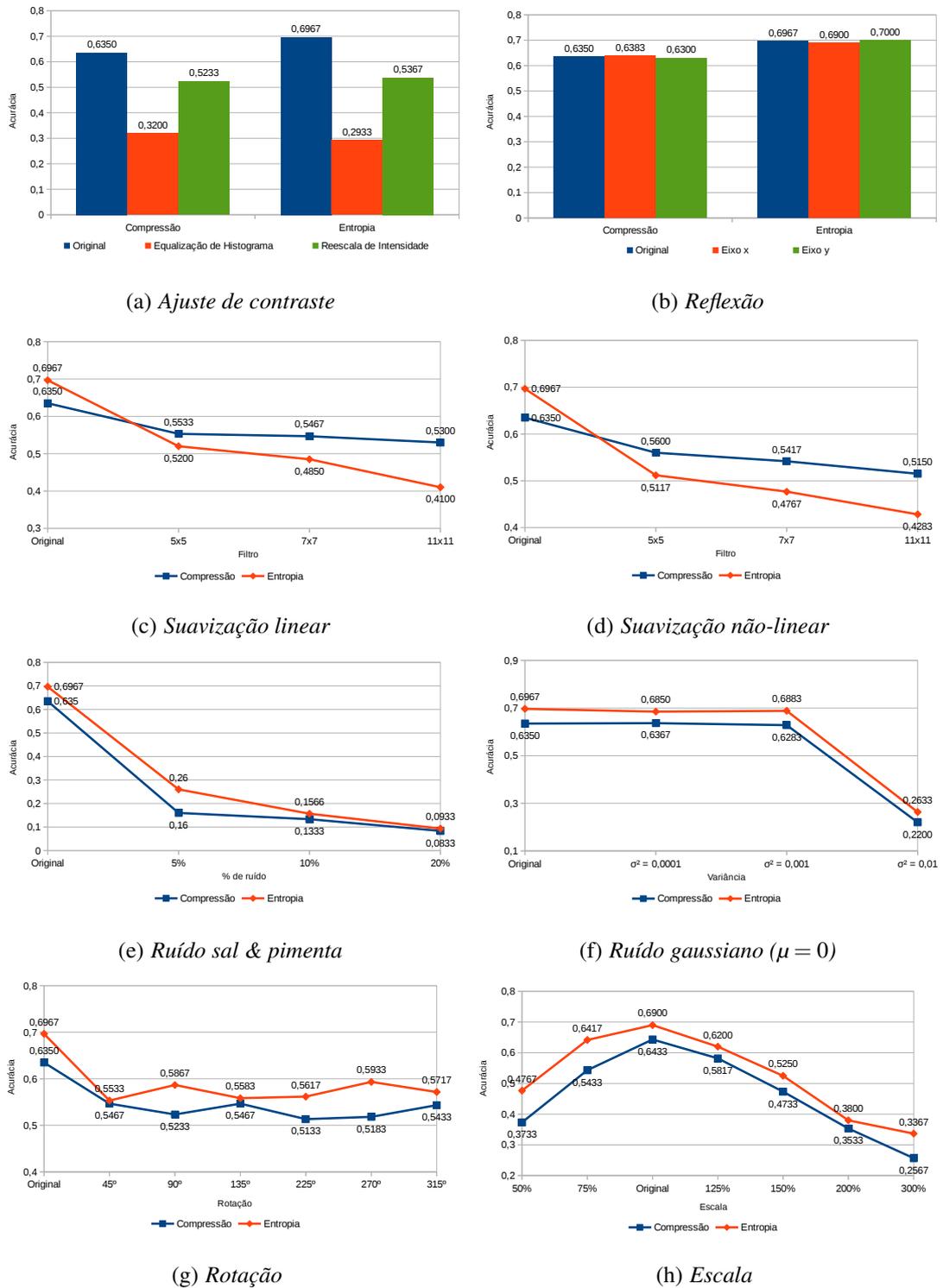


Figura 5.28: Acurácia do conjunto de teste sob transformações geométricas e processamentos de imagem utilizando as medidas de aleatoriedade de compressão LZW+Huffman e entropia

As transformações geométricas são responsáveis por grande parte dos problemas em aplicações reais. Na rotação (Figura 5.28(g)) e na escala (Figura 5.28(h)), nota-se que, para quaisquer variações, a entropia é superior à compressão. No rotacionamento das imagens, todo o conteúdo original foi preservado, sem adição ou supressão de informações. A área da imagem foi aumentada e as bordas ocasionadas pela rotação foram descartadas no processo de extração de janelas. Na medida de entropia, pode ser observado que para ângulos agudos e obtusos a acurácia é inferior em relação aos ângulos retos. Na medida de compressão essa relação é inversa, com os piores resultados observados para os ângulos retos e os melhores para os ângulos agudos e obtusos. Na transformação de escala (Figura 5.28(h)) o desempenho do RPA é influenciado tanto pela redução quanto pela ampliação da imagem, sendo mais sensível à redução.

Comparativo com DAMICORE e k-NN

Conforme discutido na Seção 3.2, o DAMICORE [140] e o CompLearn [34] são *toolkits* que promoveram significativos avanços nas pesquisas em agrupamento de dados. Esses métodos, segundo [131], são baseados na capacidade das técnicas de compressão de coletar conhecimento à medida que avança no processo de compressão de dados. As contribuições dessas ferramentas são tanto do ponto de vista da metodologia quanto na disponibilização do pacote compilado e/ou código-fonte.

Nessa seção, apresenta-se um comparativo de desempenho entre o DAMICORE, o k-NN e o RPA, utilizando a mesma base de dados e partição dos conjuntos de treinamento e teste. Por se tratar de um método de clusterização, a medida G [8], apresentada na Seção 3.3, foi utilizada para calcular a capacidade de generalização das árvores geradas pelo DAMICORE.

Os valores G das árvores geradas pelo DAMICORE utilizando 3 esquemas de compressão, PPMd, *bzip2* e *gzip*, são apresentados na Tabela 5.9. O melhor desempenho foi observado para o PPMd ($G = 0,6818$) e o pior para o *gzip* ($G = 0,4252$). No geral, o conjunto de treinamento obteve menor desempenho. Com o *dataset* (teste + treinamento) foram observadas maiores taxas para todos os esquemas de compressão. O PPMd foi capaz de gerar modelos de agrupamento com G equivalentes à acurácia do

Tabela 5.9: Valores da medida de generalização G das clusterizações realizadas pelo DAMICORE para os conjuntos de teste e treinamento da partição A utilizando diferentes compressores.

	PPMd	<i>bzip2</i>	<i>gzip</i>
Treinamento	0,6105	0,5238	0,4252
Teste	0,6293	0,5646	0,4371
Teste + Treinamento	0,6818	0,5875	0,4790

RPA, utilizando as medidas de entropia (Acurácia = 0,6967, Tabela B.16) e de compressão (Acurácia = 0,6350, Tabela B.14).

O RPA também foi comparado com o classificador k-NN, por ser um algoritmo com funcionamento semelhante, baseado na distância euclidiana entre os protótipos das classes e as amostras de teste. As vantagens do k-NN apontadas por [49] são: simplicidade, facilidade de entendimento e implementação, ausência da necessidade de intensa parametrização, pouco esforço na etapa de treinamento, eficiência na classificação de eventos raros e suporte a problemas multi-classe. Apesar dessa semelhança, o k-NN concentra a maior parte do esforço computacional na etapa de predição. O RPA, possui uma etapa de treinamento responsável por construir o modelo das classes em forma de *codebooks*. Nessa etapa de treinamento, o RPA visa construir a representação das classes conservando características predominantes e eliminando detalhes. Portanto, é caracterizada por um processo de perda da informação irrelevante (caso 1). Na etapa de predição, o RPA segmenta as imagens de teste em janelas e constrói aproximações para todas as classes do problema (caso 2). Visando comparar os resultados, o k-NN foi implementado para simular esses dois casos: 1) redução da informação irrelevante (redimensionamento das imagens) e 2) avaliação de informação local das cenas (classificação por janelas). Mantendo as mesmas condições experimentais, os conjuntos de treinamento e teste foram os mesmos utilizados nos resultados reportados para o RPA.

No primeiro caso, as imagens foram reescaladas para os formatos 5x5, 10x10, 15x15, 20x20, 25x25 e 30x30 pixels. Em seguida foram linearizadas para vetores U -dimensionais. Os resultados de acurácia são apresentados na Figura 5.29 para valores de k -vizinhos = [3, 5, 7, 9, 11, 15, 17, 19, 21]. Não houve grandes variações de acurácia, tanto em relação ao tamanho da imagem quanto o número de k -vizinhos. Entretanto, observa-se

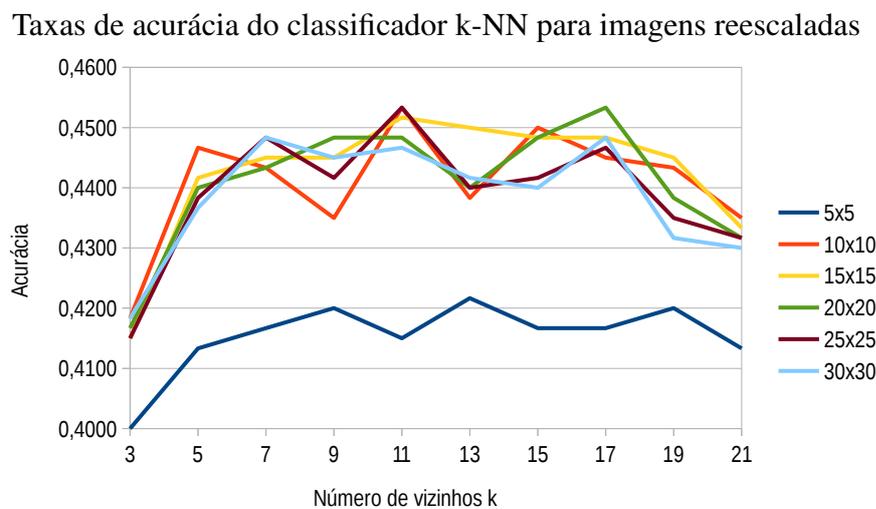


Figura 5.29: Taxas de acurácia obtidas pelo classificador k-NN aplicado às imagens da partição A em diferentes escalas e valores de vizinhança k.

que para o formato 5x5 a acurácia é inferior aos demais formatos, que alternam os picos de acurácia em função dos k -vizinhos. O melhor resultado alcançou 0,4533 de acurácia para os pares (formato, k -vizinhos): (10x10, 11), (20x20, 17) e (25x25, 11); enquanto o pior resultado foi 0,4 de acurácia observado para (5x5, 3).

No segundo caso, as imagens foram divididas em janelas, também nos tamanhos 5x5, 10x10, 15x15, 20x20, 25x25 e 30x30 pixels. Dada uma amostra de teste, suas janelas foram classificadas individualmente, com os mesmos valores de k -vizinhos do experimento anterior. Por votação majoritária, essas janelas receberam o rótulo da classe correspondente com vizinhos mais próximos. Ao final, foi realizada uma segunda votação majoritária observando a quantidade de janelas atribuídas a cada classe para se determinar o rótulo predominante da cena. Os resultados apresentados na Figura 5.30 mostram que o melhor cenário de classificação é com janelas 5x5, alcançando acurácia de 0,5783 para $k = 19$. Os demais tamanhos de janelas oscilaram em torno de 0,45 de acurácia, com tendência de redução à medida que o número de k -vizinhos aumentava, ao passo que para janelas 5x5 o k -NN ficava mais eficiente.

Mesmo com o melhor resultado de acurácia em 0,5783 alcançado pelo k -NN, dividindo as imagens em janelas, o desempenho do RPA é superior em todos os casos. Isso mostra que o processo de quantização foi capaz de tornar as imagens de teste mais discrimináveis. Além disso, a prototipação das imagens de treinamento torna a etapa de inferência mais rápida pois reduz o volume de dados das representações das classes.

Taxas de acurácia do classificador k -NN para imagens divididas em janelas

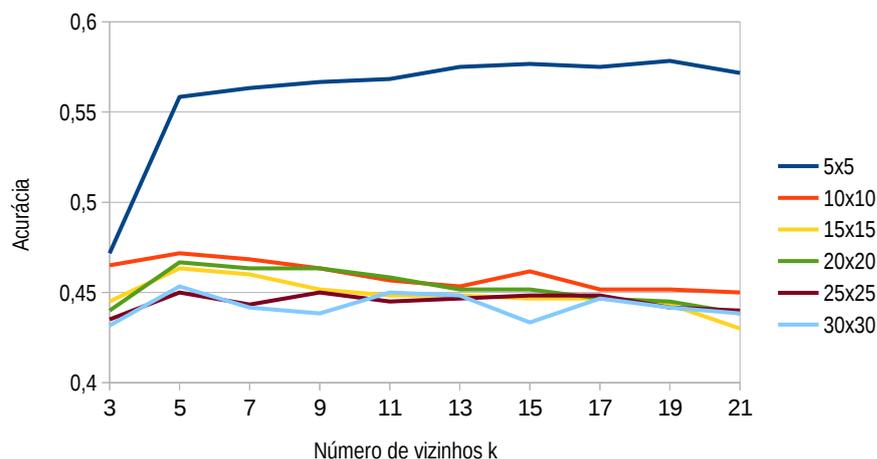


Figura 5.30: Taxas de acurácias obtidas pelo classificador k -NN aplicado às imagens da partição A divididas em janelas de diferentes tamanhos e valores de vizinhança k .

5.2.4 Escopo e Restrições do Método

Como demonstrado experimentalmente na seção anterior, a aproximação entre as imagens de teste e o conjunto de treinamento no k-NN não é suficiente para atingir as taxas de acurácia do RPA. Acredita-se que a chave para o melhor desempenho está na simulação do mecanismo de percepção do córtex visual, discutido na Seção 2.2.

De acordo com [59], a psicologia relaciona nossa percepção natural de aleatoriedade com a teoria matemática da complexidade algorítmica de Kolmogorov: quanto mais complexo o estímulo, mais aleatório será percebido. Em [60], os autores estabelecem que a relação entre quantidade e variabilidade de informação pode ser entendida como o parâmetro da complexidade visual. Portanto, o RPA tem como base a transformação da informação visual presente nas cenas: na quantização convergente, amplia a variabilidade das informações discriminantes da classe; na quantização divergente, aumenta a redundância. Entretanto, mesmo aumentando a complexidade da informação, o RPA preserva os elementos visuais fundamentais para a representação da classe verdadeira.

Segundo [65], na abordagem da Distorção Representacional (RD - *Representational Distortion*), a similaridade entre dois objetos pode ser modelada como uma *função de complexidade responsável por distorcer ou transformar um objeto em outro*. Assim, quanto menor o esforço requerido pela transformação de distorção para transitar entre duas representações, maior a similaridade entre os objetos. Essa é uma das premissas de operação do RPA. Preservando as características predominantes da classe, no processo de quantização convergente, a aproximação da imagem de teste para a classe verdadeira se torna menos dispendiosa. Essa aproximação para as demais classes exige mais distorção entre as representações. Em outras palavras, a conservação da complexidade de uma imagem de teste permite que uma função de transformação para sua classe verdadeira produza a menor distorção possível.

Do ponto de vista dos aspectos psicológicos [7, 120, 47, 59, 60, 99, 122] e neurofisiológicos [9, 32, 97, 77, 110], está claro que o funcionamento do RPA é fundamentado nas transformações de dados. Essas transformações têm como objetivo possibilitar a mensuração da complexidade da resposta de um estímulo visual, dado um modelo previamente conhecido. Na presença de informações visuais familiares ao modelo, estas tem suas estruturas preservadas o mais próximo possível de sua forma original. Informações visuais distantes do modelo são interpretadas como periféricas e tendem a ser simplificadas por não serem consideradas importantes.

Do ponto de vista matemático, o funcionamento do RPA é explicado por duas principais medidas no plano U -dimensional de projeção das amostras: interseção e dispersão. Para discutir o mecanismo de funcionamento e as restrições do RPA serão utilizadas quatro abordagens: Teoria Clássica dos Conjuntos, Autocorrelação Espacial Global [111], Teoria dos Conjuntos Aproximados [126] e Teoria da Comunicação [144];

e uma medida de *overlap* baseada no espaço euclidiano, denominada *R-value* [119]. Em geral, a eficiência do RPA está diretamente ligada às premissas:

- Maximização das distâncias extraclasse das amostras de treinamento;
- Maximização da interseção intraclasse das amostras de teste e minimização da interseção extraclasse;
- Similaridade entre a dispersão das amostras de teste e as amostras de treinamento de sua classe verdadeira.

Quanto menor é a interseção entre as classes, menos o resultado da classificação é dependente da relação de ordem e da distância entre as amostras de treinamento e teste.

Seja \mathbf{x} a projeção no espaço U -dimensional de uma janela de teste. $\mathcal{D}(\omega_w)$ representa o conjunto de protótipos da classe verdadeira, $\mathcal{D}(\omega_m)$ o conjunto de protótipos de qualquer outra classe e \mathcal{J} é o conjunto de janelas de teste. No caso trivial de separação binária entre as classes $\mathcal{D}(\omega_w)$ e $\mathcal{D}(\omega_m)$ deseja-se que

$$\{\mathbf{x} : \mathbf{x} \in \mathcal{D}(\omega_w) \wedge \mathbf{x} \in \mathcal{D}(\omega_m)\} = \emptyset, \quad (5-1)$$

portanto, $\mathcal{D}(\omega_w) \cap \mathcal{D}(\omega_m) = \emptyset$. Porém, a forma própria de extração de características e a alta dimensionalidade do espaço não favorecem esse caso ideal. Isso é esperado para qualquer problema não-trivial de reconhecimento de padrões. Assim, para maximizar o desempenho do RPA, deseja-se que a diferença espacial complementar de $\mathcal{D}(\omega_w)$ em relação a \mathcal{J} ,

$$\{\mathbf{x} : \mathbf{x} \in \mathcal{J} \wedge \mathbf{x} \notin \mathcal{D}(\omega_w)\} = \emptyset, \quad (5-2)$$

portanto $\mathcal{J} - \mathcal{D}(\omega_w) = \emptyset$, ao passo que, também, $\mathcal{J} \cap \mathcal{D}(\omega_m) = \emptyset$. Entretanto, admite-se que $\mathcal{J} \subset \mathcal{D}(\omega_w)$ ou $\mathcal{D}(\omega_w) \subset \mathcal{J}$, com a diferença simétrica, em um cenário de classificação binária, seja

$$\mathcal{D}(\omega_w) \triangle \mathcal{D}(\omega_m) \gg \mathcal{D}(\omega_w) \cap \mathcal{D}(\omega_m) \quad (5-3)$$

e

$$\mathcal{J} \cap \mathcal{D}(\omega_w) \approx \mathcal{J} \cup \mathcal{D}(\omega_w). \quad (5-4)$$

Admitindo ser uma metodologia de aproximação de protótipos para a representação que melhor conserve a complexidade do estímulo visual, o RPA é baseado na Lei de Tobler [155]: "*Todas as coisas estão relacionadas com todas as outras, mas coisas próximas estão mais relacionadas do que coisas distantes*". Essa fundamentação explica que, mesmo no caso de alta interseção entre os modelos das classes, é possível aproximar uma imagem de teste para sua classe verdadeira com certa precisão. Isso é possível em razão do princípio da Autocorrelação Espacial Global Positiva (Coeficiente I de Moran [111]). Segundo [93] "*as medidas de autocorrelação espacial procuram lidar*

simultaneamente com similaridades na localização dos objetos espaciais e de seus atributos". Portanto, a proposta tem como premissa que objetos similares em localização também são similares em atributos [93]. Nessa perspectiva, estabelece-se a analogia em que os objetos da definição de [93] são os componentes visuais das cenas (janelas) e os atributos são os pixels. Vale lembrar que esses componentes são imutáveis no contexto do estímulo visual, mas podem ter uma representação diferente no plano U -dimensional, com projeção dependente da forma como foram linearizados.

Complementarmente, a Teoria dos Conjuntos Aproximados proposta por [126] traz elementos para compreensão do mecanismo da aproximação entre as janelas e os protótipos. Segundo [127], os conjuntos aproximados constituem uma ferramenta para tratar problemas que envolvam imprecisão e incerteza. Suas possíveis aplicações incluem: redução de dados, estimativa da significância dos dados, classificação aproximada, similaridade entre os dados e descoberta de padrões e de relações de causa e efeito [127]. Em essência, a teoria se baseia em aproximar um subconjunto do domínio em conjuntos exatos denominados *aproximação inferior* ou *aproximação superior* [17]. Nesse contexto, pressupõe-se que todo objeto do universo \mathcal{U} carrega certa quantidade de informação, encontrada em alguns atributos utilizados para descrevê-lo [17].

De acordo com [52], do ponto de vista matemático, destaca-se a simplicidade da abordagem dos conjuntos aproximados, pois esta depende apenas da manipulação de conjuntos finitos, da *cardinalidade* e das *relações de equivalência*. No contexto da presente proposta, as relações de equivalência são obtidas pela quantização vetorial, que condiciona os dados a uma relação de indiscernibilidade nas janelas mais próximas de um mesmo *codevector*. A operação de quantização, de fato, realiza um processo de discretização dos dados que tem, como consequência, a simplificação. Em [17], os autores afirmam que essa redução induz à criação das *classes de equivalência*. Assim a operação de discriminação das amostras passa a ser realizada considerando apenas um elemento da partição induzida, ao invés de toda a partição.

O conceito de indução de classes de equivalência pode ser relacionado com o Teorema de Codificação da Fonte de Shannon [144]. Considere uma mensagem \mathcal{M} , um alfabeto médio \mathcal{A}_1 , obtido a partir de conteúdos semelhantes à mensagem, e um alfabeto \mathcal{A}_2 , obtido a partir de mensagens dissemelhantes. A mensagem \mathcal{M} pode ser reconstruída preservando mais informação utilizando o alfabeto \mathcal{A}_1 do que o alfabeto \mathcal{A}_2 . Esse é o caso das quantizações convergente e divergente, respectivamente. Com o alfabeto \mathcal{A}_1 , o nível de coerência pode ser preservado dentro de certos limites, admitindo perdas. Com o alfabeto \mathcal{A}_2 a informação é degradada de maneira mais severa com a substituição de símbolos díspares.

Para mensurar a interseção entre as representações das classes e as amostras de teste, foi utilizada a medida *R-value* [119], apresentada na Seção 2.5. O *R-value* é capaz

de capturar o grau de interseção de conjuntos espacialmente dispostos em um plano U -dimensional, dada a dispersão de seus elementos. Considerando uma amostra de teste, se o R -value entre suas janelas e sua classe verdadeira for maior que o R -value entre as demais classes, o RPA é capaz de aproximar a representação com eficiência. Assim, a quantização vetorial produzirá uma reconstrução da cena com mais complexidade e permitirá a correta classificação.

A Tabela 5.10 sumariza os valores $R(\mathcal{U})$ (*overlap* de todo o *dataset*) dos dados de treinamento da partição A, com *codebook* de 512 *codevectors* e taxa de convergência $\epsilon = 0,00001$, para todos os tamanhos de janelas do treinamento. O detalhamento dos valores $R(\omega_w, \omega_m)$ (*overlap* par-a-par de classes) e $R(\omega_w)$ (*overlap* da classe) para a janela 30x30 e parâmetros $k = 21$ e $\theta = 10$ são apresentados na Tabela 5.11. Como pode ser observado na Tabela 5.10, independente dos parâmetros, a disposição das amostras no plano U -dimensional induz as classes à uma interseção quase total. Mesmo sendo bem distintas em relação ao conteúdo visual, as amostras tendem a compartilhar o espaço entre as classes, tornando o problema de difícil discriminação entre os dados [119]. Essa condição pode ser atribuída à duas principais causas:

1. a fonte das características de cada amostra são os pixels extraídos das janelas. Fora de contexto, não representam nenhuma informação visual para a cognição humana e tão pouco se correlacionam com as demais amostras de sua respectiva classe;
2. ao não utilizar nenhuma técnica de seleção explícita de características, algumas amostras podem ser *outliers* ou conter atributos irrelevantes que contribuem para a maldição da dimensionalidade.

Observando a interseção entre os pares de classes na Tabela 5.11, nota-se que a maior interseção ocorreu entre as classes *montanha* e *corrida de motos* (0,6387) e a menor entre as classes *moto* e *fogos de artifício* (0,2412). A média de interseção é 0,4463,

Tabela 5.10: R -value $R(\mathcal{U})$ do conjunto de treinamento da partição A com 512 *codevectors* por classe.

	5x5	10x10	15x15	20x20	25x25	30x30
$k = 3, \theta = 1$	1	0,9995	0,9985	0,9982	0,9969	0,9917
$k = 5, \theta = 2$	1	1	0,9995	0,9980	0,9984	0,9940
$k = 7, \theta = 3$	0,9998	1	0,9998	0,9997	0,9985	0,9966
$k = 9, \theta = 4$	0,9998	1	1	0,9997	0,9993	0,9969
$k = 11, \theta = 5$	0,9998	1	1	0,9997	0,9998	0,9969
$k = 13, \theta = 6$	1	1	1	0,9998	0,9998	0,9979
$k = 15, \theta = 7$	1	1	1	1	0,9997	0,9985
$k = 17, \theta = 8$	1	1	1	1	0,9998	0,9990
$k = 19, \theta = 9$	1	1	1	1	0,9998	0,9993
$k = 21, \theta = 10$	1	1	1	1	0,9998	0,9993

Tabela 5.11: R -values $R(\omega_w, \omega_m)$ e $R(\omega_w)$ do conjunto de treinamento da partição A a partir de janelas 30×30 com 512 codevectors por classe e parâmetros $k = 21$ e $\theta = 10$

$R(\omega_w, \omega_m)$	avião	praia	construção	carro	campo	fogos de artifício	flor	corrida de motos	moto	montanha	pôr do sol	árvore
avião	-	0,5293	0,4531	0,4736	0,4639	0,2598	0,3994	0,5010	0,4521	0,5957	0,3789	0,4082
praia	0,5293	-	0,4043	0,4316	0,6045	0,2783	0,3691	0,4551	0,3359	0,5039	0,4824	0,4189
construção	0,4531	0,4043	-	0,6133	0,4072	0,3447	0,4590	0,5986	0,4170	0,5381	0,4111	0,5928
carro	0,4736	0,4316	0,6133	-	0,3828	0,4092	0,4834	0,6641	0,4033	0,5850	0,5098	0,4824
campo	0,4639	0,6045	0,4072	0,3828	-	0,2939	0,3672	0,4092	0,3340	0,4248	0,4893	0,4053
fogos de artifício	0,2598	0,2783	0,3447	0,4092	0,2939	-	0,4209	0,3984	0,2412	0,3633	0,4150	0,3545
flor	0,3994	0,3691	0,4590	0,4834	0,3672	0,4209	-	0,5547	0,3242	0,5098	0,5576	0,6143
corrida de motos	0,501	0,4551	0,5986	0,6641	0,4092	0,3984	0,5547	-	0,3740	0,6387	0,4814	0,5430
moto	0,4521	0,3359	0,4170	0,4033	0,3340	0,2412	0,3242	0,3740	-	0,3730	0,2939	0,3516
montanha	0,5957	0,5039	0,5381	0,5850	0,4248	0,3633	0,5098	0,6387	0,3730	-	0,4648	0,5039
pôr do sol	0,3789	0,4824	0,4111	0,5098	0,4893	0,4150	0,5576	0,4814	0,2939	0,4648	-	0,4551
árvore	0,4082	0,4189	0,5928	0,4824	0,4053	0,3545	0,6143	0,543	0,3516	0,5039	0,4551	-
$R(\omega_w)$	1	1	1	1	1	1	1	1	1	1	0,9922	1

sugerindo que quase a metade das amostras de uma classe qualquer ocupam o espaço de outra classe, quando o problema é de classificação binária. Entretanto, durante a etapa de *Validação*, observou-se elevadas taxas de acurácia, chegando a 1 em cenários de 2 classes. Os exemplos incluem a combinação de cenários entre a classe *fogos de artifício* e as classes *avião*, *praia*, *campo*, *montanha*, *corrida de motos*, *moto* e *árvore* com 0,3128 de interseção, em média; e entre a classe *moto* e as classes *flor*, *corrida de motos*, *pôr do sol* e *árvore* com 0,3359 de interseção média. Portanto, os resultados mostram que o RPA tem eficiência máxima, mesmo nos casos com até 1/3 de sobreposição entre as classes, em cenários de classificação binária.

A seguir, são exemplificadas situações de acerto e de erro do classificador proposto. Para cada medida de aleatoriedade, foram selecionados 4 casos, 2 de acertos e 2 de erros. São apresentadas imagens de teste que foram corretamente ou erroneamente classificadas com o máximo e o mínimo de diferença. A Figura 5.31 mostra as cenas avaliadas: na 1ª linha para compressão e na 2ª linha para entropia. Todas as visualizações das amostras foram geradas utilizando o *t-Distributed Stochastic Neighbor Embedding* (t-SNE) [161]. Segundo os autores, o t-SNE é uma técnica de transformação de dados de alta dimensionalidade em uma matriz de similaridades de duas ou três dimensões. É capaz de capturar a estrutura local dos dados U -dimensionais, além de revelar a estrutura global, como a presença de clusters em várias escalas [161].

Nos casos de acerto, ilustrados nas Figuras 5.32, 5.33, 5.36 e 5.37, do lado esquerdo estão as visualizações da imagem de teste e da classe verdadeira; do lado direito estão as visualizações da 2ª classe com maior aleatoriedade de acordo com a ordem de classificação. Todos os valores R -value estão detalhados na Tabela B.17; em negrito, os exemplos utilizados nas visualizações para todos os casos. Os aspectos de interseção e dispersão podem ser visualizados com mais facilidade nas Figuras 5.32 e 5.36. Elas mostram que, para a classe verdadeira e a imagem de teste, a interseção é maior e a

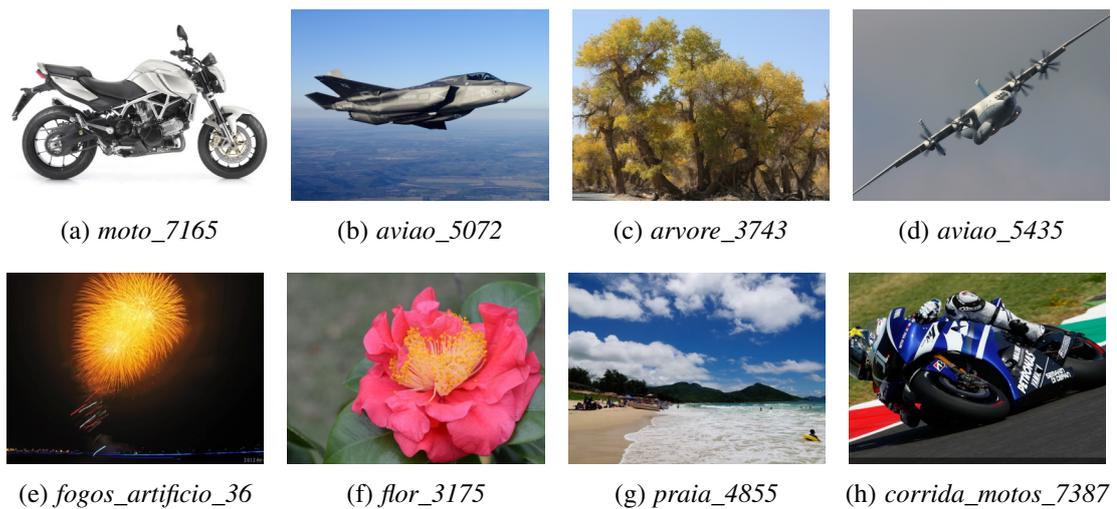


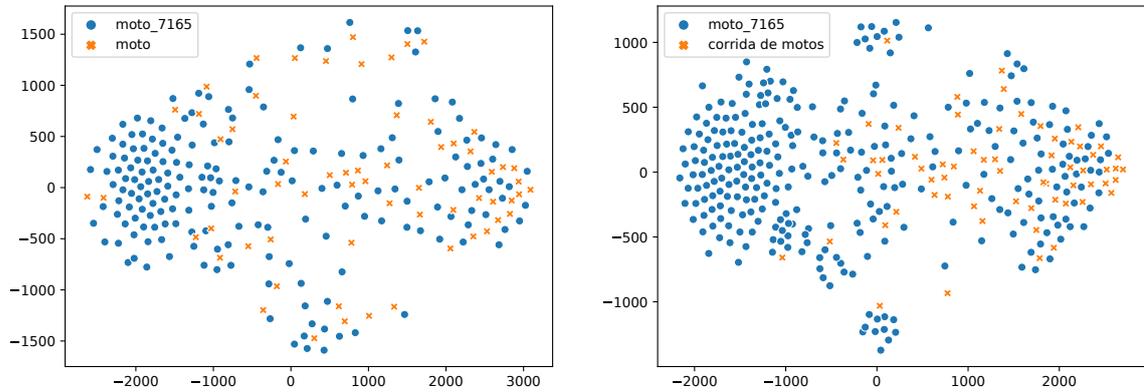
Figura 5.31: Imagens do conjunto de testes utilizadas para exemplificação e discussão do funcionamento do RPA

dispersão são mais semelhantes quando comparadas com a classe falsa (vide valores R -value). É importante destacar que as projeções da imagem de teste são diferentes pois são realizadas pelo t-SNE de acordo com a dimensionalidade dos dados de treinamento.

Nos casos de erro, ilustrados nas Figuras 5.34, 5.35, 5.38 e 5.39, do lado direito estão as visualizações da imagem de teste e da classe classificada erroneamente como verdadeira; do lado esquerdo estão as visualizações da classe verdadeira. A Tabela B.18 detalha os valores R -value dos exemplos. Pode ser observado que os erros de classificação, com diferença máxima ou mínima, estão relacionados com o baixo grau de *overlap* calculado pelo R -value entre a imagem de teste e sua classe verdadeira. Outro fator observado, que determina o erro do RPA, são concentrações de amostras em pequenas regiões. Visualmente, o t-SNE pode sugerir alto grau de interseção, que não é verificado quando o *overlap* é calculado, por exemplo, na Figura 5.34(a).

Em síntese, a quantização vetorial do RPA induz a formação de classes de equivalência para as amostras de uma imagem de teste. Assim, o *codevector* mais próximo das janelas concentra toda a informação que as descreve. Na quantização convergente, devido à interseção e à dispersão, ocorre melhor distribuição de janelas para os *codevectors*. Isso aumenta o tamanho do dicionário efetivamente utilizado (*codebooks*) para re-codificar o estímulo visual e, conseqüentemente, aumenta a complexidade intrínseca do objeto, tornando-o mais aleatório. Na quantização divergente, a concentração de informação original da imagem de teste em poucos *codevectors*, induzida pelas relações de equivalência, é maior.

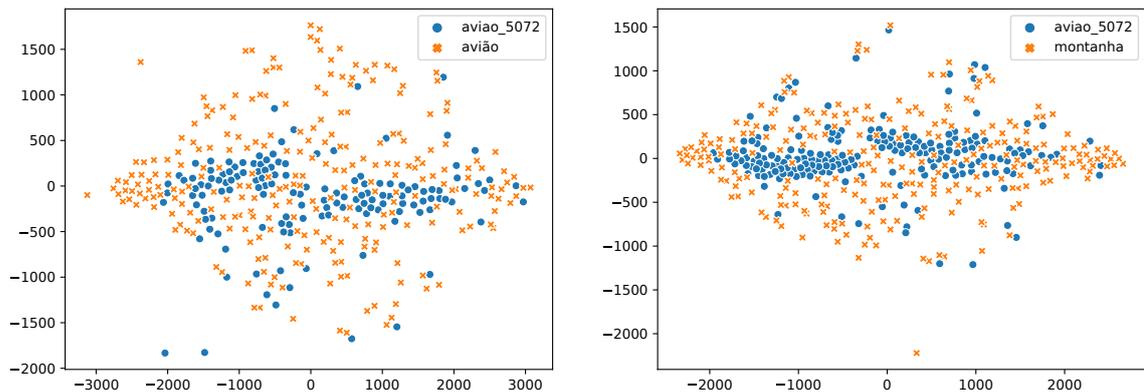
Caso de classificação correta com diferença máxima utilizando compressão



(a) Quantização convergente para (moto, 25x25, 64) com $R\text{-value} = 0,660156$ (b) Quantização divergente para (corrida de motos, 20x20, 64) com $R\text{-value} = 0,450550$

Figura 5.32: Visualização bidimensional com $t\text{-SNE}$ para o caso de classificação correta com diferença máxima (0,209607 para $k=3$ e $\theta=1$) entre a imagem de teste e suas classes verdadeira (a) e falsa (b) utilizando compressão LZW+Huffman de acordo com a Tabela B.17.

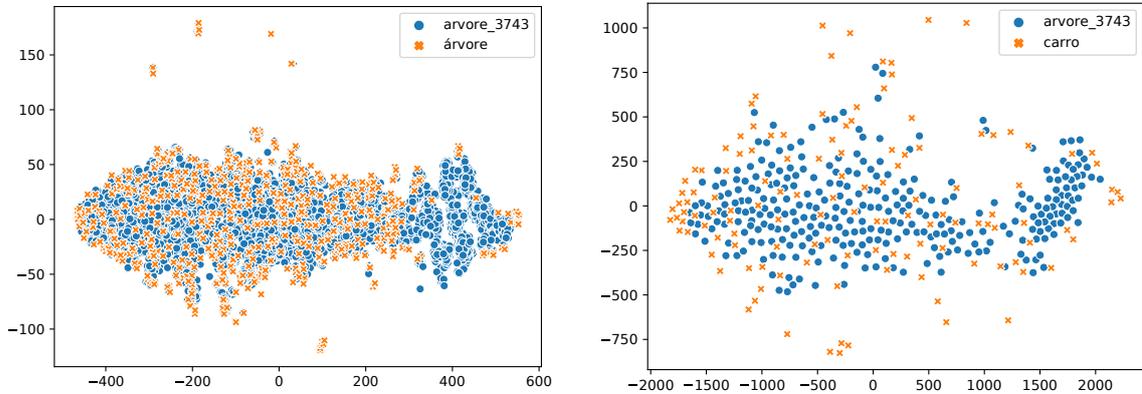
Caso de classificação correta com diferença mínima utilizando compressão



(a) Quantização convergente para (avião, 30x30, 256) com $R\text{-value} = 0,366162$ (b) Quantização divergente para (montanha, 25x25, 256) com $R\text{-value} = 0,366071$

Figura 5.33: Visualização bidimensional com $t\text{-SNE}$ para o caso de classificação correta com diferença mínima (0,000090 para $k=15$ e $\theta=7$) entre a imagem de teste e suas classes verdadeira (a) e falsa (b) utilizando compressão LZW+Huffman de acordo com a Tabela B.17.

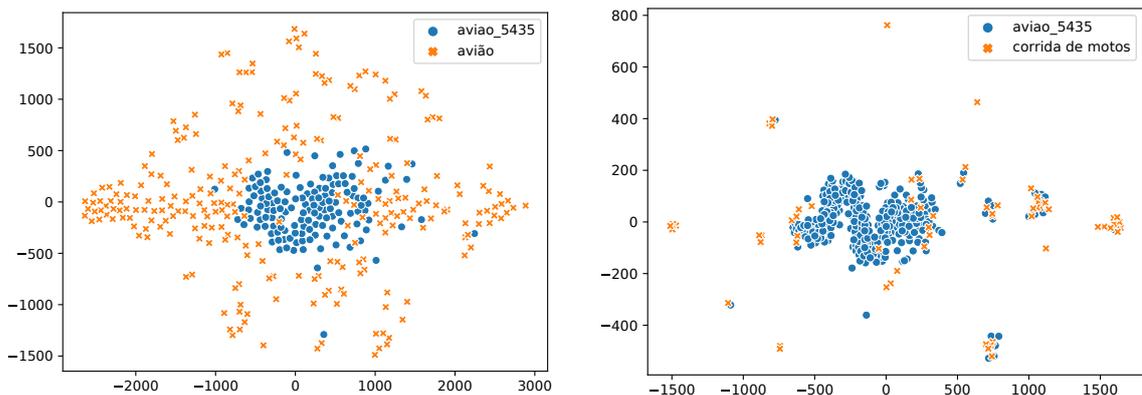
Caso de classificação incorreta com diferença máxima utilizando compressão



(a) Quantização convergente para (árvore, 5x5, 512) com R-value = 0,145896 (b) Quantização divergente para (carro, 20x20, 128) com R-value = 0,644860

Figura 5.34: Visualização bidimensional com t-SNE para o caso de classificação incorreta com diferença máxima (0,498963 para $k=7$ e $\theta=3$) entre a imagem de teste e suas classes verdadeira (a) e falsa (b) utilizando compressão LZW+Huffman de acordo com a Tabela B.17.

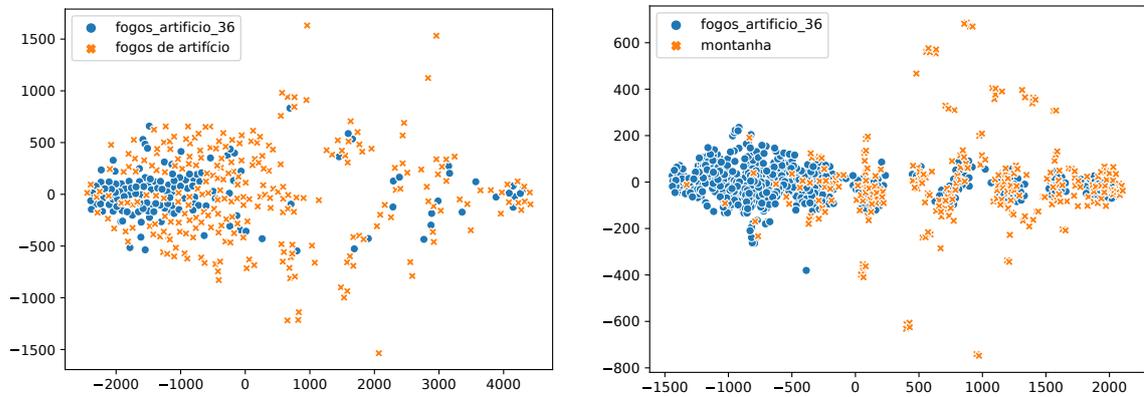
Caso de classificação incorreta com diferença mínima utilizando compressão



(a) Quantização convergente para (avião, 30x30, 256) com R-value = 0,126263 (b) Quantização divergente para (corrida de motos, 20x20, 64) com R-value = 0,129121

Figura 5.35: Visualização bidimensional com t-SNE para o caso de classificação incorreta com diferença mínima (0,002858 para $k=13$ e $\theta=6$) entre a imagem de teste e suas classes verdadeira (a) e falsa (b) utilizando compressão LZW+Huffman de acordo com a Tabela B.17.

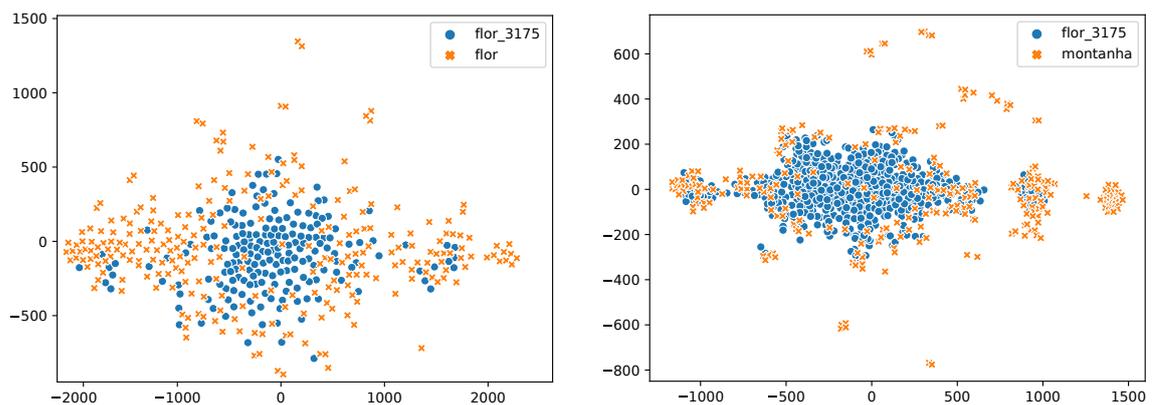
Caso de classificação correta com diferença máxima utilizando entropia



(a) *Quantização convergente para (fogos de artifício, 30x30, 256) com R-value = 0,603535* (b) *Quantização divergente para (montanha, 15x15, 256) com R-value = 0,310302*

Figura 5.36: *Visualização bidimensional com t-SNE para o caso de classificação correta com diferença máxima (0,293234 para $k=3$ e $\theta=1$) entre a imagem de teste e suas classes verdadeira (a) e falsa (b) utilizando entropia de acordo com a Tabela B.18.*

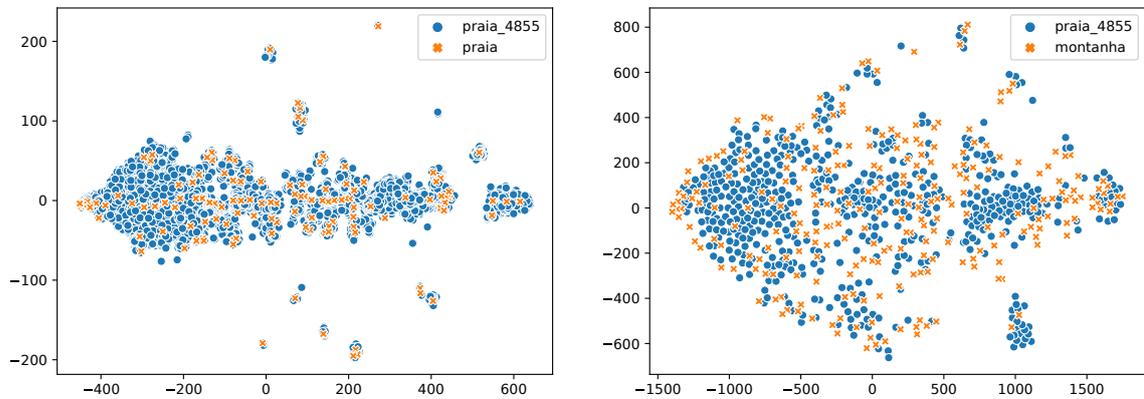
Caso de classificação correta com diferença mínima utilizando entropia



(a) *Quantização convergente para (flor, 25x25, 256) com R-value = 0,241071* (b) *Quantização divergente para (montanha, 15x15, 256) com R-value = 0,238693*

Figura 5.37: *Visualização bidimensional com t-SNE para o caso de classificação correta com diferença mínima (0,002378 para $k=19$ e $\theta=9$) entre a imagem de teste e suas classes verdadeira (a) e falsa (b) utilizando entropia de acordo com a Tabela B.18.*

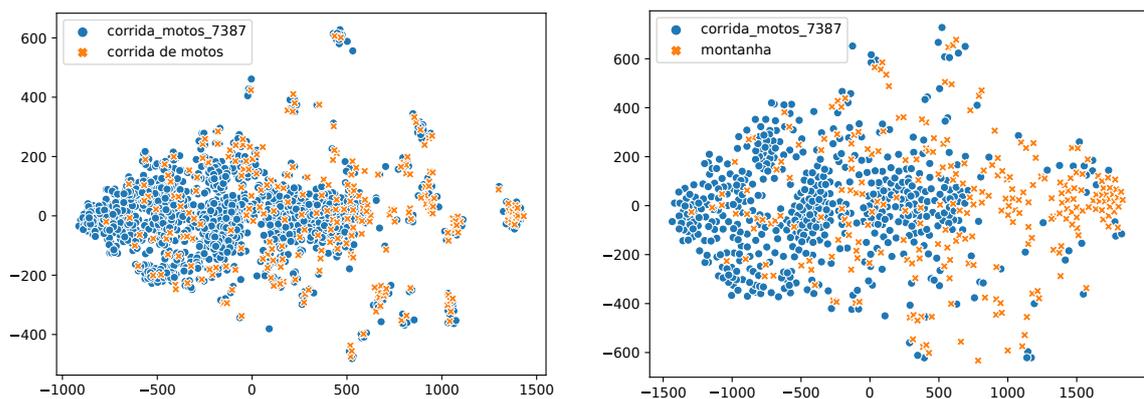
Caso de classificação incorreta com diferença máxima utilizando entropia



(a) Quantização convergente para (praia, 5x5, 128) com R-value = 0,155641 (b) Quantização divergente para (montanha, 15x15, 256) com R-value = 0,788945

Figura 5.38: Visualização bidimensional com t-SNE para o caso de classificação incorreta com diferença máxima (0,633303 para $k=3$ e $\theta=1$) entre a imagem de teste e suas classes verdadeira (a) e falsa (b) utilizando entropia de acordo com a Tabela B.18.

Caso de classificação incorreta com diferença mínima utilizando entropia



(a) Quantização convergente para (corrida de motos, 10x10, 256) com R-value = 0,568681 (b) Quantização divergente para (montanha, 15x15, 256) com R-value = 0,570352

Figura 5.39: Visualização bidimensional com t-SNE para o caso de classificação incorreta com diferença mínima (0,001670 para $k=3$ e $\theta=1$) entre a imagem de teste e suas classes verdadeira (a) e falsa (b) utilizando entropia de acordo com a Tabela B.18.

Conclusões

Os trabalhos anteriores de reconhecimento de padrões, utilizando compressão de dados, contribuíram significativamente para o desenvolvimento de soluções independentes das etapas de extração e seleção de características. Entretanto, são reportadas dificuldades em tratar problemas envolvendo dados bidimensionais. O presente trabalho propôs um método de classificação de cenas. O método é caracterizado pela intuitividade do funcionamento e do critério de decisão. Sua arquitetura é composta de etapas bem definidas, que permitem escalabilidade e a proposição de extensões diretas e melhorias. Sua base é a compressão de dados, tanto para transformação das características discriminantes das classes, quanto como medida de complexidade para tomada de decisão. Além disso, contorna alguns problemas conhecidos das técnicas antecessoras como: o desprezo à semântica, o aprendizado *lazy learning* e a necessidade de um processo auxiliar de rotulação de clusters.

A principal hipótese da pesquisa investigou que *a complexidade intrínseca de uma cena, avaliada no nível de abstração adequado em que suas principais características emergem, pode ser utilizada para quantificação da informação e discriminação em problemas de classificação*. As principais contribuições do trabalho são:

- Proposição de um classificador de cenas que utiliza técnicas de compressão de dados para extração de semântica e avaliação da complexidade visual;
- Introdução do conceito de aleatoriedade como critério de decisão, antes desprezada pelos métodos clássicos de reconhecimento de padrões ou contornada por técnicas de seleção de características;
- Revisão bibliográfica do estado da arte em medidas de similaridade de imagens baseadas em compressão, com apontamentos de lacunas e oportunidades de pesquisas.

Retomando o *primeiro objetivo* definido na Seção 1.4, foi proposto um modelo de treinamento para eliminar detalhes e maximizar características utilizando a quantização vetorial. Essa técnica de compressão de dados com perda permite realizar a aproximação ou distanciamento entre as imagens de teste e os protótipos obtidos a partir do conjunto de

treinamento. Aproximações com dicionários convergentes, ou seja, utilizando protótipos da mesma classe da cena, produzem versões com menor capacidade de compressão, isto é, mais aleatórias em relação às aproximações com dicionários divergentes. Assim, as *primeira e segunda hipóteses* da Seção 1.3, de maximização das diferenças extraclasses e ampliação da complexidade intrínseca das cenas como ferramenta de discriminação, foram validadas.

A quantização divergente impõe que os componentes de uma cena sejam substituídos por *codevectors* de outra classe, localizados ou em uma porção específica ou em regiões marginais do espaço, onde as amostras dessa classe se concentram. Dessa forma, a ocorrência de símbolos repetidos na aproximação é maior, e as características próprias da cena são perdidas. Como consequência, a nova representação da cena é realizada com uma descrição mais simples, tanto sob o aspecto qualitativo quanto quantitativo. Do ponto de vista qualitativo, transpondo a complexidade de Kolmogorov que entende um objeto pela sua aleatoriedade, foi calculado o valor da entropia algorítmica dessa descrição, utilizando compressão sem perda. Do ponto de vista quantitativo, a entropia de Shannon também mostrou ser capaz de mensurar essa diferença. Assim, a *terceira e a quarta hipóteses* se confirmaram, atingindo o *segundo objetivo* proposto, de determinar uma medida para tomada de decisão. Apesar do método proposto não se preocupar com a taxa de compressão, seu desempenho é diretamente determinado pela capacidade de promover perdas no processo de transformação de características, para eliminar detalhes e ampliar a complexidade da cena.

Por fim, no *terceiro objetivo* foi avaliada a *quinta hipótese*, para determinar se o método é capaz de gerar modelos de treinamento eficientes a partir de poucas amostras. Embora o tamanho do conjunto de treinamento não tenha influenciado no resultado da prototipação das amostras, observou-se que a etapa de *Validação* foi diretamente afetada. Quanto menor a quantidade de amostras utilizadas para determinar os melhores parâmetros de granularidade da informação, mais combinações de possíveis cenários são aceitos na avaliação de qualidade. Poucas amostras não são suficientes para manter o rigor dos parâmetros com melhor desempenho. Os resultados mostram que as piores generalizações ocorrem para a partição C e as melhores para a partição A. Portanto, avalia-se que o método é computacionalmente mais eficiente quando treinado com poucas amostras; e obtém melhor acurácia quando é fornecido um conjunto de validação maior. Para isso, o conjunto de treinamento pode ser fracionado entre as etapas de prototipação e validação, sendo admitido que as amostras de prototipação seja um subconjunto das amostras de validação.

Em relação à medida de complexidade baseada em compressão, observou-se que taxa de convergência do treinamento $\varepsilon = 0,001$ é adequada para obtenção dos *codebooks*. Entretanto, essa recomendação não objetiva determinar o fechamento de questão. Os re-

sultados indicam variações mínimas de desempenho. Portanto, em condições experimentais com pouca disponibilidade de recursos computacionais, sugere-se essa configuração. De modo geral, ao comparar as taxas de generalização, podem ser encontrados resultados muito próximos dos obtidos na validação, mas também expressivamente negativos, com menos da metade do acerto obtido na etapa de *Validação*. O objetivo principal dessa etapa é sugerir parâmetros para cada classe, dentro do conjunto total de possibilidades dos hiperparâmetros. Avalia-se que um dos componentes mais promissores para aperfeiçoar o desempenho do presente método é a etapa de *Validação*. Se encontrados parâmetros que maximizam a complexidade na quantização convergente e minimizam na quantização divergente, os resultados podem ser aprimorados e a aplicação estendida.

Dentre os métodos de compressão utilizados para cálculo da complexidade, a implementação *LZW+Huffman* foi mais eficiente. Em termos de acurácia, o *bzip2* foi equivalente mas, em termos de invariância ao tamanho do conjunto de treinamento e a taxa de convergência, o *LZW+Huffman* mostrou-se superior. Em relação ao *ZIP/gzip*, mantendo o mesmo cenário e conjuntos de treinamento e teste, tiveram desempenho expressivamente inferior. A medida de entropia, além de ter implementação e processamento mais simples, tem melhores taxas de generalização e é mais adequada para cenários maiores. Entretanto, ao contrário do observado nos métodos de compressão, não fica clara a relação entre melhores parâmetros encontrados para cada classe e as medidas de distorção e dispersão das amostras de treinamento. No *dataset* utilizado, para cenários de até 8 classes a compressão é mais eficiente, entretanto, a partir de 9 classes a entropia é superior.

Os melhores parâmetros selecionados na etapa de *Validação* mostram que o método lida bem com alta dimensionalidade, optando por cenários com classes treinadas e classificadas em espaços de até 900 dimensões (30x30). Vale ressaltar que a arquitetura do classificador proposto favorece que diferentes classes do problema sejam avaliadas em diferentes níveis de abstração. Assim, a complexidade visual e a carga semântica de cada classe são tratadas individualmente. Portanto, é apresentado um classificador onde as cenas são processadas em seu próprio domínio de características. Não há um espaço unificado, mas subespaços que preservam o escopo descritivo de cada classe. Mesmo com essa característica, o método proposto é capaz de empregar medidas de decisão para reconhecer as cenas, compostas de alta variabilidade e heterogeneidade de características.

O desempenho em relação às transformações geométricas e aos processamentos de imagem mostrou que o método é invariante para reflexões verticais e horizontais. Nos processamentos de suavização, linear e não-linear, a medida de compressão foi mais eficiente que a entropia. Entretanto, ambas medidas de aleatoriedade são sensíveis à inserção de ruídos nos dados. Nas transformações de escala e rotação as medidas são equivalentes. No comparativo de acurácia, o método se mostrou mais eficiente que o *k-NN*, tanto por similaridade entre as imagens completas quanto em janelas, e equivalente

ao DAMICORE utilizando a medida G . A superioridade de desempenho do método proposto, em relação ao k-NN, mostra que a quantização é capaz de tornar as cenas mais discrimináveis.

Do ponto de vista dos aspectos psicológicos e neurofisiológicos está claro que o funcionamento do método é fundamentado na transformação de dados. O objetivo é capturar a complexidade da resposta de um estímulo visual, dado um modelo treinado. Ao preservar as características predominantes da classe, no processo de quantização convergente, a aproximação da imagem de teste e a classe verdadeira é menos dispendiosa. Para as demais classes, a aproximação gera mais distorção entre as representações.

O funcionamento, o escopo e as restrições de aplicações do método foram demonstrados por meio da discussão transversal entre Teoria Clássica dos Conjuntos, Autocorrelação Espacial Global, Teoria dos Conjuntos Aproximados e Teoria da Comunicação. Essas abordagens integradas fundamentam como o método funciona, baseado nas medidas de interseção e dispersão das amostras. Para demonstrar experimentalmente esses princípios foi utilizada a medida de *overlap R-value*. Os resultados corroboraram que a aleatoriedade induzida na quantização convergente está correlacionada com: 1) as relações de equivalência dos vizinhos mais próximos das amostras e 2) a distorção representacional do estímulo visual.

Dividindo as tarefas de classificação em 2 categorias, quanto à natureza das classes, pode-se encontrar problemas que: 1) exigem o reconhecimento de padrões entre classes muito próximas, por exemplo, raças de cachorros; e 2) problemas envolvendo classes totalmente diferentes, por exemplo, mobiliário doméstico. Na primeira categoria, os objetos compartilham dos mesmos atributos e o processo de decisão pode ser realizado em um espaço de características unificado. Na segunda categoria, a partir das características já estruturadas, a classificação é comumente feita baseando-se na presença/ausência de determinados atributos e não necessariamente nos seus valores (árvore de decisão). Tendo em vista que o problema é analisado em diferentes níveis de abstração (computação granular), torna o método capaz de ter diferentes percepções do que é característica em uma classe e do que não é em outra. Assim, é possível atuar simultaneamente de forma híbrida nessas 2 categorias de problemas. Portanto, utilizando a aproximação da complexidade de Kolmogorov, com compressão ou entropia, pode-se tratar problemas que não possuem o mesmo conjunto de atributos. A modelagem dos componentes da cena em janelas, permite a extração de descritores de baixo nível. Em seguida, a interação desses componentes formam descritores de alto nível implícitos.

6.1 Trabalhos futuros

As seguintes ações são propostas como trabalhos futuros:

- Novos experimentos de comparação entre o método proposto e o DAMI-CORE, implementando as medidas descritas na Seção 3.5 e a extensão proposta por [26] para problemas de classificação;
- Extensão do método investigando a aplicação dos descritores de Haralick [67] como medidas de aleatoriedade;
- Parametrização automática da extração de janelas e modelagem dos *codebooks* utilizando o detector de pontos de interesse SIFT [95];
- Utilização do método proposto em imagens coloridas e em escala de cinza, quantizadas em menos cores, justificada pelas argumentações de [32] e [130]. Segundo os resultados reportados pelos autores, mesmo com apenas 2 cores as imagens continuam preservando a informação visual relevante;
- Considerando que o método trata os componentes de uma imagem como padrões de textura, pretende-se aplicá-lo em problemas de análise de textura nas áreas da biologia, medicina, robótica e ciência forense como em [19];
- Avaliação da aplicabilidade do método em problemas multi-classe nos quais, dentro de uma mesma imagem seja possível segmentar regiões de interesse da classificação como, por exemplo, determinar áreas urbanas, industriais e rurais em imagens de satélite como em [24];
- Novos experimentos com o método proposto nos *datasets* utilizados em [25] e [64] (textura);
- Utilização de outros *space-fillings* para linearização das janelas como, por exemplo, *row-major*, *column-major*, *Hilbert*, *SCPO*, *snake-row*, *snake-column*, *spiral-in-to-out*, *espiral-out-to-in* e outras encontradas no apêndice de [139]. Utilizar os conceitos de *space-fillings* e os resultados reportados por [163];
- Substituição da medida de complexidade baseada em compressão pela técnica de *turmites* proposta em [168], considerando as limitações dos compressores com perda discutidas em [150]. Os autores criticam a utilização dessas técnicas de compressão sem perda para calcular as aproximações de K . Também afirmam que grande parte desse algoritmos são mais parecidos com estimadores de entropia do que de estimadores algorítmicos.

Referências Bibliográficas

- [1] ACHARYA, T.; RAY, A. K. **Image Processing: Principles and Applications**. John Wiley & Sons, 2005.
- [2] ADOBE DEVELOPERS ASSOCIATION. **TIFF Revision 6.0**. Technical report, Adobe Systems Incorporated, 1992.
- [3] AGGARWAL, C. C.; HINNEBURG, A.; KEIM, D. A. **On the surprising behavior of distance metrics in high dimensional space**. In: *International Conference on Database Theory*, p. 420–434, Berlin, Heidelberg, 2001. Springer.
- [4] AMATO, A.; DI LECCE, V. **A knowledge based approach for a fast image retrieval system**. *Image and Vision Computing*, 26(11):1466–1480, 2008.
- [5] AMIN, A.; QURESHI, H. A.; JUNAID, M.; HABIB, M. Y.; ANJUM, W. **Modified run length encoding scheme with introduction of bit stuffing for efficient data compression**. *2011 International Conference for Internet Technology and Secured Transactions, ICITST 2011*, p. 668–672, 2011.
- [6] ARAUJO, A.; CONCI, A.; MORAN, M.; MELO, R.; RESMINI, R. **On efficient computation of texture descriptors from sum and difference histograms considering the scales of patterns**. In: *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*, volume 2018-Novem, 2019.
- [7] ATTNEAVE, F. **Some informational aspects of visual perception**. *Psychological Review*, 61(3):183–193, 1954.
- [8] BAILÃO, A. S. D. O. **Reconhecimento de padrões por processos adaptativos de compressão**. PhD thesis, Universidade Federal de Goiás, 2020.
- [9] BARLOW, H. B. **Possible principles underlying the transformations of sensory messages**. *Sensory Communication*, 1:216–234, 1961.
- [10] BENEDETTO, D.; CAGLIOTI, E.; LORETO, V. **Language trees and zipping**. *Physical Review Letters*, 88(4):4, 2002.

- [11] BENNETT, C. H.; GÁCS, P.; LI, M.; VITÁNYI, P. M. B.; ZUREK, W. H. **Information distance**. *IEEE Transactions on Information Theory*, 44(4):1407–1423, 1998.
- [12] BERKHIN, P. **Survey of clustering data mining techniques**. In: Kogan, J.; Nicholas, C.; Teboulle, M., editors, *Grouping Multidimensional Data: Recent Advances in Clustering*, p. 25–71. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [13] BEYER, K.; GOLDSTEIN, J.; RAMAKRISHNAN, R.; SHAFT, U. **When is “nearest neighbor” meaningful?** In: *ICDT '99 Proceedings of the 7th International Conference on Database Theory*, p. 217–235, 1999.
- [14] BISHOP, C. M. **Pattern Recognition and Machine Learning**. Springer, New York, 2006.
- [15] BOUREAU, Y. L.; BACH, F.; LECUN, Y.; PONCE, J. **Learning mid-level features for recognition**. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, p. 2559–2566, 2010.
- [16] BOUTELDJA, S.; KOURGLI, A. **A comparative analysis of svm, k-nn, and decision trees for high resolution satellite image scene classification**. In: *Twelfth International Conference on Machine Vision (ICMV 2019)*, p. 1–7, Amsterdam, 2019.
- [17] CABALLERO, Y.; BELLO, R.; PIZANO, Y.; ALVAREZ, D.; GARCIA, M. M.; BALTÁ, A. **La teoría de los conjuntos aproximados en la edición de conjuntos de entrenamiento para mejorar el desempeño del método k-nn**. In: Fernández-Caballero, A.; Arjona, M. G. M.; González, E. A.; Tomé, S. M., editors, *Campus Multidisciplinar en Percepción e Inteligencia (CMPI 2006)*, p. 637–645, Albacete, 2006. Universidad de Castilla-La Mancha.
- [18] CALUDE, C. S.; DESFONTAINES, D. **Universality and almost decidability**. *Fundamenta Informaticae XXI*, p. 1001–1008, 2015.
- [19] CAMPANA, B. J. L.; KEOGH, E. J. **A compression-based distance measure for texture**. *Statistical Analysis and Data Mining*, 3:381–398, 2010.
- [20] CAMPANI, C. A. P.; MENEZES, P. B. **Teorias da aleatoriedade**. *Revista de Informática Teórica e Aplicada*, 11(2):75–98, 2004.
- [21] CAMPANI, C. A. P.; MENEZES, P. B. **Introdução à complexidade de kolmogorov**. In: Mattos, J. C. B.; Rosa Jr, L. S.; Pilla, M. L., editors, *Desafios e Avanços em Computação: O Estado da Arte*, p. 33–65. Editora Universitária/UFPel, 2009.

- [22] CARPENTIERI, B. **Data compression and clustering: a "blind" approach to classification.** *Proceedings of the 6th WSEAS European Computing Conference (ECC '12)*, p. 35–40, 2012.
- [23] CEBRIÁN, M.; ALFONSECA, M.; ORTEGA, A. **The normalized compression distance is resistant to noise.** *IEEE Transactions on Information Theory*, 53(5):1895–1900, 2007.
- [24] CERRA, D.; DATCU, M. **Image classification and indexing using data compression based techniques.** *International Geoscience and Remote Sensing Symposium (IGARSS)*, 1(1):237–240, 2008.
- [25] CERRA, D.; DATCU, M. **A fast compression-based similarity measure with applications to content-based image retrieval.** *Journal of Visual Communication and Image Representation*, 23(2):293–302, 2011.
- [26] CESAR, B. K. M. **Estudo e extensão da metodologia DAMICORE para tarefas de classificação.** PhD thesis, Universidade de São Paulo, 2016.
- [27] CHAITIN, G. J. **On the length of programs for computing finite binary sequences.** *Journal of the ACM (JACM)*, 13(4):547–69, 1966.
- [28] CHAITIN, G. J. **Information-theoretic computation complexity.** *IEEE Transactions on Information Theory*, 20(1):10–15, 1974.
- [29] CHEN, C.; REN, Y.; KUO, C.-C. J. **Indoor/outdoor classification with multiple experts.** *Big Visual Data Analysis*, p. 23–63, 2016.
- [30] CHENG, G.; HAN, J.; LU, X. **Remote sensing image scene classification: Benchmark and state of the art.** *Proceedings of the IEEE*, 105(10):1865–1883, 2017.
- [31] CHESTER, U. A.; RATSABY, J. **Universal distance measure for images.** In: *IEEE 27th Convention of Electrical and Electronics Engineers in Israel, IEEEI 2012*, p. 1–4, 2012.
- [32] CHUA, J. J.; TISCHER, P. E. **Focusing the normalised information distance on the relevant information content for image similarity.** In: *Digital Image Computing: Techniques and Applications, DICTA 2010*, p. 1–7, 2010.
- [33] CILIBRASI, R. **Statistical Inference Through Data Compression.** PhD thesis, Faculteit der Natuurwetenschappen, 2007.

- [34] CILIBRASI, R.; VITÁNYI, P. M. B. **Clustering by compression**. *IEEE Transactions on Information Theory*, 51(4):1523–1545, 2005.
- [35] CIOCCA, G.; CORCHS, S.; GASPARINI, F. **Complexity perception of texture images**. In: *International Conference on Image Analysis and Processing*, Lecture Notes in Computer Science, p. 119–126, Cham, 2015. Springer International Publishing.
- [36] CIOCCA, G.; CORCHS, S.; GASPARINI, F.; BRICOLO, E.; TEBANO, R. **Does color influence image complexity perception?** In: *International Workshop on Computational Color Imaging*, p. 139–148, Cham, 2015. Springer International Publishing.
- [37] COHEN, P. R.; JENSEN, D. **Overtting explained**. In: *Preliminary Papers of the Sixth International Workshop on Artificial Intelligence and Statistics*, p. 115–122, 1997.
- [38] COLTUC, D.; DATCU, M.; COLTUC, D. **On the use of normalized compression distances for image similarity detection**. *Entropy*, 20(2):99, 2018.
- [39] CORREIA, J. A. P. **System based on compression techniques for the recognition of handwritten digits**. PhD thesis, Universidade de Aveiro, 2017.
- [40] COVER, T. M.; THOMAS, J. A. **Elements of Information Theory**. Wiley-Interscience, second edition, 2006.
- [41] DAFNER, R.; COHEN-OR, D.; MATIAS, Y. **Context-based space filling curves**. *Computer Graphics Forum*, 19(3):209–218, 2000.
- [42] DAVIES, E. R. **Computer and Machine Vision: Theory, Algorithms, Practicalities**. Elsevier, 2012.
- [43] DE ARAÚJO, F. N. C. **Rotulação Automática de Clusters Baseados em Análise de Filogenias**. PhD thesis, Universidade Federal do Piauí - UFPI, 2018.
- [44] DE PAULA, N. G. **Detecção de região saliente em imagens usando dissimilaridade de cor e amostragem por pixels aleatórios**. PhD thesis, Universidade Tecnológica Federal do Paraná, 2015.
- [45] DEUSSEN, O.; SPICKER, M.; ZHENG, Q. **Weighted linde–buzo–gray stippling**. *ACM Transactions on Graphics (TOG)*, 36(6):1–12, 2017.
- [46] DOMINGOS, P. **A few useful things to know about machine learning**. *Communications of the ACM*, 55(10):78–87, 2012.
- [47] DONDERI, D. C. **Visual complexity: A review**. *Psychological Bulletin*, 132(1):73–97, 2006.

- [48] DOS SANTOS, J. M. M. **Análise de Sinal Fisiológico usando Métodos não Lineares**. PhD thesis, Faculdade de Ciências da Universidade do Porto, 2015.
- [49] DUAN, Z. **Characters recognition of binary image using KNN**. In: *ACM International Conference Proceeding Series*, p. 116–118. Association for Computing Machinery (ACM), 2018.
- [50] DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern Classification**. John Wiley & Sons, New York, second edition, 2001.
- [51] ESCARPINATI, M. C. **Investigação de Formatos e Compressão de Imagens Digitais para Processamento de Imagens Mamográficas de Mamas Densas**. PhD thesis, Universidade de São Paulo, 2002.
- [52] ESPINOZA, J. L. **Obtención de Conjuntos Aproximados Mediante Algoritmos Genéticos**. PhD thesis, Universidad de Costa Rica, 1999.
- [53] FALLON, J. J.; PICKEL, P. F.; MCERLAIN, S. J.; BUCK, J. **Content estimation data compression**, Oct. 17 2017. US Patent 9,792,308.
- [54] FERAIDOONI, M. M.; GHARAVIAN, D. **A new approach for rotation-invariant and noise-resistant texture analysis and classification**. *Machine Vision and Applications*, 29(3):455–466, 2018.
- [55] GABRIEL, C.; WITTMANN, C.; HACKER, B.; MAUERER, W.; HUNTINGTON, E.; SABUNCU, M.; MARQUARDT, C.; LEUCHS, G. **A high-speed secure quantum random number generator based on vacuum states**. In: *Conference on Lasers and Electro-Optics 2012*, p. 2, San Jose, CA, 2012.
- [56] GABRIEL, C.; WITTMANN, C.; SYCH, D.; DONG, R.; MAUERER, W.; ANDERSEN, U. L.; MARQUARDT, C.; LEUCHS, G. **A generator for unique quantum random numbers based on vacuum states**. *Nature Photonics*, 4(10):711–715, 2010.
- [57] GAMMERMAN, A.; VOVK, V. **Kolmogorov complexity: Sources, theory and applications**. *The Computer Journal*, 42(4):252–255, 1999.
- [58] GARCIA, E. K.; FELDMAN, S.; GUPTA, M. R.; SRIVASTAVA, S. **Completely lazy learning**. *IEEE Transactions on Knowledge and Data Engineering*, 22(9):1274–1285, 2010.
- [59] GAUVRIT, N.; SOLER-TOSCANO, F.; ZENIL, H. **Natural scene statistics mediate the perception of image complexity**. *Visual Cognition*, 22(8):1084–1091, 2014.

- [60] GERLACH, C.; MARQUES, J. F. **Visual complexity exerts opposing effects on object categorization and identification.** *Visual Cognition*, 22(6):751–769, 2014.
- [61] GINESU, G.; PINTUS, M.; GIUSTO, D. D. **Objective assessment of the webp image coding algorithm.** *Signal Processing: Image Communication*, 27:867–874, 2012.
- [62] GOLDMEIER, E. **Similarity in visually perceived forms.** *Psychological Issues*, 8(1):135, 1972.
- [63] GONZALEZ, R. C.; WOODS, R. **Processamento digital de imagens.** São Paulo: Pearson Prentice Hall, 2010.
- [64] GUHA, T.; WARD, R. K. **Image similarity using sparse representation and compression distance.** *IEEE Transactions on Multimedia*, 16(4):980–987, 2014.
- [65] HAHN, U.; CHATER, N.; RICHARDSON, L. B. **Similarity as transformation.** *Cognition*, 87:1–32, 2003.
- [66] HANSEN, T.; GEGENFURTNER, K. R. **Independence of color and luminance edges in natural scenes.** *Visual Neuroscience*, 26(1):35–49, 2009.
- [67] HARALICK, R.; SHANMUGAN, K.; DINSTEN, I. **Textural features for image classification.** *IEEE Transactions on Systems, Man and Cybernetics*, 3:610–621, 1973.
- [68] HAYKIN, S. **Sistemas de Comunicação Analógicos e Digitais.** Bookman, 4^a edition, 2004.
- [69] HECKBERT, P. **Color image quantization for frame buffer display.** In: *9th Annual Conference on Computer Graphics and Interactive Techniques*, p. 297–307, Boston, Massachusetts, 1982.
- [70] HEIDEMANN, G.; RITTER, H. **On the contribution of compression to visual pattern recognition.** *International Conference on Computer Vision Theory and Applications (VISAPP 2008)*, p. 83–90, 2008.
- [71] HEIDEMANN, G.; RITTER, H. **Data compression - a generic principle of pattern recognition?** *Vision and Computer Graphics. Theory and*, p. 202–212, 2009.
- [72] ITU RADIOCOMMUNICATION SECTOR. **Recommendation itu-r bt.709-6: Parameter values for the hdtv standards for production and international programme exchange.** Technical report, International Telecommunication Union, 2015.
- [73] JAIN, A. K.; DUIN, R. P. W.; MAO, J. **Statistical pattern recognition: A review.** *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

- [74] JAIN, A. K.; DUIN, R. P.; MAO, J. **Statistical pattern recognition: A review.** *IEEE Transactions on pattern analysis and machine intelligence*, 22(1):4–37, 2000.
- [75] JAIN, N.; SALANKAR, S. S. **Content based image retrieval using improved gabor wavelet transform and linear discriminant analysis.** In: *3rd International Conference for Convergence in Technology, I2CT 2018*, p. 1–4. IEEE, 2018.
- [76] KAMILARIS, A.; PRENAFETA-BOLDÚ, F. X. **Deep learning in agriculture: A survey.** *Computers and Electronics in Agriculture*, 147(July 2017):70–90, 2018.
- [77] KANDEL, E. R.; SCHWARTZ, J. H.; JESSELL, T. M.; SIEGELBAUM, S. A.; HUDSPETH, A. J. **Princípios de Neurociências.** AMGH Editora, Porto Alegre, quinta edição, 2014.
- [78] KANUNGO, T.; MOUNT, D. M.; NETANYAHU, N. S.; PIATKO, C. D.; SILVERMAN, R.; WU, A. Y. **An efficient k-means clustering algorithm: Analysis and implementation.** *IEEE Transactions Pattern Analysis and Machine Intelligence*, 24(7):881–892, 2002.
- [79] KOHONEN, T. **Self-organizing maps.** In: Kohonen, T., editor, *Springer Series in Information Sciences*. Springer, 1995.
- [80] KOLMOGOROV, A. N. **Three approaches to the quantitative definiion of information**, 1965.
- [81] KRCÁL, L. **Incremental Clustering-Based Compression.** PhD thesis, Czech Technical University in Prague, 2014.
- [82] KUMAR, S. N.; LENIN FRED, A.; SEBASTIN VARGHESE, P. **Compression of ct images using contextual vector quantization with simulated annealing for telemedicine application.** *Journal of Medical Systems*, 42(11), 2018.
- [83] LAI, J. Z. **Fast encoding algorithms for tree-structured vector quantization.** *Image and Vision Computing*, 15(11):867–871, 1997.
- [84] LECUN, Y.; BENGIO, Y.; HINTON, G. **Deep learning.** *Nature*, 521(7553):436–444, 2015.
- [85] L'ECUYER, P.; MUNGER, D.; ORESHKIN, B.; SIMARD, R. **Random numbers for parallel computers: Requirements and methods, with emphasis on gpus.** *Mathematics and Computers in Simulation*, 135:3–17, 2017.
- [86] LEEUWENBERG, E. L. J. **A perceptual coding language for visual and auditory patterns.** *The American Journal of Psychology*, 84(3):307–349, 1971.

- [87] LEGAZ-APARICIO, Á. G.; VERDÚ-MONEDERO, R.; ENGAN, K. **Noise robust and rotation invariant framework for texture analysis and classification.** *Applied Mathematics and Computation*, 335:124–132, 2018.
- [88] LI, M.; CHEN, X.; LI, X.; MA, B.; VITÁNYI, P. M. B. **The similarity metric.** *IEEE Transactions on Information Theory*, 50(12):3250–3264, 2004.
- [89] LI, M.; VITÁNYI, P. M. B. **An Introduction to Kolmogorov Complexity and Its Applications.** Springer, second edition, 1993.
- [90] LINDE, Y.; BUZO, A.; GRAY, R. M. **An algorithm for vector quantizer design.** *IEEE Transactions on Communications*, 28(1):84–95, 1980.
- [91] LINDEN, R. **Técnicas de Agrupamento.** *Revista de Sistemas de Informação da FSMA*, 4:18–36, 2009.
- [92] LIU, G.-H.; YANG, J.-Y.; LI, Z. **Content-based image retrieval using computational visual attention model.** *Pattern Recognition*, 48(8):2554–2566, 2015.
- [93] LONGLEY, P. A.; GOODCHILD, M. F.; MAGUIRE, D. J.; RHIND, D. W. **Sistemas e ciência da informação geográfica.** Bookman Editora, 2009.
- [94] LOPES, L. A.; MACHADO, V. P.; RABÊLO, R. A.; FERNANDES, R. A.; LIMA, B. V. **Automatic labelling of clusters of discrete and continuous data with supervised machine learning.** *Knowledge-Based Systems*, 106:231–241, 2016.
- [95] LOWE, D. G. **Object recognition from local scale-invariant features.** In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, p. 1150–1157. IEEE, 1999.
- [96] LU, Y.; CHEN, W.; POOR, H. V. **User preference aware lossless data compression at the edge.** *IEEE Transactions on Communications*, p. 1–16, 2020.
- [97] LYU, S. **Divisive normalization: Justification and effectiveness as efficient coding transform.** *Advances in Neural Information Processing Systems*, p. 1522–1530, 2010.
- [98] MAANI, R.; KALRA, S.; YANG, Y. H. **Noise robust rotation invariant features for texture classification.** *Pattern Recognition*, 46(8):2103–2116, 2013.
- [99] MACHADO, P.; ROMERO, J.; NADAL, M.; SANTOS, A.; CORREIA, J.; CARBALLAL, A. **Computerized measures of visual complexity.** *Acta Psychologica*, 160:43–57, 2015.

- [100] MANDHALA, V. N.; SUJATHA, V.; DEVI, B. R. **Scene classification using support vector machines.** In: *Proceedings of 2014 IEEE International Conference on Advanced Communication, Control and Computing Technologies, ICACCCT 2014*, p. 1807–1810. IEEE, 2014.
- [101] MARGENSTERN, M. **Turing machines with two letters and two states.** *Complex Systems*, 19(1):29–43, 2010.
- [102] MARQUARDT, C. **Random numbers game with quantum dice**, 2010.
- [103] MARTINET, J.; ELSAYAD, I. **Mid-level image descriptors.** *Intelligent Multimedia Databases and Information Retrieval: Advancing Applications and Technologies*, p. 46–60, 2011.
- [104] MATHWORKS. **Reduce the Number of Colors in an Image**, 2018.
- [105] MATOS, A.; MOREIRA, N. **Introdução à complexidade de kolmogorov.** Technical report, Universidade do Porto, Porto, Portugal, 2007.
- [106] MAYO, M.; WATSON, A. T. **Automatic species identification of live moths.** *Knowledge-Based Systems*, 20(2):195–202, 2007.
- [107] MEISEIL, W. S. **Computer-Oriented Approaches to Pattern Recognition.** Academic Press, 1972.
- [108] MIANO, J. **Compressed Image File Formats.** Addison-Wesley Professional, 1999.
- [109] MOL, L. D.; BULLYNCK, M.; DAYLIGHT, E. G. **Less is more in the fifties: Encounters between logical minimalism and computer design during the 1950s.** *IEEE Annals of the History of Computing*, 40(1):19–45, 2018.
- [110] MOORE, T.; ZIRNSAK, M. **Neural mechanisms of selective visual attention.** *Annual Review of Psychology*, 68:47–72, 2017.
- [111] MORAN, P. A. P. **Notes on continuous stochastic phenomena.** *Biometrika*, 37(2):17–23, 1950.
- [112] MORTENSEN, J.; WU, J. J.; FURST, J.; ROGERS, J.; RAICU, D. **Effect of image linearization on normalized compression distance.** *Signal Processing, Image Processing and Pattern Recognition*, 61:106–116, 2009.
- [113] MURPHY, K. P. **Machine Learning: A Probabilistic Perspective.** MIT Press, London, England, 2012.

- [114] NAIK, A. K.; U, D. A. **Scalable image classification using compression**. *Journal of Computer Engineering (IOSR-JCE)*, 18(4):1–5, 2016.
- [115] NELSON, M.; GAILLY, J.-L. **The Data Compression Book**. M&T Books, second edition, 1995.
- [116] NIKVAND, N. **Image Information Distance Analysis and Applications**. PhD thesis, University of Waterloo, 2014.
- [117] NIKVAND, N.; WANG, Z. **Generic image similarity based on kolmogorov complexity**. In: *2010 IEEE International Conference on Image Processing*, p. 309–312, Hong Kong, 2010.
- [118] OEHLER, K. L.; GRAY, R. **Combining image classification and image compression using vector quantization**. In: *Data Compression Conference (DCC '93)*, p. 2–11, Snowbird, UT, USA, 1993.
- [119] OH, S. **A new dataset evaluation method based on category overlap**. *Computers in Biology and Medicine*, 41(2):115–122, 2011.
- [120] OLIVA, A.; MACK, M. L.; SHRESTHA, M.; PEEPER, A. **Identifying the perceptual dimensions of visual complexity of scenes**. In: *Proc. of the 26th Annual Meeting of the Cognitive Science Society*, p. 1041–1046, 2004.
- [121] OLSHAUSEN, B. A.; FIELD, D. J. **Emergence of simple-cell receptive field properties by learning a sparse code for natural images**. *Nature*, 381:607–609, 1996.
- [122] PALUMBO, L.; MAKIN, A. D. J.; BERTAMINI, M. **Examining visual complexity and its influence on perceived duration**. *Journal of Vision*, 14(14):1–18, 2017.
- [123] PARTININGSIH, N. D. A.; FRATAMA, R. R.; SARI, C. A.; SETIADI, D. R. I. M.; RACHMAWANTO, E. H. **Handwriting ownership recognition using contrast enhancement and lbp feature extraction based on knn**. In: *5th International Conference on Information Technology, Computer and Electrical Engineering, ICITACEE 2018*, p. 342–346. IEEE, 2018.
- [124] PASTOR, F. S. **Mecanismos de Optimización en el Preprocesado para H.265/HEVC**. PhD thesis, Universidad Complutense de Madrid, 2016.
- [125] PATEL, R.; PATEL, B. P. **An analytical study on comparison of different image compression formats**. *International Journal for Innovative Research in Science & Technology*, 1(7):24–31, 2014.

- [126] PAWLAK, Z. **Rough sets**. *International Journal of Computer & Information Sciences*, 11(5):341–356, 1982.
- [127] PAWLAK, Z.; GRZYMALA-BUSSE, J.; SLOWINSKI, R.; ZIARKO, W. **Rough Sets**. *Communications of The ACM*, 38(11):89–95, 1995.
- [128] PAYNE, A.; SINGH, S. **Indoor vs. outdoor scene classification in digital photographs**. *Pattern Recognition*, 38(10):1533–1545, 2005.
- [129] PEDRINI, H.; SCHWARTZ, W. R. **Análise de imagens digitais: princípios, algoritmos e aplicações**. Thomson Learning, 2008.
- [130] PINHO, A. J.; FERREIRA, P. J. S. G. **Image similarity using the normalized compression distance based on finite context models**. In: *18th IEEE International Conference on Image Processing (ICIP)*, p. 1993–1996, 2011.
- [131] PINHO, A. J.; PRATAS, D.; FERREIRA, P. J. S. G. **A new compressor for measuring distances among images**. *International Conference Image Analysis and Recognition*, 8814(1):30–37, 2014.
- [132] PINHO, A. J.; PRATAS, D.; FERREIRA, P. J. **Authorship attribution using relative compression**. In: *Data Compression Conference Proceedings*, p. 329–338, 2016.
- [133] RANGEL, J. C.; CAZORLA, M.; GARCÍA-VAREA, I.; MARTÍNEZ-GÓMEZ, J.; FROMONT, É.; SEBBAN, M. **Scene classification based on semantic labeling**. *Advanced Robotics*, 30(11-12):758–769, 2016.
- [134] RAO, M. S.; REDDY, B. **Comparative analysis of pattern recognition methods: An overview**. *Indian Journal of Computer Science and Engineering (IJCSE)*, 2(3):385–390, 2011.
- [135] REDDY EDLA, D.; GONDLEKAR, V.; GAUNS, V. **Hk-means: A heuristic approach to initialize and estimate the number of clusters in biological data**. *Acta Physica Polonica A*, 130(1):78–82, 2016.
- [136] ROKACH, L. **Decision forest: Twenty years of research**. *Information Fusion*, 27:111–125, 2016.
- [137] ROSSINI, J. C.; GALERA, C. **Atenção visual: estudos comportamentais da seleção baseada no espaço e no objeto**. *Estudos de Psicologia (Natal)*, 11(1):79–86, 2006.
- [138] RUSSELL, K. N.; DO, M. T.; HUFF, J. C.; PLATNICK, N. I. **Introducing spida-web: Wavelets, neural networks and internet accessibility in an image-based**

- automated identification system.** *Automated Taxon Identification in Systematics: Theory, Approaches and Applications*, 74:131–152, 2007.
- [139] SALOMON, D. **Data Compression: The Complete Reference.** Springer, third edition, 2004.
- [140] SANCHES, A.; CARDOSO, J. M.; DELBEM, A. C. B. **Identifying merge-beneficial software kernels for hardware implementation.** In: *2011 International Conference on Reconfigurable Computing and FPGAs, ReConFig 2011*, p. 74–79, 2011.
- [141] SANTOS, P. L. V. A. D. C.; SANTANA, R. C. G. **Dado e Granularidade na perspectiva da Informação e Tecnologia: uma interpretação pela Ciência da Informação.** *Ciência da Informação*, 42(2):199–209, 2015.
- [142] SAYOOD, K. **Introduction to Data Compression.** Morgan Kaufmann, San Francisco, third edition, 2006.
- [143] SHAH, A. A.; RANA, K. **A Review on Computer Vision - Scene Classification Techniques.** In: *2019 Third International Conference on Inventive Systems and Control (ICISC)*, p. 558–566. IEEE, 2019.
- [144] SHANNON, C. E. **A mathematical theory of communication.** *The Bell System Technical Journal*, 27(April 1928):379–423, 1948.
- [145] SHEIKH, H. R.; BOVIK, A. C. **Image information and visual quality.** *IEEE Transactions on Image Processing*, 15(2):430–444, 2006.
- [146] SHEPARD, R. N. **Stimulus and response generalization: A stochastic model relating generalization to distance in psychological space.** *Psychometrika*, 22(4):325–345, 1957.
- [147] SHI, N.; LIU, X.; GUAN, Y. **Research on k-means clustering algorithm: An improved k-means clustering algorithm.** *3rd International Symposium on Intelligent Information Technology and Security Informatics, IITSI 2010*, p. 63–67, 2010.
- [148] SILVA, E. A.; PANETTA, K.; AGAIAN, S. S. **Quantifying image similarity using measure of enhancement by entropy.** In: *Defense and Security Symposium*, p. 1–12, Orlando, 2007.
- [149] SMEULDERS, A. W. M.; WORRING, M.; SANTINI, S.; GUPTA, A.; JAIN, R. **Content-based image retrieval at the end of the early years.** *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

- [150] SOLER-TOSCANO, F.; ZENIL, H. **A computable measure of algorithmic probability by finite approximations with an application to integer sequences.** *Complexity*, 2017:1–10, 2017.
- [151] SOLOMONOFF, R. J. **A preliminary report on a general theory of inductive inference.** *Revision of Report*, 131, 1960.
- [152] SUJEE, R.; SESH, V. B. **Natural scene classification.** In: *2019 International Conference on Computer Communication and Informatics, ICCCI 2019*, p. 1–7, Coimbatore, 2019. IEEE.
- [153] SUTTON, R. S.; BARTO, A. G. **Reinforcement Learning: An Introduction.** MIT Press, London, second edition, 2017.
- [154] THEODORIDIS, S.; KOUTROUMBAS, K. **Nonlinear classifiers.** *Pattern Recognition, Academic Press*, p. 151–260, 2009.
- [155] TOBLER, W. R. **A computer movie simulation urban growth in detroit region.** *Economic Geography*, 46:234–240, 1970.
- [156] TONG, Z.; SHI, D.; YAN, B.; WEI, J. **A review of indoor-outdoor scene classification.** *Advances in Intelligent Systems Research*, 134:469–474, 2017.
- [157] TRAN, N. **The normalized compression distance and image distinguishability.** *Human Vision and Electronic Imaging XII*, 6492(February 2007):11, 2007.
- [158] TRAN, N. **A perceptual similarity measure based on smoothing filters and the normalized compression distance.** *Human Vision and Electronic Imaging XV*, 7527(February 2010):1–13, 2010.
- [159] TVERSKY, A. **Features of similarity.** *Psychological Review*, 84(4):327–352, 1977.
- [160] UMBAUGH, S. E. **Digital Image Processing and Analysis with MATLAB and CVIPtools: Applications with MATLAB® and CVIPtools.** CRC Press, 2017.
- [161] VAN DER MAATEN, L.; HINTON, G. **Visualizing data using t-sne laurens.** *Journal of Machine Learning Research*, 9(Nov.):2579–2605, 2008.
- [162] VÁZQUEZ, P. P.; MARCO, J. **Using normalized compression distance for image similarity measurement: An experimental study.** *Visual Computer*, 28(11):1063–1084, 2012.
- [163] VEMURI, B. C.; SAHNI, S.; CHEN, F.; KAPOOR, C.; LEONARD, C.; FITZSIMMONS, J. **Lossless image compression**, 2002.

- [164] WATANABE, T.; SUGAWARA, K.; SUGIHARA, H. **A new pattern representation scheme using data compression.** *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):579–590, 2002.
- [165] XIE, L.; LEE, F.; LIU, L.; KOTANI, K.; CHEN, Q. **Scene recognition: A comprehensive survey.** *Pattern Recognition*, 102, 2020.
- [166] YANG, H.; HUANG, C.; WANG, F.; SONG, K.; ZHENG, S.; YIN, Z. **Large-scale and rotation-invariant template matching using adaptive radial ring code histograms.** *Pattern Recognition*, 91:345–356, 2019.
- [167] YU, D.; LIU, P.; YE, Z.; TANG, X.; ZHAO, W. **Hierarchy-associated semantic-rule inference framework for classifying indoor scenes.** *Journal of Electronic Imaging*, 25(2):1–16, 2016.
- [168] ZENIL, H.; SOLER-TOSCANO, F.; DELAHAYE, J.-P.; GAUVRIT, N. **Two-dimensional kolmogorov complexity and validation of the coding theorem method by compressibility.** *PeerJ Computer Science*, p. 1–31, 2015.

Cenários dos Experimentos

ID	Cenário
1	[avião, 30x30, 128], [campo, 5x5, 512], [fogos de artifício, 10x10, 512], [flor, 15x15, 512], [moto, 15x15, 512], [pôr do sol, 30x30, 256], [árvore, 10x10, 512]
2	[avião, 30x30, 128], [campo, 5x5, 512], [fogos de artifício, 10x10, 512], [flor, 15x15, 512], [corrida de motos, 10x10, 128], [moto, 15x15, 512], [pôr do sol, 30x30, 256], [árvore, 10x10, 512]
3	[avião, 30x30, 128], [praia, 10x10, 128], [campo, 5x5, 512], [fogos de artifício, 10x10, 512], [flor, 15x15, 512], [corrida de motos, 10x10, 128], [moto, 15x15, 512], [pôr do sol, 30x30, 256], [árvore, 10x10, 512]
4	[avião, 5x5, 256], [campo, 5x5, 256], [fogos de artifício, 10x10, 512], [flor, 15x15, 512], [corrida de motos, 10x10, 512], [moto, 10x10, 512], [árvore, 5x5, 512]
5	[avião, 5x5, 256], [campo, 5x5, 256], [fogos de artifício, 10x10, 512], [flor, 15x15, 512], [corrida de motos, 10x10, 512], [moto, 10x10, 512], [pôr do sol, 5x5, 256], [árvore, 5x5, 512]
6	[avião, 5x5, 256], [construção, 5x5, 512], [campo, 5x5, 256], [fogos de artifício, 10x10, 512], [flor, 15x15, 512], [corrida de motos, 10x10, 512], [moto, 10x10, 512], [pôr do sol, 5x5, 256], [árvore, 5x5, 512]
7	[avião, 20x20, 512], [praia, 10x10, 256], [construção, 5x5, 512], [campo, 5x5, 256], [fogos de artifício, 10x10, 512], [flor, 15x15, 512], [corrida de motos, 10x10, 512], [moto, 10x10, 512], [pôr do sol, 5x5, 256], [árvore, 5x5, 512]
8	[avião, 20x20, 512], [praia, 10x10, 256], [construção, 5x5, 512], [carro, 20x20, 512], [campo, 5x5, 256], [fogos de artifício, 10x10, 512], [flor, 15x15, 512], [corrida de motos, 10x10, 512], [moto, 10x10, 512], [pôr do sol, 5x5, 256], [árvore, 5x5, 512]
9	[avião, 20x20, 256], [campo, 10x10, 512], [fogos de artifício, 10x10, 512], [moto, 20x20, 256], [pôr do sol, 15x15, 512], [árvore, 10x10, 512]
10	[avião, 25x25, 256], [construção, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 20x20, 512], [moto, 20x20, 512], [pôr do sol, 30x30, 256]
11	[avião, 25x25, 256], [construção, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 20x20, 512], [moto, 20x20, 512], [montanha, 15x15, 512], [pôr do sol, 30x30, 256]
12	[avião, 25x25, 256], [praia, 5x5, 512], [construção, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 20x20, 512], [moto, 20x20, 512], [montanha, 15x15, 512], [pôr do sol, 30x30, 256]
13	[avião, 30x30, 256], [construção, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 25x25, 512], [corrida de motos, 20x20, 512], [moto, 25x25, 512], [montanha, 15x15, 512], [pôr do sol, 25x25, 512], [árvore, 15x15, 512]
14	[avião, 30x30, 256], [praia, 5x5, 512], [construção, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 25x25, 512], [corrida de motos, 20x20, 512], [moto, 25x25, 512], [montanha, 15x15, 512], [pôr do sol, 25x25, 512], [árvore, 15x15, 512]
15	[avião, 30x30, 512], [construção, 10x10, 512], [fogos de artifício, 30x30, 256], [flor, 20x20, 512], [moto, 25x25, 64], [pôr do sol, 30x30, 256], [árvore, 5x5, 512]
16	[avião, 25x25, 512], [construção, 10x10, 512], [fogos de artifício, 25x25, 512], [flor, 20x20, 512], [corrida de motos, 10x10, 512], [moto, 25x25, 64], [pôr do sol, 25x25, 512], [árvore, 5x5, 512]
17	[avião, 25x25, 512], [construção, 10x10, 512], [campo, 25x25, 512], [fogos de artifício, 25x25, 512], [flor, 20x20, 512], [corrida de motos, 10x10, 512], [moto, 25x25, 64], [pôr do sol, 25x25, 512], [árvore, 5x5, 512]
18	[avião, 20x20, 256], [campo, 10x10, 512], [fogos de artifício, 10x10, 512], [moto, 20x20, 256], [pôr do sol, 15x15, 512], [árvore, 10x10, 512]
19	[avião, 25x25, 256], [construção, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 20x20, 512], [moto, 20x20, 512], [pôr do sol, 30x30, 256]

Continua na próxima página ...

Tabela A.1 Cenários dos experimentos (continuação).

ID	Cenário
20	[avião, 25x25, 256], [construção, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 20x20, 512], [moto, 20x20, 512], [montanha, 15x15, 512], [pôr do sol, 30x30, 256]
21	[avião, 25x25, 256], [praia, 5x5, 512], [construção, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 20x20, 512], [moto, 20x20, 512], [montanha, 15x15, 512], [pôr do sol, 30x30, 256]
22	[avião, 30x30, 256], [construção, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 25x25, 512], [corrida de motos, 20x20, 512], [moto, 25x25, 512], [montanha, 15x15, 512], [pôr do sol, 25x25, 512], [árvore, 15x15, 512]
23	[avião, 30x30, 256], [praia, 5x5, 512], [construção, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 25x25, 512], [corrida de motos, 20x20, 512], [moto, 25x25, 512], [montanha, 15x15, 512], [pôr do sol, 25x25, 512], [árvore, 15x15, 512]
24	[praia, 5x5, 512], [construção, 10x10, 512], [fogos de artifício, 5x5, 512], [flor, 10x10, 512], [corrida de motos, 10x10, 512], [moto, 25x25, 128], [árvore, 5x5, 512]
25	[praia, 5x5, 512], [construção, 10x10, 512], [fogos de artifício, 5x5, 512], [flor, 10x10, 512], [corrida de motos, 10x10, 512], [moto, 25x25, 128], [pôr do sol, 30x30, 128], [árvore, 5x5, 512]
26	[avião, 20x20, 256], [praia, 5x5, 512], [construção, 10x10, 512], [fogos de artifício, 5x5, 512], [flor, 10x10, 512], [moto, 20x20, 256], [montanha, 10x10, 512], [pôr do sol, 25x25, 256], [árvore, 5x5, 512]
27	[avião, 20x20, 256], [praia, 5x5, 512], [construção, 10x10, 512], [fogos de artifício, 5x5, 512], [flor, 10x10, 512], [corrida de motos, 10x10, 512], [moto, 20x20, 256], [montanha, 10x10, 512], [pôr do sol, 25x25, 256], [árvore, 5x5, 512]
28	[avião, 20x20, 256], [praia, 5x5, 512], [construção, 10x10, 512], [campo, 5x5, 512], [fogos de artifício, 5x5, 512], [flor, 10x10, 512], [corrida de motos, 10x10, 512], [moto, 20x20, 256], [montanha, 10x10, 512], [pôr do sol, 30x30, 128], [árvore, 5x5, 512]
29	[praia, 10x10, 256], [fogos de artifício, 25x25, 512], [flor, 20x20, 512], [corrida de motos, 10x10, 512], [moto, 5x5, 512], [pôr do sol, 10x10, 256], [árvore, 5x5, 512]
30	[avião, 5x5, 256], [campo, 5x5, 256], [fogos de artifício, 25x25, 512], [flor, 20x20, 512], [corrida de motos, 10x10, 512], [moto, 5x5, 512], [pôr do sol, 10x10, 256], [árvore, 5x5, 512]
31	[avião, 5x5, 256], [construção, 5x5, 512], [campo, 5x5, 256], [fogos de artifício, 25x25, 512], [flor, 20x20, 512], [corrida de motos, 10x10, 512], [moto, 5x5, 512], [pôr do sol, 10x10, 256], [árvore, 5x5, 512]
32	[avião, 5x5, 256], [praia, 5x5, 256], [construção, 5x5, 512], [campo, 5x5, 256], [fogos de artifício, 25x25, 512], [flor, 20x20, 512], [corrida de motos, 10x10, 512], [moto, 5x5, 512], [pôr do sol, 10x10, 256], [árvore, 5x5, 512]
33	[avião, 30x30, 512], [praia, 5x5, 128], [construção, 5x5, 256], [carro, 20x20, 512], [campo, 5x5, 128], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [corrida de motos, 10x10, 256], [moto, 15x15, 512], [pôr do sol, 5x5, 128], [árvore, 5x5, 256]
34	[avião, 25x25, 256], [construção, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 15x15, 512], [moto, 20x20, 512]
35	[avião, 25x25, 256], [construção, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 15x15, 512], [moto, 20x20, 512], [pôr do sol, 20x20, 512]
36	[avião, 20x20, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 20x20, 512], [corrida de motos, 20x20, 512], [moto, 25x25, 512], [pôr do sol, 25x25, 512], [árvore, 15x15, 512]
37	[avião, 20x20, 512], [praia, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 20x20, 512], [corrida de motos, 20x20, 512], [moto, 25x25, 512], [pôr do sol, 30x30, 256], [árvore, 15x15, 512]
38	[avião, 25x25, 128], [praia, 15x15, 256], [construção, 10x10, 512], [campo, 5x5, 256], [fogos de artifício, 5x5, 512], [flor, 15x15, 256], [corrida de motos, 15x15, 256], [moto, 25x25, 128], [montanha, 15x15, 256], [pôr do sol, 20x20, 256]
39	[avião, 25x25, 128], [praia, 15x15, 256], [construção, 10x10, 512], [campo, 5x5, 256], [fogos de artifício, 5x5, 512], [flor, 15x15, 256], [corrida de motos, 15x15, 256], [moto, 25x25, 128], [montanha, 15x15, 256], [pôr do sol, 20x20, 256], [árvore, 5x5, 512]
40	[avião, 25x25, 128], [praia, 15x15, 256], [construção, 10x10, 512], [carro, 10x10, 512], [campo, 5x5, 256], [fogos de artifício, 5x5, 512], [flor, 15x15, 256], [corrida de motos, 15x15, 256], [moto, 25x25, 128], [montanha, 15x15, 256], [pôr do sol, 20x20, 256], [árvore, 5x5, 512]
41	[avião, 30x30, 256], [construção, 10x10, 512], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [moto, 25x25, 64], [árvore, 5x5, 512]
42	[avião, 30x30, 256], [campo, 25x25, 512], [fogos de artifício, 20x20, 512], [flor, 20x20, 512], [corrida de motos, 15x15, 64], [moto, 25x25, 64], [pôr do sol, 25x25, 512], [árvore, 10x10, 256]

Continua na próxima página ...

Tabela A.1 Cenários dos experimentos (continuação).

ID	Cenário
43	[avião, 30x30, 256], [construção, 10x10, 256], [campo, 25x25, 512], [fogos de artifício, 20x20, 512], [flor, 20x20, 512], [corrida de motos, 15x15, 64], [moto, 25x25, 64], [pôr do sol, 25x25, 512], [árvore, 10x10, 512]
44	[avião, 30x30, 256], [construção, 10x10, 256], [carro, 20x20, 256], [campo, 25x25, 512], [fogos de artifício, 30x30, 512], [flor, 20x20, 512], [corrida de motos, 15x15, 64], [moto, 25x25, 64], [pôr do sol, 25x25, 512], [árvore, 10x10, 512]
45	[avião, 30x30, 256], [praia, 25x25, 512], [construção, 10x10, 256], [carro, 20x20, 256], [campo, 25x25, 512], [fogos de artifício, 30x30, 512], [flor, 20x20, 512], [corrida de motos, 15x15, 64], [moto, 25x25, 64], [pôr do sol, 25x25, 512], [árvore, 10x10, 512]
46	[avião, 25x25, 256], [construção, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 15x15, 512], [moto, 20x20, 512]
47	[avião, 25x25, 256], [construção, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 15x15, 512], [moto, 20x20, 512], [pôr do sol, 20x20, 512]
48	[avião, 20x20, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 20x20, 512], [corrida de motos, 20x20, 512], [moto, 25x25, 512], [pôr do sol, 25x25, 512], [árvore, 15x15, 512]
49	[avião, 20x20, 512], [praia, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 20x20, 512], [corrida de motos, 20x20, 512], [moto, 25x25, 512], [pôr do sol, 30x30, 256], [árvore, 15x15, 512]
50	[avião, 25x25, 128], [praia, 15x15, 256], [construção, 10x10, 512], [campo, 5x5, 256], [fogos de artifício, 5x5, 512], [flor, 15x15, 256], [corrida de motos, 15x15, 256], [moto, 25x25, 128], [montanha, 15x15, 256], [pôr do sol, 20x20, 256]
51	[avião, 25x25, 128], [praia, 15x15, 256], [construção, 10x10, 512], [campo, 5x5, 256], [fogos de artifício, 5x5, 512], [flor, 15x15, 256], [corrida de motos, 15x15, 256], [moto, 25x25, 128], [montanha, 15x15, 256], [pôr do sol, 20x20, 256], [árvore, 5x5, 512]
52	[avião, 25x25, 128], [praia, 15x15, 256], [construção, 10x10, 512], [carro, 10x10, 512], [campo, 5x5, 256], [fogos de artifício, 5x5, 512], [flor, 15x15, 256], [corrida de motos, 15x15, 256], [moto, 25x25, 128], [montanha, 15x15, 256], [pôr do sol, 20x20, 256], [árvore, 5x5, 512]
53	[avião, 30x30, 128], [carro, 15x15, 512], [campo, 10x10, 512], [fogos de artifício, 15x15, 256], [flor, 15x15, 512], [moto, 15x15, 512], [árvore, 10x10, 512]
54	[avião, 25x25, 256], [campo, 10x10, 512], [fogos de artifício, 15x15, 256], [flor, 15x15, 512], [corrida de motos, 15x15, 512], [moto, 25x25, 256], [pôr do sol, 25x25, 256], [árvore, 10x10, 512]
55	[avião, 25x25, 256], [carro, 15x15, 512], [campo, 10x10, 512], [fogos de artifício, 15x15, 256], [flor, 15x15, 512], [corrida de motos, 15x15, 512], [moto, 25x25, 256], [pôr do sol, 25x25, 256], [árvore, 10x10, 512]
56	[avião, 25x25, 256], [praia, 30x30, 128], [carro, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 256], [flor, 15x15, 512], [corrida de motos, 15x15, 512], [moto, 25x25, 256], [pôr do sol, 25x25, 256], [árvore, 10x10, 512]
57	[avião, 25x25, 256], [praia, 30x30, 128], [construção, 15x15, 512], [carro, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 256], [flor, 15x15, 512], [corrida de motos, 15x15, 512], [moto, 25x25, 256], [pôr do sol, 25x25, 256], [árvore, 10x10, 512]
58	[avião, 25x25, 256], [praia, 30x30, 128], [construção, 15x15, 512], [carro, 15x15, 512], [campo, 10x10, 512], [fogos de artifício, 15x15, 256], [flor, 15x15, 512], [corrida de motos, 15x15, 512], [moto, 25x25, 256], [montanha, 15x15, 512], [pôr do sol, 25x25, 256], [árvore, 10x10, 512]
59	[avião, 30x30, 512], [fogos de artifício, 30x30, 256], [flor, 25x25, 256], [corrida de motos, 5x5, 256], [moto, 15x15, 512], [pôr do sol, 5x5, 128], [árvore, 5x5, 256]
60	[avião, 20x20, 256], [praia, 5x5, 128], [fogos de artifício, 30x30, 256], [flor, 25x25, 256], [corrida de motos, 10x10, 256], [moto, 10x10, 512], [pôr do sol, 5x5, 128], [árvore, 5x5, 256]
61	[avião, 20x20, 256], [praia, 5x5, 128], [construção, 5x5, 256], [fogos de artifício, 30x30, 256], [flor, 25x25, 256], [corrida de motos, 10x10, 256], [moto, 10x10, 512], [pôr do sol, 5x5, 128], [árvore, 5x5, 256]
62	[avião, 20x20, 512], [praia, 10x10, 256], [construção, 5x5, 512], [campo, 5x5, 256], [fogos de artifício, 25x25, 512], [flor, 20x20, 512], [corrida de motos, 10x10, 512], [moto, 5x5, 512], [pôr do sol, 10x10, 256], [árvore, 5x5, 512]
63	[avião, 25x25, 256], [praia, 5x5, 128], [construção, 5x5, 256], [carro, 10x10, 256], [campo, 5x5, 128], [fogos de artifício, 15x15, 128], [flor, 10x10, 128], [corrida de motos, 10x10, 256], [moto, 10x10, 512], [pôr do sol, 10x10, 128], [árvore, 5x5, 256]
64	[avião, 25x25, 128], [construção, 10x10, 512], [campo, 5x5, 256], [fogos de artifício, 10x10, 512], [flor, 15x15, 256], [moto, 20x20, 256], [pôr do sol, 30x30, 128]
65	[praia, 30x30, 512], [construção, 10x10, 512], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [corrida de motos, 15x15, 512], [moto, 30x30, 64], [árvore, 5x5, 512]

Continua na próxima página . . .

Tabela A.1 Cenários dos experimentos (continuação).

ID	Cenário
66	[avião, 30x30, 512], [praia, 30x30, 512], [construção, 10x10, 256], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [corrida de motos, 15x15, 512], [moto, 30x30, 64], [árvore, 5x5, 512]
67	[avião, 30x30, 512], [praia, 30x30, 512], [construção, 10x10, 256], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corrida de motos, 15x15, 512], [moto, 30x30, 64], [pôr do sol, 30x30, 512], [árvore, 5x5, 512]
68	[avião, 30x30, 256], [praia, 30x30, 512], [construção, 10x10, 256], [campo, 30x30, 512], [fogos de artifício, 25x25, 512], [flor, 25x25, 512], [corrida de motos, 20x20, 256], [moto, 30x30, 64], [pôr do sol, 30x30, 512], [árvore, 5x5, 512]
69	[avião, 25x25, 128], [construção, 10x10, 512], [campo, 5x5, 256], [fogos de artifício, 10x10, 512], [flor, 15x15, 256], [moto, 20x20, 256], [pôr do sol, 30x30, 128]
70	[praia, 25x25, 128], [construção, 10x10, 512], [fogos de artifício, 5x5, 512], [flor, 15x15, 256], [corrida de motos, 10x10, 512], [moto, 25x25, 128], [árvore, 5x5, 512]
71	[praia, 25x25, 128], [construção, 10x10, 512], [fogos de artifício, 5x5, 512], [flor, 15x15, 256], [corrida de motos, 10x10, 512], [moto, 25x25, 128], [pôr do sol, 30x30, 128], [árvore, 5x5, 512]
72	[avião, 25x25, 512], [praia, 20x20, 512], [construção, 20x20, 512], [campo, 15x15, 512], [fogos de artifício, 20x20, 256], [flor, 25x25, 512], [corrida de motos, 25x25, 512], [moto, 25x25, 512], [árvore, 15x15, 512]
73	[avião, 20x20, 256], [praia, 5x5, 512], [construção, 10x10, 512], [fogos de artifício, 5x5, 512], [flor, 15x15, 256], [corrida de motos, 10x10, 512], [moto, 20x20, 256], [montanha, 10x10, 512], [pôr do sol, 30x30, 128], [árvore, 5x5, 512]
74	[avião, 25x25, 512], [praia, 20x20, 512], [construção, 20x20, 512], [campo, 15x15, 512], [fogos de artifício, 20x20, 256], [flor, 25x25, 512], [corrida de motos, 25x25, 512], [moto, 25x25, 512], [montanha, 10x10, 512], [pôr do sol, 30x30, 128], [árvore, 15x15, 512]
75	[avião, 5x5, 256], [construção, 5x5, 512], [campo, 5x5, 256], [fogos de artifício, 25x25, 512], [flor, 20x20, 512], [corrida de motos, 10x10, 512], [moto, 10x10, 512], [árvore, 5x5, 512]
76	[avião, 5x5, 256], [construção, 5x5, 512], [campo, 5x5, 256], [fogos de artifício, 20x20, 512], [flor, 20x20, 512], [corrida de motos, 10x10, 512], [moto, 10x10, 512], [pôr do sol, 10x10, 256], [árvore, 5x5, 512]
77	[avião, 20x20, 512], [praia, 5x5, 256], [construção, 5x5, 512], [campo, 5x5, 256], [fogos de artifício, 20x20, 512], [flor, 20x20, 512], [corrida de motos, 5x5, 512], [moto, 10x10, 512], [pôr do sol, 10x10, 256], [árvore, 5x5, 512]
78	[avião, 20x20, 512], [praia, 5x5, 256], [construção, 5x5, 512], [carro, 5x5, 512], [campo, 5x5, 256], [fogos de artifício, 20x20, 512], [flor, 20x20, 512], [corrida de motos, 5x5, 512], [moto, 10x10, 512], [pôr do sol, 10x10, 256], [árvore, 5x5, 512]
79	[avião, 25x25, 256], [construção, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 15x15, 512], [moto, 20x20, 512], [pôr do sol, 25x25, 256]
80	[avião, 25x25, 256], [construção, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 15x15, 512], [moto, 20x20, 512], [montanha, 15x15, 512], [pôr do sol, 25x25, 256]
81	[avião, 25x25, 512], [construção, 15x15, 128], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [moto, 30x30, 64], [pôr do sol, 25x25, 512], [árvore, 5x5, 512]
82	[avião, 30x30, 128], [construção, 15x15, 128], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [moto, 30x30, 64], [pôr do sol, 25x25, 512], [árvore, 5x5, 512]
83	[avião, 25x25, 256], [construção, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 15x15, 512], [moto, 20x20, 512], [pôr do sol, 25x25, 256]
84	[avião, 25x25, 256], [construção, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 15x15, 512], [flor, 15x15, 512], [moto, 20x20, 512], [montanha, 15x15, 512], [pôr do sol, 25x25, 256]
85	[construção, 10x10, 512], [fogos de artifício, 5x5, 512], [flor, 15x15, 256], [corrida de motos, 10x10, 512], [moto, 25x25, 128], [pôr do sol, 30x30, 128], [árvore, 5x5, 512]
86	[construção, 10x10, 512], [fogos de artifício, 5x5, 512], [flor, 15x15, 256], [corrida de motos, 10x10, 512], [moto, 25x25, 128], [montanha, 15x15, 256], [pôr do sol, 30x30, 128], [árvore, 5x5, 512]
87	[praia, 5x5, 512], [construção, 10x10, 512], [fogos de artifício, 5x5, 512], [flor, 15x15, 256], [corrida de motos, 10x10, 512], [moto, 25x25, 128], [montanha, 15x15, 256], [pôr do sol, 30x30, 128], [árvore, 5x5, 512]
88	[praia, 30x30, 256], [construção, 20x20, 512], [campo, 15x15, 512], [fogos de artifício, 25x25, 128], [flor, 25x25, 512], [corrida de motos, 30x30, 256], [moto, 25x25, 512], [montanha, 15x15, 512], [pôr do sol, 30x30, 128], [árvore, 15x15, 512]
89	[avião, 30x30, 512], [construção, 5x5, 256], [campo, 5x5, 128], [fogos de artifício, 30x30, 512], [flor, 25x25, 256], [corrida de motos, 10x10, 256], [moto, 15x15, 512], [árvore, 5x5, 256]
90	[avião, 30x30, 512], [construção, 5x5, 256], [campo, 5x5, 128], [fogos de artifício, 30x30, 256], [flor, 25x25, 256], [corrida de motos, 10x10, 256], [moto, 15x15, 512], [pôr do sol, 5x5, 128], [árvore, 5x5, 256]

Continua na próxima página . . .

Tabela A.1 Cenários dos experimentos (continuação).

ID	Cenário
91	[avião, 30x30, 512], [construção, 5x5, 256], [campo, 5x5, 128], [fogos de artifício, 30x30, 256], [flor, 25x25, 256], [corrida de motos, 10x10, 256], [moto, 20x20, 512], [montanha, 15x15, 256], [pôr do sol, 5x5, 128], [árvore, 5x5, 256]
92	[avião, 30x30, 512], [construção, 5x5, 256], [carro, 25x25, 512], [campo, 5x5, 128], [fogos de artifício, 30x30, 256], [flor, 25x25, 256], [corrida de motos, 10x10, 256], [moto, 15x15, 512], [montanha, 15x15, 256], [pôr do sol, 5x5, 128], [árvore, 5x5, 256]
93	[avião, 30x30, 512], [praia, 5x5, 128], [construção, 5x5, 256], [carro, 25x25, 512], [campo, 5x5, 128], [fogos de artifício, 30x30, 256], [flor, 25x25, 256], [corrida de motos, 10x10, 256], [moto, 15x15, 512], [montanha, 15x15, 256], [pôr do sol, 5x5, 128], [árvore, 5x5, 256]
94	[avião, 25x25, 512], [fogos de artifício, 20x20, 256], [flor, 20x20, 512], [moto, 25x25, 512], [pôr do sol, 30x30, 256], [árvore, 15x15, 512]
95	[avião, 30x30, 256], [campo, 5x5, 512], [fogos de artifício, 20x20, 256], [flor, 20x20, 512], [moto, 30x30, 256], [pôr do sol, 30x30, 256], [árvore, 15x15, 512]
96	[avião, 30x30, 256], [campo, 5x5, 512], [fogos de artifício, 20x20, 256], [flor, 20x20, 512], [corrida de motos, 20x20, 512], [moto, 30x30, 256], [pôr do sol, 30x30, 256], [árvore, 15x15, 512]
97	[avião, 20x20, 512], [praia, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 20x20, 256], [flor, 20x20, 512], [corrida de motos, 20x20, 512], [moto, 20x20, 512], [pôr do sol, 20x20, 512], [árvore, 15x15, 512]
98	[avião, 25x25, 128], [praia, 15x15, 256], [construção, 15x15, 256], [campo, 5x5, 256], [fogos de artifício, 10x10, 512], [flor, 15x15, 256], [moto, 20x20, 256], [montanha, 15x15, 256], [pôr do sol, 20x20, 256], [árvore, 5x5, 512]
99	[avião, 25x25, 128], [praia, 15x15, 256], [construção, 10x10, 512], [campo, 5x5, 256], [fogos de artifício, 10x10, 512], [flor, 15x15, 256], [corrida de motos, 10x10, 512], [moto, 20x20, 256], [montanha, 15x15, 256], [pôr do sol, 20x20, 256], [árvore, 5x5, 512]
100	[avião, 30x30, 512], [construção, 10x10, 64], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [moto, 20x20, 64], [árvore, 5x5, 512]
101	[praia, 30x30, 512], [construção, 10x10, 512], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [corrida de motos, 25x25, 256], [moto, 25x25, 128], [árvore, 5x5, 512]
102	[praia, 30x30, 512], [construção, 10x10, 512], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [corrida de motos, 25x25, 256], [moto, 25x25, 128], [pôr do sol, 30x30, 512], [árvore, 5x5, 512]
103	[avião, 30x30, 512], [construção, 15x15, 256], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corrida de motos, 20x20, 256], [moto, 25x25, 128], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 5x5, 512]
104	[avião, 30x30, 256], [construção, 15x15, 128], [carro, 20x20, 128], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corrida de motos, 20x20, 64], [moto, 25x25, 64], [pôr do sol, 30x30, 512], [árvore, 5x5, 512]
105	[avião, 30x30, 256], [praia, 30x30, 512], [construção, 15x15, 128], [carro, 20x20, 128], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corrida de motos, 20x20, 64], [moto, 25x25, 64], [pôr do sol, 30x30, 512], [árvore, 5x5, 512]
106	[avião, 30x30, 256], [praia, 30x30, 512], [construção, 15x15, 128], [carro, 20x20, 128], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corrida de motos, 20x20, 64], [moto, 25x25, 64], [montanha, 25x25, 256], [pôr do sol, 30x30, 512], [árvore, 5x5, 512]
107	[avião, 25x25, 512], [fogos de artifício, 20x20, 256], [flor, 20x20, 512], [moto, 25x25, 512], [pôr do sol, 30x30, 256], [árvore, 15x15, 512]
108	[avião, 30x30, 256], [campo, 5x5, 512], [fogos de artifício, 20x20, 256], [flor, 20x20, 512], [moto, 30x30, 256], [pôr do sol, 30x30, 256], [árvore, 15x15, 512]
109	[avião, 30x30, 256], [campo, 5x5, 512], [fogos de artifício, 20x20, 256], [flor, 20x20, 512], [corrida de motos, 20x20, 512], [moto, 30x30, 256], [pôr do sol, 30x30, 256], [árvore, 15x15, 512]
110	[avião, 20x20, 512], [praia, 15x15, 512], [campo, 5x5, 512], [fogos de artifício, 20x20, 256], [flor, 20x20, 512], [corrida de motos, 20x20, 512], [moto, 20x20, 512], [pôr do sol, 20x20, 512], [árvore, 15x15, 512]
111	[avião, 25x25, 128], [praia, 15x15, 256], [construção, 15x15, 256], [campo, 5x5, 256], [fogos de artifício, 10x10, 512], [flor, 15x15, 256], [moto, 20x20, 256], [montanha, 15x15, 256], [pôr do sol, 20x20, 256], [árvore, 5x5, 512]
112	[avião, 25x25, 128], [praia, 15x15, 256], [construção, 10x10, 512], [campo, 5x5, 256], [fogos de artifício, 10x10, 512], [flor, 15x15, 256], [corrida de motos, 10x10, 512], [moto, 20x20, 256], [montanha, 15x15, 256], [pôr do sol, 20x20, 256], [árvore, 5x5, 512]
113	[avião, 30x30, 512], [praia, 20x20, 512], [construção, 20x20, 512], [carro, 25x25, 512], [fogos de artifício, 25x25, 256], [flor, 25x25, 512], [corrida de motos, 25x25, 512], [moto, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]

Continua na próxima página . . .

Tabela A.1 Cenários dos experimentos (continuação).

ID	Cenário
114	[avião, 30x30, 512], [praia, 20x20, 512], [construção, 20x20, 512], [carro, 25x25, 512], [fogos de artifício, 25x25, 256], [flor, 25x25, 512], [corrida de motos, 25x25, 512], [moto, 25x25, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
115	[praia, 30x30, 512], [construção, 15x15, 256], [carro, 15x15, 256], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [moto, 15x15, 512], [pôr do sol, 30x30, 512], [árvore, 25x25, 512]
116	[praia, 30x30, 512], [construção, 15x15, 256], [carro, 15x15, 256], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [moto, 15x15, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 25x25, 512]
117	[avião, 25x25, 256], [praia, 30x30, 512], [construção, 15x15, 256], [carro, 15x15, 256], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [moto, 15x15, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 25x25, 512]
118	[avião, 30x30, 512], [carro, 30x30, 512], [campo, 20x20, 512], [fogos de artifício, 25x25, 512], [flor, 30x30, 512], [corrida de motos, 30x30, 512], [montanha, 25x25, 512], [árvore, 20x20, 512]
119	[avião, 30x30, 512], [carro, 30x30, 512], [campo, 20x20, 512], [fogos de artifício, 25x25, 512], [flor, 30x30, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 25x25, 512], [árvore, 20x20, 512]
120	[avião, 30x30, 512], [praia, 20x20, 512], [carro, 30x30, 512], [fogos de artifício, 25x25, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
121	[avião, 30x30, 512], [praia, 20x20, 512], [carro, 30x30, 512], [campo, 5x5, 512], [fogos de artifício, 25x25, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
122	[avião, 30x30, 512], [praia, 20x20, 512], [construção, 20x20, 512], [carro, 30x30, 512], [campo, 5x5, 512], [fogos de artifício, 25x25, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
123	[avião, 30x30, 512], [construção, 20x20, 256], [campo, 30x30, 512], [fogos de artifício, 25x25, 256], [moto, 30x30, 64], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
124	[avião, 20x20, 512], [construção, 10x10, 256], [campo, 25x25, 512], [fogos de artifício, 15x15, 512], [flor, 20x20, 512], [corrida de motos, 15x15, 128], [moto, 20x20, 64], [pôr do sol, 25x25, 512], [árvore, 5x5, 512]
125	[avião, 25x25, 512], [praia, 25x25, 512], [construção, 10x10, 256], [campo, 25x25, 512], [fogos de artifício, 15x15, 512], [flor, 20x20, 512], [corrida de motos, 10x10, 512], [moto, 30x30, 128], [pôr do sol, 25x25, 512], [árvore, 5x5, 512]
126	[avião, 30x30, 512], [praia, 30x30, 512], [construção, 20x20, 256], [campo, 30x30, 512], [fogos de artifício, 25x25, 256], [flor, 25x25, 512], [corrida de motos, 15x15, 512], [moto, 30x30, 64], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 5x5, 128]
127	[avião, 30x30, 512], [praia, 30x30, 512], [construção, 10x10, 256], [carro, 20x20, 256], [campo, 30x30, 512], [fogos de artifício, 25x25, 256], [flor, 25x25, 512], [corrida de motos, 10x10, 128], [moto, 30x30, 64], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 5x5, 128]
128	[avião, 30x30, 512], [carro, 30x30, 512], [campo, 20x20, 512], [fogos de artifício, 25x25, 512], [flor, 30x30, 512], [corrida de motos, 30x30, 512], [montanha, 25x25, 512], [árvore, 20x20, 512]
129	[avião, 30x30, 512], [carro, 30x30, 512], [campo, 20x20, 512], [fogos de artifício, 25x25, 512], [flor, 30x30, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 25x25, 512], [árvore, 20x20, 512]
130	[avião, 30x30, 512], [praia, 20x20, 512], [carro, 30x30, 512], [fogos de artifício, 25x25, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
131	[avião, 30x30, 512], [praia, 20x20, 512], [carro, 30x30, 512], [campo, 5x5, 512], [fogos de artifício, 25x25, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
132	[avião, 30x30, 512], [praia, 20x20, 512], [construção, 20x20, 512], [carro, 30x30, 512], [campo, 5x5, 512], [fogos de artifício, 25x25, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
133	[avião, 30x30, 512], [construção, 20x20, 512], [carro, 25x25, 512], [campo, 20x20, 256], [fogos de artifício, 15x15, 512], [flor, 25x25, 512], [corrida de motos, 25x25, 512], [moto, 25x25, 512], [montanha, 20x20, 512], [árvore, 15x15, 512]
134	[avião, 30x30, 512], [praia, 20x20, 512], [construção, 20x20, 512], [carro, 25x25, 512], [campo, 15x15, 512], [fogos de artifício, 15x15, 512], [flor, 25x25, 512], [corrida de motos, 25x25, 512], [moto, 25x25, 512], [montanha, 20x20, 512], [árvore, 15x15, 512]

Continua na próxima página . . .

Tabela A.1 Cenários dos experimentos (continuação).

ID	Cenário
135	[avião, 30x30, 512], [praia, 20x20, 512], [construção, 20x20, 512], [carro, 25x25, 512], [campo, 15x15, 512], [fogos de artifício, 15x15, 512], [flor, 25x25, 512], [corrida de motos, 25x25, 512], [moto, 25x25, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
136	[praia, 30x30, 512], [carro, 20x20, 512], [campo, 30x30, 512], [fogos de artifício, 25x25, 512], [flor, 30x30, 512], [corrida de motos, 20x20, 512], [moto, 15x15, 512], [pôr do sol, 30x30, 512], [árvore, 25x25, 512]
137	[avião, 25x25, 512], [praia, 30x30, 512], [carro, 20x20, 512], [campo, 30x30, 512], [fogos de artifício, 25x25, 512], [flor, 30x30, 512], [corrida de motos, 20x20, 512], [moto, 15x15, 512], [pôr do sol, 30x30, 512], [árvore, 25x25, 512]
138	[avião, 25x25, 512], [praia, 30x30, 512], [construção, 20x20, 512], [carro, 20x20, 512], [campo, 30x30, 512], [fogos de artifício, 25x25, 512], [flor, 30x30, 512], [corrida de motos, 20x20, 512], [moto, 15x15, 512], [pôr do sol, 30x30, 512], [árvore, 25x25, 512]
139	[avião, 30x30, 512], [carro, 25x25, 512], [campo, 15x15, 512], [fogos de artifício, 20x20, 512], [flor, 25x25, 512], [moto, 30x30, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
140	[avião, 30x30, 512], [praia, 20x20, 512], [carro, 25x25, 512], [campo, 15x15, 512], [fogos de artifício, 25x25, 256], [flor, 25x25, 512], [corrida de motos, 25x25, 512], [moto, 30x30, 512], [montanha, 20x20, 512], [árvore, 15x15, 512]
141	[avião, 30x30, 512], [praia, 20x20, 512], [carro, 25x25, 512], [campo, 15x15, 512], [fogos de artifício, 20x20, 512], [flor, 25x25, 512], [corrida de motos, 25x25, 512], [moto, 30x30, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
142	[praia, 30x30, 512], [construção, 10x10, 256], [fogos de artifício, 25x25, 512], [flor, 25x25, 512], [corrida de motos, 10x10, 512], [moto, 25x25, 64], [pôr do sol, 30x30, 512], [árvore, 5x5, 512]
143	[avião, 30x30, 512], [praia, 30x30, 512], [construção, 10x10, 256], [fogos de artifício, 25x25, 512], [flor, 25x25, 512], [corrida de motos, 10x10, 512], [moto, 25x25, 64], [pôr do sol, 30x30, 512], [árvore, 5x5, 512]
144	[praia, 30x30, 512], [carro, 25x25, 256], [campo, 30x30, 512], [fogos de artifício, 25x25, 512], [flor, 30x30, 512], [corrida de motos, 15x15, 512], [moto, 30x30, 64], [montanha, 30x30, 256], [pôr do sol, 30x30, 512], [árvore, 5x5, 256]
145	[avião, 25x25, 512], [praia, 30x30, 512], [construção, 10x10, 128], [carro, 25x25, 256], [campo, 30x30, 512], [fogos de artifício, 25x25, 512], [flor, 30x30, 512], [corrida de motos, 10x10, 256], [moto, 25x25, 64], [pôr do sol, 30x30, 512], [árvore, 10x10, 512]
146	[avião, 30x30, 512], [praia, 30x30, 512], [construção, 10x10, 128], [carro, 25x25, 256], [campo, 30x30, 512], [fogos de artifício, 25x25, 512], [flor, 30x30, 512], [corrida de motos, 10x10, 256], [moto, 25x25, 64], [montanha, 30x30, 256], [pôr do sol, 30x30, 512], [árvore, 10x10, 512]
147	[avião, 30x30, 512], [carro, 25x25, 512], [campo, 15x15, 512], [fogos de artifício, 20x20, 512], [flor, 25x25, 512], [moto, 30x30, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
148	[avião, 30x30, 512], [praia, 20x20, 512], [carro, 25x25, 512], [campo, 15x15, 512], [fogos de artifício, 25x25, 256], [flor, 25x25, 512], [corrida de motos, 25x25, 512], [moto, 30x30, 512], [montanha, 20x20, 512], [árvore, 15x15, 512]
149	[avião, 30x30, 512], [praia, 20x20, 512], [carro, 25x25, 512], [campo, 15x15, 512], [fogos de artifício, 20x20, 512], [flor, 25x25, 512], [corrida de motos, 25x25, 512], [moto, 30x30, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
150	[avião, 30x30, 512], [praia, 20x20, 512], [construção, 20x20, 512], [carro, 30x30, 256], [campo, 15x15, 512], [fogos de artifício, 20x20, 512], [flor, 25x25, 512], [moto, 25x25, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
151	[avião, 30x30, 512], [praia, 20x20, 512], [construção, 20x20, 512], [carro, 25x25, 512], [campo, 15x15, 512], [fogos de artifício, 15x15, 512], [flor, 25x25, 512], [corrida de motos, 25x25, 512], [moto, 25x25, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
152	[avião, 25x25, 512], [praia, 30x30, 512], [construção, 5x5, 256], [fogos de artifício, 25x25, 512], [flor, 30x30, 512], [corrida de motos, 5x5, 256], [moto, 10x10, 512], [pôr do sol, 30x30, 512], [árvore, 5x5, 256]
153	[avião, 25x25, 512], [praia, 30x30, 512], [construção, 20x20, 512], [carro, 20x20, 512], [campo, 30x30, 512], [fogos de artifício, 25x25, 512], [flor, 30x30, 512], [moto, 10x10, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 256]
154	[avião, 25x25, 512], [praia, 30x30, 512], [construção, 20x20, 512], [carro, 20x20, 512], [campo, 30x30, 512], [fogos de artifício, 25x25, 512], [flor, 30x30, 512], [corrida de motos, 20x20, 512], [moto, 10x10, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 256]

Continua na próxima página . . .

Tabela A.1 Cenários dos experimentos (continuação).

ID	Cenário
155	[avião, 25x25, 512], [praia, 30x30, 512], [construção, 20x20, 512], [carro, 20x20, 512], [campo, 30x30, 512], [fogos de artifício, 25x25, 512], [flor, 30x30, 512], [corrida de motos, 20x20, 512], [moto, 10x10, 512], [montanha, 30x30, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 256]
156	[avião, 30x30, 512], [construção, 25x25, 512], [campo, 15x15, 512], [fogos de artifício, 25x25, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
157	[avião, 30x30, 512], [praia, 20x20, 512], [construção, 25x25, 512], [campo, 15x15, 512], [fogos de artifício, 25x25, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
158	[avião, 30x30, 512], [praia, 20x20, 512], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 15x15, 512], [fogos de artifício, 25x25, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
159	[avião, 30x30, 512], [praia, 20x20, 512], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 15x15, 512], [fogos de artifício, 25x25, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 25x25, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
160	[avião, 30x30, 256], [construção, 15x15, 256], [campo, 30x30, 512], [fogos de artifício, 20x20, 512], [flor, 25x25, 512], [corrida de motos, 20x20, 128], [moto, 25x25, 64], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
161	[avião, 30x30, 256], [praia, 30x30, 512], [construção, 15x15, 256], [campo, 30x30, 512], [fogos de artifício, 20x20, 512], [flor, 25x25, 512], [corrida de motos, 20x20, 128], [moto, 25x25, 64], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
162	[avião, 30x30, 512], [construção, 25x25, 512], [campo, 15x15, 512], [fogos de artifício, 25x25, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
163	[avião, 30x30, 512], [praia, 20x20, 512], [construção, 25x25, 512], [campo, 15x15, 512], [fogos de artifício, 25x25, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
164	[avião, 30x30, 512], [praia, 20x20, 512], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 15x15, 512], [fogos de artifício, 25x25, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
165	[avião, 30x30, 512], [praia, 20x20, 512], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 15x15, 512], [fogos de artifício, 25x25, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 25x25, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
166	[avião, 30x30, 512], [construção, 20x20, 512], [carro, 25x25, 512], [campo, 15x15, 512], [fogos de artifício, 20x20, 256], [flor, 25x25, 512], [corrida de motos, 25x25, 512], [moto, 25x25, 512], [montanha, 25x25, 256], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
167	[avião, 30x30, 512], [praia, 20x20, 512], [construção, 20x20, 512], [carro, 25x25, 512], [campo, 15x15, 512], [fogos de artifício, 20x20, 256], [flor, 25x25, 512], [corrida de motos, 25x25, 512], [moto, 25x25, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
168	[avião, 25x25, 512], [praia, 5x5, 128], [construção, 5x5, 256], [fogos de artifício, 30x30, 256], [flor, 25x25, 256], [corrida de motos, 10x10, 256], [moto, 15x15, 512], [pôr do sol, 10x10, 128], [árvore, 5x5, 256]
169	[praia, 30x30, 512], [carro, 15x15, 512], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corrida de motos, 15x15, 512], [moto, 10x10, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
170	[avião, 25x25, 512], [praia, 30x30, 512], [carro, 15x15, 512], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corrida de motos, 15x15, 512], [moto, 10x10, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
171	[avião, 25x25, 512], [praia, 30x30, 512], [construção, 5x5, 256], [carro, 20x20, 512], [campo, 5x5, 128], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [corrida de motos, 20x20, 512], [moto, 10x10, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 5x5, 256]
172	[avião, 30x30, 512], [praia, 15x15, 512], [carro, 25x25, 512], [fogos de artifício, 20x20, 256], [flor, 20x20, 512], [corrida de motos, 25x25, 512], [moto, 30x30, 512], [montanha, 20x20, 512], [pôr do sol, 25x25, 512], [árvore, 15x15, 512]
173	[avião, 30x30, 128], [praia, 30x30, 512], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corrida de motos, 20x20, 128], [moto, 30x30, 128], [pôr do sol, 30x30, 512], [árvore, 10x10, 512]
174	[avião, 30x30, 128], [praia, 30x30, 512], [construção, 15x15, 256], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corrida de motos, 20x20, 128], [moto, 30x30, 64], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]

Continua na próxima página ...

Tabela A.1 Cenários dos experimentos (continuação).

ID	Cenário
175	[avião, 30x30, 512], [praia, 15x15, 512], [carro, 25x25, 512], [fogos de artifício, 20x20, 256], [flor, 20x20, 512], [corrida de motos, 25x25, 512], [moto, 30x30, 512], [montanha, 20x20, 512], [pôr do sol, 25x25, 512], [árvore, 15x15, 512]
176	[avião, 30x30, 512], [praia, 20x20, 512], [construção, 20x20, 512], [carro, 25x25, 512], [campo, 15x15, 512], [fogos de artifício, 20x20, 256], [flor, 25x25, 512], [corrida de motos, 25x25, 512], [moto, 25x25, 512], [montanha, 20x20, 512], [árvore, 15x15, 512]
177	[avião, 30x30, 512], [praia, 20x20, 512], [construção, 20x20, 512], [carro, 25x25, 512], [campo, 15x15, 512], [fogos de artifício, 20x20, 512], [flor, 25x25, 512], [corrida de motos, 25x25, 512], [moto, 25x25, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
178	[avião, 30x30, 512], [praia, 5x5, 128], [construção, 5x5, 256], [fogos de artifício, 25x25, 256], [flor, 30x30, 256], [corrida de motos, 5x5, 256], [moto, 15x15, 512], [pôr do sol, 10x10, 128], [árvore, 5x5, 256]
179	[avião, 30x30, 512], [praia, 5x5, 128], [construção, 5x5, 256], [carro, 25x25, 512], [fogos de artifício, 20x20, 128], [flor, 30x30, 256], [corrida de motos, 10x10, 256], [moto, 15x15, 512], [pôr do sol, 10x10, 128], [árvore, 5x5, 256]
180	[avião, 30x30, 512], [praia, 5x5, 128], [construção, 5x5, 256], [carro, 25x25, 512], [fogos de artifício, 20x20, 128], [flor, 30x30, 256], [corrida de motos, 10x10, 256], [moto, 15x15, 512], [montanha, 30x30, 512], [pôr do sol, 10x10, 128], [árvore, 5x5, 256]
181	[avião, 30x30, 512], [praia, 10x10, 128], [construção, 5x5, 256], [carro, 25x25, 512], [campo, 20x20, 128], [fogos de artifício, 15x15, 128], [flor, 30x30, 256], [corrida de motos, 10x10, 256], [moto, 15x15, 512], [montanha, 30x30, 512], [pôr do sol, 10x10, 128], [árvore, 5x5, 256]
182	[avião, 25x25, 512], [praia, 15x15, 512], [construção, 20x20, 512], [fogos de artifício, 20x20, 256], [flor, 20x20, 512], [moto, 30x30, 256], [montanha, 20x20, 512], [pôr do sol, 30x30, 256], [árvore, 15x15, 512]
183	[avião, 30x30, 256], [praia, 30x30, 512], [construção, 20x20, 256], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [moto, 30x30, 64], [árvore, 20x20, 512]
184	[avião, 30x30, 256], [praia, 30x30, 512], [carro, 25x25, 256], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [corrida de motos, 20x20, 256], [moto, 30x30, 64], [árvore, 20x20, 512]
185	[avião, 30x30, 256], [praia, 30x30, 512], [construção, 15x15, 512], [campo, 30x30, 512], [fogos de artifício, 25x25, 512], [flor, 30x30, 512], [corrida de motos, 20x20, 256], [moto, 30x30, 64], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
186	[avião, 30x30, 256], [praia, 30x30, 512], [construção, 15x15, 512], [campo, 30x30, 512], [fogos de artifício, 25x25, 512], [flor, 30x30, 512], [corrida de motos, 20x20, 256], [moto, 30x30, 64], [montanha, 30x30, 256], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
187	[avião, 30x30, 256], [praia, 30x30, 512], [construção, 10x10, 512], [carro, 25x25, 256], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [corrida de motos, 20x20, 256], [moto, 30x30, 64], [montanha, 30x30, 256], [pôr do sol, 30x30, 512], [árvore, 5x5, 512]
188	[avião, 25x25, 512], [praia, 15x15, 512], [construção, 20x20, 512], [fogos de artifício, 20x20, 256], [flor, 20x20, 512], [moto, 30x30, 256], [montanha, 20x20, 512], [pôr do sol, 30x30, 256], [árvore, 15x15, 512]
189	[praia, 30x30, 256], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 25x25, 256], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 25x25, 512], [árvore, 15x15, 512]
190	[praia, 30x30, 256], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 15x15, 512], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
191	[avião, 30x30, 512], [praia, 20x20, 512], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 15x15, 512], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
192	[construção, 30x30, 512], [carro, 30x30, 512], [campo, 5x5, 64], [fogos de artifício, 30x30, 256], [flor, 25x25, 128], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 30x30, 256], [pôr do sol, 25x25, 128], [árvore, 30x30, 512]
193	[avião, 30x30, 128], [construção, 30x30, 512], [carro, 30x30, 512], [campo, 5x5, 64], [fogos de artifício, 30x30, 256], [flor, 25x25, 128], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 30x30, 256], [pôr do sol, 25x25, 128], [árvore, 30x30, 512]
194	[avião, 20x20, 256], [praia, 30x30, 512], [construção, 20x20, 512], [carro, 20x20, 512], [campo, 30x30, 512], [fogos de artifício, 20x20, 512], [flor, 25x25, 256], [corrida de motos, 20x20, 512], [moto, 20x20, 512], [montanha, 30x30, 256], [pôr do sol, 30x30, 512], [árvore, 25x25, 512]

Continua na próxima página ...

Tabela A.1 Cenários dos experimentos (continuação).

ID	Cenário
195	[praia, 25x25, 512], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 20x20, 512], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 30x30, 256], [árvore, 20x20, 512]
196	[praia, 30x30, 512], [construção, 15x15, 512], [campo, 30x30, 512], [fogos de artifício, 20x20, 256], [flor, 30x30, 512], [corrida de motos, 20x20, 256], [moto, 30x30, 128], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
197	[avião, 30x30, 512], [praia, 30x30, 512], [construção, 15x15, 512], [campo, 30x30, 512], [fogos de artifício, 20x20, 256], [flor, 30x30, 512], [corrida de motos, 20x20, 256], [moto, 30x30, 128], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
198	[avião, 30x30, 512], [praia, 30x30, 512], [construção, 15x15, 512], [carro, 25x25, 256], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [corrida de motos, 15x15, 512], [moto, 30x30, 128], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
199	[praia, 25x25, 512], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 20x20, 512], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 30x30, 256], [árvore, 20x20, 512]
200	[praia, 25x25, 256], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 15x15, 512], [fogos de artifício, 25x25, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 30x30, 256], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
201	[avião, 30x30, 256], [praia, 25x25, 128], [construção, 25x25, 512], [carro, 25x25, 512], [campo, 25x25, 128], [fogos de artifício, 30x30, 512], [corrida de motos, 25x25, 512], [moto, 20x20, 512], [montanha, 30x30, 512], [pôr do sol, 30x30, 256], [árvore, 30x30, 512]
202	[avião, 30x30, 256], [praia, 25x25, 128], [construção, 25x25, 512], [carro, 25x25, 512], [campo, 25x25, 128], [fogos de artifício, 30x30, 512], [flor, 30x30, 256], [corrida de motos, 25x25, 512], [moto, 20x20, 512], [montanha, 30x30, 512], [pôr do sol, 30x30, 256], [árvore, 30x30, 512]
203	[avião, 30x30, 512], [praia, 15x15, 512], [construção, 25x25, 512], [carro, 25x25, 512], [campo, 5x5, 512], [fogos de artifício, 25x25, 512], [flor, 30x30, 256], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 25x25, 512], [pôr do sol, 25x25, 512], [árvore, 15x15, 512]
204	[avião, 30x30, 512], [praia, 30x30, 512], [construção, 20x20, 256], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [corrida de motos, 20x20, 256], [moto, 25x25, 64], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
205	[avião, 30x30, 512], [praia, 30x30, 512], [construção, 20x20, 256], [campo, 30x30, 512], [fogos de artifício, 25x25, 256], [flor, 30x30, 512], [corrida de motos, 20x20, 256], [moto, 25x25, 64], [pôr do sol, 30x30, 512], [árvore, 25x25, 512]
206	[avião, 30x30, 512], [praia, 30x30, 512], [construção, 20x20, 256], [carro, 20x20, 256], [campo, 30x30, 512], [fogos de artifício, 10x10, 512], [flor, 30x30, 512], [corrida de motos, 20x20, 256], [moto, 25x25, 64], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
207	[avião, 30x30, 512], [praia, 30x30, 512], [construção, 20x20, 256], [carro, 20x20, 256], [campo, 30x30, 512], [fogos de artifício, 10x10, 512], [flor, 30x30, 512], [corrida de motos, 20x20, 256], [moto, 25x25, 64], [montanha, 25x25, 256], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
208	[avião, 30x30, 512], [praia, 15x15, 512], [construção, 25x25, 512], [carro, 25x25, 512], [campo, 5x5, 512], [fogos de artifício, 25x25, 512], [flor, 30x30, 256], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 25x25, 512], [pôr do sol, 25x25, 512], [árvore, 15x15, 512]
209	[praia, 20x20, 512], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 15x15, 512], [fogos de artifício, 25x25, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 30x30, 256], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
210	[avião, 30x30, 512], [praia, 20x20, 512], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 15x15, 512], [fogos de artifício, 25x25, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
211	[construção, 25x25, 512], [carro, 25x25, 512], [campo, 20x20, 128], [fogos de artifício, 30x30, 512], [flor, 30x30, 256], [corrida de motos, 25x25, 512], [moto, 20x20, 512], [montanha, 30x30, 512], [pôr do sol, 30x30, 256], [árvore, 30x30, 512]
212	[avião, 30x30, 512], [construção, 25x25, 512], [carro, 25x25, 512], [campo, 20x20, 128], [fogos de artifício, 30x30, 512], [flor, 30x30, 256], [corrida de motos, 25x25, 512], [moto, 20x20, 512], [montanha, 30x30, 512], [pôr do sol, 30x30, 256], [árvore, 30x30, 512]
213	[avião, 25x25, 512], [praia, 30x30, 512], [construção, 25x25, 512], [carro, 25x25, 512], [campo, 30x30, 256], [fogos de artifício, 30x30, 512], [flor, 30x30, 256], [corrida de motos, 25x25, 512], [moto, 20x20, 512], [montanha, 30x30, 512], [pôr do sol, 30x30, 256], [árvore, 30x30, 512]

Continua na próxima página . . .

Tabela A.1 Cenários dos experimentos (continuação).

ID	Cenário
214	[praia, 30x30, 256], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 25x25, 256], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 25x25, 512], [árvore, 25x25, 256]
215	[praia, 30x30, 256], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 25x25, 256], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 25x25, 256]
216	[avião, 30x30, 512], [praia, 30x30, 256], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 25x25, 256], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 25x25, 256]
217	[avião, 25x25, 256], [praia, 30x30, 512], [construção, 10x10, 256], [campo, 30x30, 512], [fogos de artifício, 15x15, 512], [flor, 30x30, 512], [corrida de motos, 15x15, 128], [moto, 25x25, 128], [pôr do sol, 30x30, 512], [árvore, 5x5, 512]
218	[avião, 25x25, 256], [praia, 30x30, 512], [construção, 10x10, 256], [carro, 30x30, 128], [campo, 30x30, 512], [fogos de artifício, 30x30, 256], [flor, 30x30, 512], [corrida de motos, 15x15, 128], [moto, 30x30, 64], [pôr do sol, 30x30, 512], [árvore, 5x5, 512]
219	[praia, 30x30, 256], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 25x25, 256], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 25x25, 512], [árvore, 25x25, 256]
220	[praia, 30x30, 256], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 25x25, 256], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 25x25, 256]
221	[avião, 30x30, 512], [praia, 30x30, 256], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 25x25, 256], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 25x25, 256]
222	[praia, 25x25, 512], [construção, 20x20, 512], [carro, 30x30, 512], [campo, 25x25, 256], [fogos de artifício, 25x25, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
223	[avião, 30x30, 512], [praia, 30x30, 128], [construção, 20x20, 512], [carro, 25x25, 512], [campo, 20x20, 256], [fogos de artifício, 20x20, 512], [flor, 20x20, 512], [corrida de motos, 25x25, 512], [moto, 25x25, 512], [montanha, 30x30, 128], [pôr do sol, 25x25, 512], [árvore, 15x15, 512]
224	[avião, 25x25, 512], [praia, 30x30, 512], [carro, 20x20, 512], [campo, 30x30, 512], [fogos de artifício, 25x25, 512], [flor, 30x30, 512], [corrida de motos, 20x20, 512], [moto, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 25x25, 512]
225	[avião, 25x25, 512], [praia, 30x30, 512], [construção, 15x15, 256], [carro, 20x20, 512], [campo, 30x30, 512], [fogos de artifício, 25x25, 512], [flor, 30x30, 512], [corrida de motos, 20x20, 512], [moto, 20x20, 512], [pôr do sol, 30x30, 512], [árvore, 25x25, 512]
226	[avião, 30x30, 512], [praia, 30x30, 512], [construção, 15x15, 256], [carro, 20x20, 512], [campo, 25x25, 256], [fogos de artifício, 25x25, 512], [flor, 30x30, 512], [corrida de motos, 20x20, 512], [moto, 20x20, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 25x25, 512]
227	[avião, 30x30, 512], [construção, 20x20, 512], [carro, 25x25, 512], [fogos de artifício, 25x25, 512], [flor, 20x20, 512], [corrida de motos, 25x25, 512], [moto, 30x30, 512], [montanha, 20x20, 512], [pôr do sol, 25x25, 512], [árvore, 15x15, 512]
228	[avião, 30x30, 512], [construção, 20x20, 512], [carro, 25x25, 512], [campo, 20x20, 256], [fogos de artifício, 25x25, 512], [flor, 20x20, 512], [corrida de motos, 25x25, 512], [moto, 30x30, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 256], [árvore, 15x15, 512]
229	[avião, 30x30, 512], [praia, 15x15, 512], [construção, 20x20, 512], [carro, 25x25, 512], [campo, 5x5, 512], [fogos de artifício, 25x25, 512], [flor, 20x20, 512], [corrida de motos, 25x25, 512], [moto, 30x30, 512], [montanha, 20x20, 512], [pôr do sol, 25x25, 512], [árvore, 15x15, 512]
230	[avião, 30x30, 512], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [corrida de motos, 15x15, 256], [moto, 30x30, 256], [pôr do sol, 30x30, 512], [árvore, 5x5, 512]
231	[avião, 30x30, 512], [praia, 30x30, 512], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [corrida de motos, 15x15, 256], [moto, 30x30, 256], [pôr do sol, 30x30, 512], [árvore, 5x5, 512]
232	[avião, 30x30, 512], [praia, 30x30, 512], [carro, 30x30, 128], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [corrida de motos, 15x15, 256], [moto, 25x25, 64], [pôr do sol, 30x30, 512], [árvore, 5x5, 512]

Continua na próxima página ...

Tabela A.1 Cenários dos experimentos (continuação).

ID	Cenário
233	[avião, 30x30, 512], [praia, 30x30, 512], [construção, 10x10, 512], [carro, 30x30, 128], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [corrida de motos, 15x15, 256], [moto, 25x25, 64], [pôr do sol, 30x30, 512], [árvore, 5x5, 512]
234	[avião, 30x30, 512], [praia, 30x30, 512], [construção, 10x10, 512], [carro, 30x30, 128], [campo, 30x30, 512], [fogos de artifício, 30x30, 512], [flor, 30x30, 512], [corrida de motos, 15x15, 256], [moto, 25x25, 64], [montanha, 30x30, 256], [pôr do sol, 30x30, 512], [árvore, 5x5, 512]
235	[avião, 30x30, 512], [construção, 20x20, 512], [carro, 25x25, 512], [fogos de artifício, 25x25, 512], [flor, 20x20, 512], [corrida de motos, 25x25, 512], [moto, 30x30, 512], [montanha, 20x20, 512], [pôr do sol, 25x25, 512], [árvore, 15x15, 512]
236	[avião, 30x30, 512], [construção, 20x20, 512], [carro, 25x25, 512], [campo, 20x20, 256], [fogos de artifício, 25x25, 512], [flor, 20x20, 512], [corrida de motos, 25x25, 512], [moto, 30x30, 512], [montanha, 20x20, 512], [pôr do sol, 30x30, 256], [árvore, 15x15, 512]
237	[avião, 30x30, 512], [praia, 15x15, 512], [construção, 20x20, 512], [carro, 25x25, 512], [campo, 5x5, 512], [fogos de artifício, 25x25, 512], [flor, 20x20, 512], [corrida de motos, 25x25, 512], [moto, 30x30, 512], [montanha, 20x20, 512], [pôr do sol, 25x25, 512], [árvore, 15x15, 512]
238	[praia, 30x30, 256], [construção, 30x30, 256], [carro, 30x30, 512], [campo, 25x25, 256], [fogos de artifício, 25x25, 512], [flor, 30x30, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 30x30, 256], [pôr do sol, 30x30, 512], [árvore, 15x15, 512]
239	[avião, 30x30, 512], [construção, 25x25, 512], [carro, 25x25, 512], [campo, 30x30, 256], [fogos de artifício, 30x30, 512], [corrida de motos, 25x25, 512], [moto, 20x20, 512], [pôr do sol, 30x30, 256], [árvore, 30x30, 512]
240	[avião, 30x30, 512], [praia, 30x30, 256], [construção, 30x30, 512], [carro, 25x25, 512], [fogos de artifício, 30x30, 256], [flor, 30x30, 256], [corrida de motos, 25x25, 512], [moto, 25x25, 512], [pôr do sol, 30x30, 256], [árvore, 30x30, 512]
241	[avião, 30x30, 512], [praia, 30x30, 256], [construção, 30x30, 512], [carro, 25x25, 512], [campo, 20x20, 128], [fogos de artifício, 30x30, 256], [flor, 30x30, 256], [corrida de motos, 25x25, 512], [moto, 25x25, 512], [pôr do sol, 30x30, 256], [árvore, 30x30, 512]
242	[avião, 30x30, 512], [praia, 30x30, 256], [construção, 30x30, 512], [carro, 25x25, 512], [campo, 20x20, 128], [fogos de artifício, 30x30, 256], [flor, 30x30, 256], [corrida de motos, 25x25, 512], [moto, 25x25, 512], [montanha, 30x30, 512], [pôr do sol, 30x30, 256], [árvore, 30x30, 512]
243	[avião, 30x30, 256], [praia, 20x20, 256], [construção, 20x20, 512], [carro, 20x20, 512], [fogos de artifício, 25x25, 256], [flor, 25x25, 256], [moto, 25x25, 512], [montanha, 15x15, 512], [pôr do sol, 20x20, 512], [árvore, 20x20, 256]
244	[praia, 25x25, 512], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 20x20, 512], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
245	[avião, 30x30, 256], [praia, 20x20, 512], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 15x15, 512], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
246	[praia, 30x30, 512], [carro, 25x25, 128], [campo, 30x30, 512], [fogos de artifício, 30x30, 256], [flor, 25x25, 512], [corrida de motos, 20x20, 128], [moto, 30x30, 64], [pôr do sol, 30x30, 512], [árvore, 5x5, 512]
247	[avião, 30x30, 256], [praia, 30x30, 512], [carro, 25x25, 128], [campo, 30x30, 512], [fogos de artifício, 30x30, 256], [flor, 25x25, 512], [corrida de motos, 20x20, 128], [moto, 25x25, 64], [pôr do sol, 30x30, 512], [árvore, 5x5, 512]
248	[avião, 30x30, 256], [praia, 20x20, 256], [construção, 20x20, 512], [carro, 20x20, 512], [fogos de artifício, 25x25, 256], [flor, 25x25, 256], [moto, 25x25, 512], [montanha, 15x15, 512], [pôr do sol, 20x20, 512], [árvore, 20x20, 256]
249	[praia, 25x25, 512], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 20x20, 512], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]
250	[avião, 30x30, 256], [praia, 20x20, 512], [construção, 25x25, 512], [carro, 30x30, 512], [campo, 15x15, 512], [fogos de artifício, 30x30, 512], [flor, 25x25, 512], [corrida de motos, 30x30, 512], [moto, 30x30, 512], [montanha, 25x25, 512], [pôr do sol, 30x30, 512], [árvore, 20x20, 512]

Tabela A.1: Cenários dos experimentos

Tabulação dos Dados

Tabela B.1: Distorções médias da prototipação da partição A.

Janela	Dicionário	Convergência	avião	praia	construção	carro	campo	fogos de artifício	flor	corrida de moto	moto	montanha	pôr do sol	árvore	ψ ₁	ψ ₂
64		0.1	2967	2488	9848	7032	3667	5372	2236	7776	9825	5877	1678	13702	6039	5607
		0.01	2703	2366	9480	6830	3446	4940	1928	7439	9243	5530	1485	13103	5708	
		0.001	2615	2317	9072	6494	3380	4719	1857	7012	8827	5257	1438	12808	5483	
		0.0001	2545	2269	9000	6370	3336	4591	1835	6996	8777	5242	1415	12674	5421	
		0.00001	2501	2252	8920	6338	3309	4552	1829	6918	8714	5234	1424*	12627	5385	
128		0.1	2372	2133	8376	5924	3173	4554	1703	6513	8188	4976	1340	12359	5134	4777
		0.01	2168	2042	8119	5689	3064	4282	1558	6186	7725	4752	1239	11801	4875	
		0.001	2062	1926	7815	5348	2987	4033	1490	5940	7348	4525	1160	11454	4674	
		0.0001	2007	1895	7650	5288	2939	3952	1469	5842	7269	4445	1144	11367	4606	
		0.00001	1994	1892	7624	5268	2918	3926	1456	5805	7239	4433	1141	11325	4585	
256		0.1	1935	1923	7269	5104	2891	4130	1353	5613	6891	4269	1119	11270	4481	4128
		0.01	1778	1714	7030	4745	2757	3699	1263	5286	6452	4101	1018	10670	4209	
		0.001	1670	1662	6717	4490	2666	3487	1202	4999	6181	3903	959	10374	4026	
		0.0001	1638	1629	6610	4417	2624	3414	1179	4943	6109	3854	945	10259	3968	
		0.00001	1629	1625	6587	4412	2611	3401	1177	4926	6075	3841	939	10239	3955	
512		0.1	1599	1617	6355	4321	2639	3473	1105	4762	5812	3851	932	10285	3894	3585
		0.01	1458	1488	6100	4032	2479	3192	1033	4439	5441	3563	848	9667	3650	
		0.001	1377	1433	5852	3798	2393	2997	978	4267	5231	3416	799	9399	3495	
		0.0001	1356	1414	5764	3730	2358	2946	961	4210	5177	3360	785	9306	3447	
		0.00001	1349	1409	5737	3723	2346	2929	960	4206	5160	3354	784	9293	3437	
64		0.1	22814	16966	67812	53098	20875	33601	17752	58386	79603	42650	11791	86288	42636	40047
		0.01	21733	16422	66340	49937	20571	30959	16704	56105	74825	39467	11156	81889	40509	
		0.001	20598	15882	64440	48586	20114	29970	15957	54149	72634	38519	10692	80778	39360	
		0.0001	20171	15793	63655	47781	19930	29489	15695	53562	72291	37700	10583	80272	38910	
		0.00001	20173*	15602	63667*	47586	19833	29410	15645	53749*	72006	37639	10555	79920	38817	
128		0.1	19375	15397	62103	45529	19271	30843	14499	52747	68757	38441	9835	79661	38038	35586
		0.01	17901	14564	59997	43696	18713	28381	13903	49005	65272	35282	9476	76605	36066	
		0.001	17227	13753	58025	42182	18243	27280	13396	47515	63073	34048	8972	74813	34877	
		0.0001	16991	13649	57137	41583	18032	26862	13234	47179	62687	33378	8872	74317	34510	
		0.00001	16855	13606	57003	41569	17945	26778	13191	47092	62656	33551	8839	74187	34439	
256		0.1	16423	13957	57078	41536	17805	28277	12320	47700	61991	34605	8517	74271	34540	31912
		0.01	15335	12690	53950	38322	17047	25776	11759	43683	57395	31660	7983	71095	32224	
		0.001	14562	12157	52057	36927	16639	24820	11284	42067	55335	30511	7639	69781	31148	
		0.0001	14411	12026	51537	36450	16515	24394	11162	41652	55092	30124	7536	69376	30856	
		0.00001	14385	11970	51430	36347	16428	24433*	11109	41625	55016	30041	7526	69163	30789	
512		0.1	13991	11917	52181	37455	16558	25748	10453	40835	53159	31169	7288	69549	30859	28532
		0.01	12995	11043	48703	33501	15620	23237	9946	38376	49831	28382	6818	66669	28760	
		0.001	12468	10710	47030	32265	15256	22294	9559	37300	48575	27338	6540	65121	27871	
		0.0001	12300	10624	46597	31932	15157	21964	9428	36884	48324	27006	6455	64696	27614	
		0.00001	12262	10605	46510	31892	15048	21927	9423	36951*	48156	26959	6436	64535	27559	
64		0.1	70035	50054	200217	166417	58075	87340	56495	174872	244442	123523	36075	231506	124921	118005
		0.01	66283	48410	192885	150308	56091	82316	54528	170153	233320	116289	35179	223602	119114	
		0.001	64474	47502	187974	146498	54729	80473	52269	165284	227572	112609	33598	220338	116127	
		0.0001	63105	46742	186453	144752	54296	79042	51863	163850	225952	111946	33055	219194	115021	
		0.00001	63816*	46410	185939	144981*	54384*	78846	51715	163750	225778	111287	33008	218179	114841	
128		0.1	60581	45754	183084	150079	53349	81291	48032	160167	223135	112545	31107	216566	113808	103637
		0.01	57278	42742	176566	133466	50965	76861	46139	152320	206925	105143	29920	211198	107460	
		0.001	55347	41116	171077	128968	50127	73969	44941	147862	200539	101467	28300	206448	104180	
		0.0001	54432	40704	169563	128101	49575	73264	44521	145974	199857	100696	28082	204968	103312	
		0.00001	54456*	40691	169027	127529	49531	72781	44636*	146292*	199305	100089	27933	204640	103076	
256		0.1	52137	41293	168762	126409	49135	75303	41610	146402	202478	101785	26939	204069	103027	95933
		0.01	49102	37322	160228	118177	46835	70460	39840	136285	183271	94747	25053	198495	96651	
		0.001	47516	36140	155891	114438	45866	68119	38535	131756	178359	91693	24193	193869	93865	
		0.0001	46955	35937	154377	113696	45490	67435	38181	130817	177402	91122	23807	192466	93140	
		0.00001	47015*	35659	154099	113621	45347	67153	38107	131121*	176906	90646	23805	192310	92982	
512		0.1	44776	35223	154754	114649	45475	68810	35729	132383	171476	92576	22880	192002	92561	86135
		0.01	41647	32634	145586	104850	42924	63732	34061	120428	162363	85095	21172	18454	86579	
		0.001	40730	31825	141214	101368	42039	61618	33143	117323	157734	82488	20485	181316	84274	
		0.0001	40221	31577	140725	100193	41719	60881	32745	116819	156700	82025	20259	180502	83697	
		0.00001	39964	31393	140342	100063	41562	60802	32732	116719	156638	81843	20213	180516*	83566	
64		0.1	156228	106757	422153	361057	117534	174022	130379	404043	568821	262912	80502	464015	270702	253515
		0.01	148695	104383	404009	331561	114096	163763	125548	372147	517895	248108	77168	450570	254829	
		0.001	144123	101454	396163	321853	110563	159266	121541	361040	509309	242457	74380	442999	248762	
		0.0001	141663	99573	393256	319245	109954	158141	120649	360207	504654	239520	73282	440177	246624	
		0.00001	142287*	100052*	391267	318424	109850	157589	120415	360939*	504968*	239526*	74093*	439638	246587	
128		0.1	141128	97441	388808	308607	107925	161056	112969	347024	486566	240287	69109	456576	241383	228174
		0.01	129114	91885	370474	294435	103090	151408	107325	333782	44977	229753	65776	424440	229856	
		0.001	123801	88012	362355	286290	100932	147580	105314	327274	452975	217675	62276	416586	224293	
		0.0001	123559	87171	358602	284218	99833	146554	103978	325469	451031	216100	61855	414676	222766	
		0.00001	123508	87322*	358499	284306*	99753	146398	104346*	324906*	449556*	216224*	62455*	415849	222571	
256		0.1	119557	84031	358286	281236	99601	148641	97672	317662	443931	217240	59256	410227	219778	206303
		0.01	110931	79526	338143	263119	94012	139875	94125	301141	411096	202673	55			

Tabela B.2: Distorções médias da prototipação da partição B.

Janela	Dicionário	Convergência	avião	praia	construção	carro	campo	fogos de artifício	flor	corrida de moto	moto	montanha	pôr do sol	árvore	ψ ₁	ψ ₂
64		0.1	2988	2659	9781	6950	3853	4925	2241	8369	9917	6244	1606	13704	6103	5626
		0.01	2630	2568	9080	6610	3736	4519	1987	7747	9233	5770	1408	13247	5711	
		0.001	2522	2462	8698	6378	3602	4303	1903	7382	8811	5555	1389	12908	5493	
		0.0001	2519	2448	8508	6301	3558	4217	1856	7347	8685	5491	1346	12771	5421	
		0.00001	2500	2428	8493	6272	3554	4179	1850	7309	8667	5453	1358*	12758	5402	
128	5x5	0.1	2354	2297	8063	5892	3494	4217	1701	6889	8141	5177	1248	12416	5158	4790
		0.01	2107	2209	7793	5648	3299	3932	1562	6550	7665	4935	1175	11922	4900	
		0.001	2010	2090	7380	5371	3199	3713	1504	6215	7272	4735	1103	11599	4683	
		0.0001	1990	2063	7230	5258	3162	3633	1470	6143	7181	4638	1083	11459	4609	
		0.00001	1974	2060	7236*	5252	3152	3597	1461	6139	7181*	4625	1078	11430	4599	
256		0.1	1932	1977	6891	5099	3170	3815	1361	5892	6811	4457	1029	11347	4482	4128
		0.01	1722	1865	6674	4713	2969	3379	1250	5587	6379	4250	962	10755	4209	
		0.001	1621	1806	6337	4464	2887	3186	1203	5281	6095	4071	911	10442	4025	
		0.0001	1591	1783	6212	4394	2832	3127	1182	5214	6022	4015	892	10341	3967	
		0.00001	1591*	1778	6202	4389	2829	3096	1171	5194	6017	4001	894*	10316	3956	
512		0.1	1573	1781	5978	4336	2884	3195	1104	5019	5746	4015	862	10326	3902	3571
		0.01	1392	1614	5735	3952	2674	2898	1035	4672	5371	3691	788	9733	3627	
		0.001	1317	1559	5463	3742	2583	2730	963	4471	5119	3537	750	9448	3474	
		0.0001	1303	1542	5373	3698	2541	2683	949	4427	5071	3493	736	9354	3431	
		0.00001	1298	1535	5372	3680	2533	2663	942	4415	5069	3476	733	9335	3421	
64		0.1	22873	17784	65004	51701	22294	30186	17763	61725	79492	44240	11231	87153	42621	39921
		0.01	21106	17370	63205	48079	21311	28060	16721	58537	75028	41022	10611	82047	40258	
		0.001	20163	16893	61099	47642	21007	27270	15954	56790	72459	39845	10209	81075	39200	
		0.0001	19937	16754	60900	46711	20938	26952	15749	56458	71816	39525	10104	80347	38849	
		0.00001	19852	16615	60368	46544	20821	26824	15683	56232	71574	39353	10019	80228	38676	
128	10x10	0.1	19146	16138	59159	44526	20434	27879	14515	53469	67674	39899	9350	79829	37668	35335
		0.01	17410	15417	56873	43074	19546	25645	13828	51841	64126	36467	8781	76138	35762	
		0.001	16718	14635	54785	40993	19191	24989	13304	49809	62460	35485	8462	75117	34662	
		0.0001	16716	14605	54107	40784	19058	24500	13109	49503	62093	34912	8424	74733	34348	
		0.00001	16554	14509	53870	40684	18985	24333	13031	49382	62118*	34754	8330	74261	34234	
256		0.1	16150	14696	54034	40619	18907	25460	12128	48828	60944	34378	7927	74090	34013	31496
		0.01	14748	13204	50971	37368	17985	23266	11500	45321	55578	32658	7429	71125	31763	
		0.001	14207	12868	48946	35729	17520	22478	11089	43982	54237	31448	7133	69603	30770	
		0.0001	14002	12774	48501	35657	17397	22159	10958	43622	53847	31079	7079	69895	30505	
		0.00001	13992	12687	48397	35337	17377	21916	10895	43465	53910*	31101*	7033	69025*	30428	
512		0.1	13356	12498	48842	34690	17560	23004	10207	42092	51580	31234	6748	68975	30066	27847
		0.01	12162	11530	45440	32250	16372	20668	9546	39593	48098	28692	6250	65608	28017	
		0.001	11836	11252	43690	30844	16038	19886	9227	38458	46889	27892	6000	64418	27203	
		0.0001	11625	11163	43382	30833	15947	19627	9128	38257	46644	27568	5950	63922	27004	
		0.00001	11665*	11108	43180	30683	15884	19422	9113	38130	46825*	27545	5925	63894	26948	
64		0.1	70864	51962	191024	159209	60073	78057	56370	190585	242636	128751	33948	230771	124521	117258
		0.01	66075	50356	183233	143821	57250	75530	53826	177801	235142	122035	32655	223860	118297	
		0.001	63430	49639	179613	140867	56218	71990	52128	173152	225254	117827	31615	220578	115168	
		0.0001	62576	48746	178338	139309	55733	71368	51931	172130	224488	116862	31493	218602	114298	
		0.00001	62354	48696	177293	139133	55701	71280	51892	171519	224274	116625	31113	218187	114006	
128	15x15	0.1	59462	47424	174582	135024	55027	72028	48024	164495	218965	117023	29065	215983	111425	104738
		0.01	55562	44639	165033	128791	52364	68326	45783	157224	202679	109225	27262	207900	105399	
		0.001	54133	42984	161350	124399	51402	66537	44646	153897	198107	105683	26325	205245	102892	
		0.0001	53117	42478	160114	123390	51182	65625	44145	152911	196440	104784	26124	204591	102075	
		0.00001	53155*	42241	159838	123577*	50957	65778*	44029	152983*	196295	104153	26062	203687	101896	
256		0.1	49955	40995	159537	122054	50705	66177	41083	150298	186841	101177	24897	202347	99672	93507
		0.01	47082	38496	148608	112914	48397	62094	39113	140421	177555	97452	22974	194980	94174	
		0.001	45406	37486	145110	109455	47180	60582	37582	136802	172739	94142	22159	191699	91695	
		0.0001	45106	37142	144377	108821	46737	59660	37272	135625	171689	93998	21857	190124	91034	
		0.00001	44721	37121	144319	108706	46778*	59598	37324*	135509	171665	93496	21983*	190329*	90963	
512		0.1	41068	35135	143831	104363	46479	57158	34353	129868	160522	91874	20498	187964	87789	82447
		0.01	38344	33229	132997	98021	43634	54898	32243	121792	153984	86333	18895	180984	82946	
		0.001	37283	32153	129728	94876	42784	53115	31239	120118	150024	83687	18179	177707	80908	
		0.0001	37053	32050	129091	94655	42357	51853	30960	118898	149669	82818	17930	176914	80354	
		0.00001	36753	31972	129408*	94174	42214	51657	30813	118866	149811*	82802	17829	176531	80236	
64		0.1	154565	110412	396533	341427	119568	151132	128746	418118	542799	272010	75921	462138	264447	249619
		0.01	145199	108664	383931	316913	113671	143828	123311	386522	512888	254774	72559	447581	250820	
		0.001	140308	105547	375661	309375	111778	141357	120427	375644	503094	251633	69824	443727	245698	
		0.0001	137972	104433	372151	307797	111332	140124	118390	375477	499858	249770	69501	438966	243814	
		0.00001	137593	104122	372757*	305866	110654	140130*	119075*	372763	500902*	249041	68779	438110	243316	
128	20x20	0.1	129403	100419	364639	309598	109252	140077	110511	359460	489368	247213	65298	434361	238300	223755
		0.01	123306	94660	348636	280365	103420	133465	105394	344590	453433	229628	60166	420474	224795	
		0.001	119782	91925	339694	272267	102006	130578	102598	335316	443579	225601	57913	413957	219585	
		0.0001	117625	91600	337557	271024	100921	128748	101506	335529*	440002	222127	57108	411597	217853	
		0.00001	117695*	91350	336170	270604	101715*	128818*	101873*	334013	443402*	224230*	57188*	410636	218142*	
256		0.1	109372	85647	332253	260617	99612	136039	92829	325121	418574	213242	54434	405579	210352	198541
		0.01	102408	80555	312611	247349	93940	120133	88142	306574	3943					

Tabela B.3: Distorções médias da prototipação da partição C.

Janela	Dicionário	Convergência	avião	praia	construção	carro	campo	fogos de artifício	flor	corrida de moto	moto	montanha	pôr do sol	árvore	ψ ₁	ψ ₂
64	0.1	2666	2772	9544	7248	3665	5061	2162	8626	9205	5846	1587	13349	5978	5516	
	0.01	2443	2690	8926	6376	3452	4669	1945	8118	8587	5487	1418	13070	5598		
	0.001	2280	2565	8475	6081	3422	4377	1890	7860	8268	5198	1372	12823	5384		
	0.0001	2297*	2545	8360	6019	3379	4293	1857	7743	8174	5156	1353	12688	5322		
	0.00001	2291	2544	8300	6021*	3365	4257	1831	7704	8136	5131	1340	12629	5296		
128	0.1	2123	2389	7874	5859	3309	4272	1659	7327	7577	4895	1247	12208	5062	4686	
	0.01	1886	2266	7618	5461	3107	3983	1530	6923	7083	4668	1128	11762	4785		
	0.001	1819	2178	7230	5062	3011	3713	1477	6590	6792	4448	1077	11472	4572		
	0.0001	1790	2157	7122	5012	2974	3657	1451	6467	6716	4398	1067	11349	4513		
	0.00001	1784	2148	7071	4988	2964	3634	1444	6487*	6681	4383	1056	11308	4496		
5x5	0.1	1701	2056	6779	4881	2985	3648	1331	6271	6264	4227	1027	11173	4362	4010	
	0.01	1512	1938	6518	4404	2772	3356	1235	5833	5834	4004	931	10560	4075		
	0.001	1433	1870	6158	4216	2691	3155	1178	5544	5613	3828	884	10303	3906		
	0.0001	1418	1853	6086	4179	2663	3103	1161	5464*	5553	3790	864	10209	3861		
	0.00001	1408	1845	6037	4164	2647	3094	1154	5464*	5552	3778	863	10166	3848		
256	0.1	1368	1780	5831	4088*	2695	3098	1071	5326	5224	3809	837	10161	3774	3442	
	0.01	1130	1660	5570	3683	2457	2813	977	4825	4874	3448	758	9513	3491		
	0.001	1144	1596	5268	3507	2391	2666	938	4686	4647	3320	717	9259	3345		
	0.0001	1127	1577	5199	3452	2368	2626	923	4616	4604	3288	709	9174	3305		
	0.00001	1119	1580*	5182	3444	2355	2599	918	4607	4602	3276	707	9148	3295		
64	0.1	21376	18495	65183	49096	20764	31380	17508	65584	74622	40829	11372	86547	41896	39127	
	0.01	19643	18257	63043	45354	20344	29351	16561	61878	69362	37871	10575	81726	39497		
	0.001	18631	17408	60855	43875	19939	28397	16019	59905	67611	36765	10155	80714	38356		
	0.0001	18293	17445*	60093	43601	19796	28025	15732	59143	67319	36474	9979	79879	37982		
	0.00001	18347*	17351	60248*	43291	19772	27950	15770*	59144*	66801	36279	10089*	79794	37903		
128	0.1	17385	16866	59228	41610	19268	28809	14546	55972	63225	36771	9460	79388	36877	34438	
	0.01	15777	15742	56100	39873	18408	26602	13577	53863	59233	33394	8768	75574	34742		
	0.001	15223	15170	54064	38233	17974	25515	13266	52166	57862	32659	8285	74257	33723		
	0.0001	15102	15174*	53437	37901	17836	25129	13105	52091	57466	32417	8225	73853	33478		
	0.00001	15133*	15076	53295	37624	17801	25145*	13098	51541	57495*	32328	8246*	73639	33368		
10x10	0.1	14164	14474	53668	37506	17670	26129	12096	50776	54403	31319	7875	73605	32807	30351	
	0.01	13083	13657	49843	33800	16451	23360	11285	46959	51204	29808	7172	69751	30531		
	0.001	12576	13214	47710	32793	16266	22680	10919	45556	49800	28861	6842	68302	29626		
	0.0001	12458	13093	47404	32516	16125	22360	10820	45193	49508	28591	6799	68065	29411		
	0.00001	12447	13094*	47343	32505	16102	22284	10805	45216*	49304	28608*	6780	68089*	29382		
512	0.1	11488	12421	45671	31502	16113	21959	9961	43252	46322	28247	6428	67958	28444	26384	
	0.01	10392	11619	43546	28830	14787	20192	9170	40685	43395	25907	5861	63929	26526		
	0.001	10108	11320	41915	27768	14565	19518	8910	39363	42128	25240	5592	62943	25781		
	0.0001	9966	11220	41606	27600	14487	19120	8803	39200	42121	25082	5512	62533	25604		
	0.00001	9901	11212	41688*	27465	14474	19132*	8735	39243*	41990	25065	5536*	62361	25567		
64	0.1	65762	54287	192952	141536	56759	81585	56577	196312	230691	120380	35144	231188	121931	114470	
	0.01	59598	52191	183513	133101	54873	76133	54127	185638	216555	111944	33008	224210	115408		
	0.001	57981	50997	179056	128814	53472	74739	52392	179340	210092	107506	31670	220726	112322		
	0.0001	56823	50053	177684	128327	53181	73947	52399*	178980	210698*	107024	31641	21772	111544		
	0.00001	57132*	50094*	177512	127970	53256*	73906	51926	178095	209087	106487	31496	21701	111237		
128	0.1	54193	49218	173711	121823	51461	74220	48054	169666	197766	109406	29376	215200	107841	101278	
	0.01	50146	45050	163990	116112	49572	70188	45295	162695	186968	98110	27043	207149	101860		
	0.001	48333	43785	159181	112767	48374	68007	44254	159120	181588	95965	26109	203464	99245		
	0.0001	48028	43678	157932	112768*	47968	67037	43691	158310	180890	95067	25893	202402	98639		
	0.00001	47867	43744*	159070*	112198*	48007*	67609*	43639	158125	181974*	94701	26005*	202695*	98803*		
256	0.1	43402	41711	157108	109824	46957	64897	40392	154843	169588	91863	23941	200356	95407	89239	
	0.01	40829	38518	146146	100551	44468	62963	37785	143327	161060	86829	22091	190113	89557		
	0.001	40031	37500	141629	98466	43095	60703	36887	140274	156853	84732	21159	188167	87458		
	0.0001	38991	37503*	140487	97266	42977	59368	36499	139389	156549	84191	21109	187228	86796		
	0.00001	39235*	37367	140914*	98084*	43283*	59999*	36260	139385	156880*	83883	21411*	187019	86977*		
512	0.1	33411	33982	132662	91452	40418	54557	32638	129559	142296	78289	19022	182885	80931	76216	
	0.01	31151	32256	127970	83498	38716	52049	30367	122578	136007	75248	17359	174219	76785		
	0.001	30194	30690	123816	82675	37871	49366	29776	119642	132319	72601	16426	171995	74781		
	0.0001	29235	30931*	123034	81380	37822	47930	29515	119371	132015	72246	16520*	170897	74242		
	0.00001	29961*	30839	123545*	81760*	37677	48014*	28980	118877	132771*	72407*	16549*	170717	74342*		
64	0.1	145133	115410	403444	302069	114756	159976	131149	420769	512314	248471	78732	465428	258138	242801	
	0.01	134678	109616	387856	286381	107521	149651	124836	396694	476843	230522	72257	446347	243600		
	0.001	132213	107034	374817	280786	106817	146767	121467	389057	467410	224457	70283	441103	238518		
	0.0001	128332	106616	372244	279854	106050	146846*	120224	387700	466751	225956*	69908	436517	237250		
	0.00001	128150	106621*	372199	278321	105398	145351	119950	385069	466180	224494	69854	436435	236502		
128	0.1	117971	100195	365388	261728	99499	144459	109600	379403	439425	224217	64813	432662	228280	214641	
	0.01	109852	94433	345009	250717	97044	136442	103894	351062	412217	205443	58521	415506	215012		
	0.001	109442	92707	335187	245797	94662	134100	102580	345841	403913	199830	56331	409393	210757		
	0.0001	108184	92450*	332460	246019*	94779*	133553	101204	343591	404765*	198992	55510	407454	209807		
	0.00001	107501	91953	331355	244087	94511	130839	101067	341250	405258*	199177*	55682*	408105*	209249		
20x20	0.1	93688	85279	314726	226113	90679	121450	108984	320709	376503	189645	51053	397415	196388	186585	
	0.01	88743	78546	305492	214628	85299	116982	87209	309854	355415	178013	46159	382085			

Tabela B.4: *Distâncias médias absolutas e relativas dos centroides dos codebooks entre as partições de treinamento com $\epsilon = 0, 1$.*

Janela	Dicionário	A e B		B e C		A e C	
		Absoluta	Relativa	Absoluta	Relativa	Absoluta	Relativa
5x5	64	9,2790	0,73%	11,6834	0,92%	16,5037	1,29%
	128	9,7616	0,77%	11,2826	0,88%	16,5909	1,30%
	256	9,5660	0,75%	10,9554	0,86%	16,4950	1,29%
	512	9,4602	0,74%	10,8799	0,85%	16,3782	1,28%
10x10	64	18,4898	0,73%	27,8193	1,09%	35,0172	1,37%
	128	18,4353	0,72%	27,2695	1,07%	34,7670	1,36%
	256	18,2254	0,71%	27,4811	1,08%	35,1388	1,38%
	512	17,7261	0,70%	27,0250	1,06%	35,2626	1,38%
15x15	64	30,5979	0,80%	43,3979	1,13%	56,0640	1,47%
	128	30,8659	0,81%	43,4853	1,14%	57,0846	1,49%
	256	30,9438	0,81%	43,1914	1,13%	56,9950	1,49%
	512	30,1232	0,79%	43,2681	1,13%	56,1151	1,47%
20x20	64	43,3357	0,85%	63,6687	1,25%	81,5576	1,60%
	128	45,2945	0,89%	63,0590	1,24%	82,8425	1,62%
	256	45,8308	0,90%	62,8721	1,23%	82,3154	1,61%
	512	45,9056	0,90%	61,3167	1,20%	82,0755	1,61%
25x25	64	61,4848	0,96%	83,4760	1,31%	101,0860	1,59%
	128	60,7574	0,95%	83,9237	1,32%	102,5012	1,61%
	256	61,3787	0,96%	84,4818	1,33%	102,8590	1,61%
	512	64,7000	1,01%	83,0127	1,30%	102,8536	1,61%
30x30	64	94,0923	1,23%	119,0955	1,56%	128,0273	1,67%
	128	95,0826	1,24%	115,6667	1,51%	128,0627	1,67%
	256	96,7227	1,26%	114,9915	1,50%	128,4427	1,68%
	512	103,0010	1,35%	114,7753	1,50%	128,8916	1,68%

Tabela B.5: *Distâncias médias absolutas e relativas dos centroides dos codebooks entre as partições de treinamento com $\epsilon = 0,01$.*

Janela	Dicionário	A e B		B e C		A e C	
		Absoluta	Relativa	Absoluta	Relativa	Absoluta	Relativa
5x5	64	10,0651	0,79%	10,8085	0,85%	16,0966	1,26%
	128	9,5397	0,75%	9,9001	0,78%	15,4378	1,21%
	256	9,1908	0,72%	10,1724	0,80%	15,3987	1,21%
	512	9,0895	0,71%	9,9637	0,78%	15,2450	1,20%
10x10	64	23,3242	0,91%	28,5926	1,12%	35,8404	1,41%
	128	21,9721	0,86%	27,5129	1,08%	34,0230	1,33%
	256	21,3849	0,84%	25,6539	1,01%	33,7658	1,32%
	512	21,7313	0,85%	25,1367	0,99%	33,7215	1,32%
15x15	64	31,2820	0,82%	45,6651	1,19%	55,3875	1,45%
	128	31,7349	0,83%	45,7303	1,20%	56,0335	1,46%
	256	31,8126	0,83%	45,4544	1,19%	56,9597	1,49%
	512	31,3967	0,82%	45,5965	1,19%	57,0367	1,49%
20x20	64	44,2697	0,87%	66,8923	1,31%	78,4272	1,54%
	128	45,3102	0,89%	64,2536	1,26%	78,9860	1,55%
	256	47,1523	0,92%	64,9333	1,27%	78,6760	1,54%
	512	48,6918	0,95%	65,8258	1,29%	80,4745	1,58%
25x25	64	59,7323	0,94%	93,0328	1,46%	105,9539	1,66%
	128	61,0183	0,96%	92,1348	1,45%	101,0820	1,59%
	256	62,5377	0,98%	94,4502	1,48%	101,9850	1,60%
	512	64,2999	1,01%	94,4290	1,48%	102,4807	1,61%
30x30	64	79,4103	1,04%	121,9409	1,59%	147,1067	1,92%
	128	80,3199	1,05%	123,3702	1,61%	150,2668	1,96%
	256	81,1817	1,06%	123,9346	1,62%	152,3473	1,99%
	512	82,0182	1,07%	123,1537	1,61%	151,6448	1,98%

Tabela B.6: Distâncias médias absolutas e relativas dos centroides dos codebooks entre as partições de treinamento com $\epsilon = 0,001$.

Janela	Dicionário	A e B		B e C		A e C	
		Absoluta	Relativa	Absoluta	Relativa	Absoluta	Relativa
5x5	64	9,9465	0,78%	12,8756	1,01%	18,2006	1,43%
	128	9,6671	0,76%	12,7598	1,00%	17,4341	1,37%
	256	8,7659	0,69%	12,9607	1,02%	16,6857	1,31%
	512	8,2800	0,65%	12,5376	0,98%	16,2666	1,28%
10x10	64	23,0927	0,91%	26,6148	1,04%	29,4569	1,16%
	128	23,3342	0,92%	26,9038	1,06%	28,2562	1,11%
	256	22,7894	0,89%	27,7702	1,09%	29,1654	1,14%
	512	22,3057	0,87%	27,4761	1,08%	29,3089	1,15%
15x15	64	32,5057	0,85%	40,7534	1,07%	52,8117	1,38%
	128	32,1067	0,84%	39,0230	1,02%	52,1960	1,36%
	256	31,9426	0,84%	38,3791	1,00%	52,0079	1,36%
	512	32,6556	0,85%	38,1323	1,00%	52,8498	1,38%
20x20	64	51,0680	1,00%	70,4131	1,38%	75,5889	1,48%
	128	53,9411	1,06%	71,1707	1,40%	73,2439	1,44%
	256	52,5296	1,03%	71,3861	1,40%	74,6260	1,46%
	512	54,3909	1,07%	72,5121	1,42%	77,0088	1,51%
25x25	64	73,7900	1,16%	77,1771	1,21%	89,7525	1,41%
	128	73,3074	1,15%	79,4702	1,25%	88,5559	1,39%
	256	72,9061	1,14%	79,2646	1,24%	90,1043	1,41%
	512	73,5235	1,15%	80,0463	1,26%	90,8504	1,43%
30x30	64	83,3041	1,09%	116,4404	1,52%	134,1038	1,75%
	128	88,0262	1,15%	117,6513	1,54%	136,8763	1,79%
	256	90,0194	1,18%	120,8559	1,58%	140,8221	1,84%
	512	92,0533	1,20%	121,8137	1,59%	142,7792	1,87%

Tabela B.7: Distâncias médias absolutas e relativas dos centroides dos codebooks entre as partições de treinamento com $\epsilon = 0,0001$.

Janela	Dicionário	A e B		B e C		A e C	
		Absoluta	Relativa	Absoluta	Relativa	Absoluta	Relativa
5x5	64	9,0172	0,71%	12,0762	0,95%	16,0235	1,26%
	128	8,1881	0,64%	13,3349	1,05%	16,6336	1,30%
	256	7,6363	0,60%	13,2684	1,04%	16,3709	1,28%
	512	7,7949	0,61%	12,9385	1,01%	16,4770	1,29%
10x10	64	19,1954	0,75%	28,9914	1,14%	27,8848	1,09%
	128	19,8874	0,78%	27,1843	1,07%	28,2216	1,11%
	256	19,8973	0,78%	26,4982	1,04%	27,9575	1,10%
	512	20,2442	0,79%	26,8927	1,05%	28,6514	1,12%
15x15	64	43,6665	1,14%	48,6230	1,27%	58,2919	1,52%
	128	36,0664	0,94%	47,7486	1,25%	55,4583	1,45%
	256	37,6833	0,99%	48,9517	1,28%	55,5979	1,45%
	512	37,4678	0,98%	48,5787	1,27%	55,4541	1,45%
20x20	64	38,1693	0,75%	72,2670	1,42%	72,8446	1,43%
	128	37,9919	0,74%	68,5747	1,34%	72,6602	1,42%
	256	38,6012	0,76%	69,8225	1,37%	73,0275	1,43%
	512	38,8146	0,76%	70,6172	1,38%	73,7819	1,45%
25x25	64	74,5306	1,17%	96,5395	1,51%	101,8510	1,60%
	128	72,5686	1,14%	95,2473	1,49%	103,0736	1,62%
	256	74,5715	1,17%	95,4756	1,50%	103,3666	1,62%
	512	75,6994	1,19%	96,1625	1,51%	104,0579	1,63%
30x30	64	98,8988	1,29%	152,4798	1,99%	163,3091	2,13%
	128	97,8013	1,28%	147,8415	1,93%	162,6628	2,13%
	256	99,4942	1,30%	150,2864	1,96%	162,5448	2,12%
	512	99,7988	1,30%	151,7322	1,98%	163,0026	2,13%

Tabela B.8: *Distâncias médias absolutas e relativas dos centroides dos codebooks entre as partições de treinamento com $\epsilon = 0,00001$.*

Janela	Dicionário	A e B		B e C		A e C	
		Absoluta	Relativa	Absoluta	Relativa	Absoluta	Relativa
5x5	64	9,8608	0,77%	14,6845	1,15%	16,7442	1,31%
	128	9,0903	0,71%	13,5988	1,07%	15,7791	1,24%
	256	7,7326	0,61%	13,4002	1,05%	15,3699	1,21%
	512	7,8277	0,61%	12,9873	1,02%	15,9566	1,25%
10x10	64	22,2440	0,87%	20,2866	0,80%	32,6067	1,28%
	128	22,5237	0,88%	19,2240	0,75%	31,6560	1,24%
	256	21,9834	0,86%	18,3317	0,72%	31,3105	1,23%
	512	21,1661	0,83%	18,3381	0,72%	30,6423	1,20%
15x15	64	30,0739	0,79%	50,2760	1,31%	60,1542	1,57%
	128	31,0379	0,81%	51,5686	1,35%	62,1513	1,62%
	256	31,9245	0,83%	50,9919	1,33%	61,6295	1,61%
	512	32,3696	0,85%	51,1281	1,34%	60,5941	1,58%
20x20	64	47,2855	0,93%	70,5163	1,38%	82,0275	1,61%
	128	46,8038	0,92%	70,8182	1,39%	80,2869	1,57%
	256	47,0734	0,92%	70,0437	1,37%	80,3116	1,57%
	512	47,3204	0,93%	69,9940	1,37%	81,4026	1,60%
25x25	64	61,2649	0,96%	80,0045	1,25%	86,7993	1,36%
	128	62,3705	0,98%	81,1266	1,27%	85,9018	1,35%
	256	60,7005	0,95%	81,8486	1,28%	86,3906	1,36%
	512	61,7274	0,97%	82,2333	1,29%	87,7321	1,38%
30x30	64	106,6783	1,39%	130,6905	1,71%	161,7932	2,11%
	128	106,1528	1,39%	132,0238	1,73%	162,0217	2,12%
	256	109,4984	1,43%	131,5661	1,72%	163,1089	2,13%
	512	109,4867	1,43%	131,7209	1,72%	164,3749	2,15%

Tabela B.9: *Medidas objetivas utilizadas para avaliar a qualidade média das imagens quantizadas com os dicionários convergentes e divergentes com $\epsilon = 0, 1$.*

Partição	Janela	Dicionário	Quantização Convergente				Quantização Divergente			
			MSE	SSIM	PSNR	VIF	MSE	SSIM	PSNR	VIF
A	5x5	64	40,5123	0,7630	25,6567	0,2859	48,9883	0,7005	24,8817	0,2526
		128	37,3973	0,7957	26,4196	0,3214	43,1221	0,7426	25,6304	0,2850
		256	35,1320	0,8215	27,0766	0,3523	39,5712	0,7749	26,2362	0,3140
		512	33,1567	0,8439	27,7519	0,3826	37,0292	0,8029	26,8406	0,3417
	10x10	64	51,0505	0,6354	23,1574	0,1613	60,3373	0,5670	22,3592	0,1360
		128	47,9454	0,6649	23,7039	0,1861	54,9675	0,5987	22,7290	0,1565
		256	45,6806	0,6904	24,1871	0,2093	51,2104	0,6294	23,3347	0,1759
		512	43,7130	0,7159	24,7181	0,2342	48,8264	0,6540	23,5737	0,1946
	15x15	64	57,3700	0,5813	21,9433	0,1143	68,1577	0,5172	21,2669	0,0942
		128	54,3084	0,6046	22,4078	0,1319	61,7735	0,5379	21,4321	0,1070
		256	51,8482	0,6277	22,8775	0,1499	58,3871	0,5581	21,7839	0,1193
		512	49,8614	0,6511	23,3744	0,1701	56,3743	0,5714	21,9985	0,1310
	20x20	64	62,0753	0,5514	21,1131	0,0913	73,6804	0,4811	20,1462	0,0742
		128	58,8652	0,5735	21,6027	0,1057	68,3548	0,4992	20,6123	0,0836
		256	56,4006	0,5933	22,0663	0,1203	65,1058	0,5065	20,6543	0,0912
		512	54,3145	0,6170	22,5938	0,1382	62,6170	0,5158	20,9851	0,0978
	25x25	64	65,3840	0,5354	20,5425	0,0788	77,8671	0,4627	19,5651	0,0640
		128	62,3375	0,5531	21,0113	0,0907	72,6625	0,4747	19,9308	0,0694
		256	59,9637	0,5722	21,4940	0,1040	69,3670	0,4753	20,0009	0,0739
		512	57,6704	0,5993	22,0800	0,1222	67,6479	0,4813	20,2085	0,0782
	30x30	64	68,0615	0,5237	20,0851	0,0699	80,6395	0,4503	19,1514	0,0560
		128	65,3078	0,5385	20,5489	0,0801	76,2910	0,4552	19,5408	0,0599
		256	62,6315	0,5602	21,0821	0,0945	73,3480	0,4518	19,5362	0,0631
		512	60,1426	0,5909	21,7244	0,1144	71,3729	0,4543	19,6225	0,0656
B	5x5	64	40,9606	0,7604	25,6021	0,2843	49,7612	0,6951	24,6470	0,2511
		128	37,7453	0,7947	26,3838	0,3205	43,4619	0,7420	25,5791	0,2844
		256	35,4291	0,8211	27,0623	0,3520	39,6706	0,7768	26,2278	0,3132
		512	33,4631	0,8433	27,7186	0,3819	37,2783	0,7996	26,6101	0,3389
	10x10	64	51,2309	0,6348	23,1410	0,1609	61,2281	0,5596	22,2407	0,1346
		128	48,0961	0,6656	23,7166	0,1866	55,9670	0,5908	22,4306	0,1542
		256	45,8687	0,6916	24,2235	0,2103	52,0408	0,6202	22,9037	0,1725
		512	43,9281	0,7184	24,7902	0,2354	49,8097	0,6439	23,3006	0,1892
	15x15	64	57,4394	0,5796	21,9444	0,1138	68,3348	0,5069	20,9600	0,0927
		128	54,3371	0,6057	22,4622	0,1326	63,0982	0,5293	21,4859	0,1049
		256	51,9521	0,6311	22,9765	0,1526	60,5074	0,5443	21,5200	0,1158
		512	49,8052	0,6593	23,5622	0,1751	58,0393	0,5576	21,7639	0,1254
	20x20	64	62,2046	0,5521	21,1716	0,0917	74,0065	0,4733	20,1049	0,0722
		128	59,0179	0,5736	21,6611	0,1057	68,8745	0,4880	20,4810	0,0795
		256	56,4586	0,5997	22,2216	0,1240	65,7388	0,4939	20,5241	0,0860
		512	53,8673	0,6336	22,9234	0,1480	64,1462	0,5039	20,7361	0,0916
	25x25	64	65,4591	0,5353	20,6068	0,0790	77,9534	0,4517	19,5322	0,0614
		128	62,3727	0,5571	21,1389	0,0931	73,9027	0,4517	19,6072	0,0660
		256	59,7818	0,5838	21,7196	0,1103	70,9864	0,4593	19,7993	0,0695
		512	56,5371	0,6269	22,5766	0,1396	69,3063	0,4641	20,0197	0,0724
	30x30	64	67,9641	0,5263	20,2097	0,0724	80,7165	0,4350	19,0534	0,0537
		128	65,0114	0,5444	20,7173	0,0840	77,0614	0,4296	19,0583	0,0561
		256	62,0643	0,5778	21,4049	0,1048	74,4810	0,4350	19,2994	0,0584
		512	57,9115	0,6299	22,3994	0,1453	73,3052	0,4392	19,3388	0,0603
C	5x5	64	40,7491	0,7633	25,6984	0,2860	51,0229	0,6922	24,7877	0,2498
		128	37,6710	0,7963	26,4747	0,3219	45,3380	0,7235	24,8812	0,2790
		256	35,2772	0,8236	27,1845	0,3546	41,2044	0,7581	25,5125	0,3071
		512	33,2764	0,8463	27,8838	0,3856	38,1510	0,7827	25,9256	0,3327
	10x10	64	51,2476	0,6360	23,2332	0,1613	63,3065	0,5538	22,3811	0,1327
		128	48,1034	0,6676	23,8205	0,1874	57,9524	0,5756	22,3838	0,1511
		256	45,7343	0,6961	24,3952	0,2135	54,0318	0,6053	22,9341	0,1692
		512	43,5796	0,7261	25,0449	0,2413	51,4705	0,6243	23,0260	0,1840
	15x15	64	57,3649	0,5819	22,0382	0,1147	70,7330	0,4889	20,7797	0,0899
		128	54,0661	0,6105	22,6022	0,1354	65,8703	0,5105	21,2193	0,1013
		256	51,6342	0,6381	23,1905	0,1568	62,5775	0,5212	21,2060	0,1110
		512	49,1404	0,6740	23,9453	0,1847	60,4317	0,5337	21,4501	0,1190
	20x20	64	61,6022	0,5547	21,2686	0,0916	76,2083	0,4516	19,8727	0,0693
		128	58,6855	0,5796	21,8479	0,1092	71,7093	0,4634	20,2074	0,0764
		256	55,8298	0,6126	22,5205	0,1313	68,7047	0,4728	20,3988	0,0814
		512	52,3360	0,6593	23,4765	0,1663	67,4716	0,4767	20,3434	0,0857
	25x25	64	65,1875	0,5400	20,7556	0,0813	79,8765	0,4289	19,3129	0,0586
		128	61,6252	0,5682	21,3953	0,0978	75,8674	0,4332	19,4739	0,0618
		256	58,5698	0,6056	22,1904	0,1231	73,7988	0,4337	19,4764	0,0647
		512	53,4052	0,6653	23,4276	0,1763	72,3079	0,4429	19,6310	0,0673
	30x30	64	67,5132	0,5282	20,3167	0,0737	81,9095	0,4126	18,8991	0,0499
		128	64,1594	0,5582	21,0061	0,0911	79,5578	0,4146	19,0948	0,0524
		256	60,3213	0,6062	21,9124	0,1237	77,0201	0,4092	18,8094	0,0543
		512	52,7391	0,6810	23,4492	0,2025	75,9299	0,4201	19,1060	0,0565

Tabela B.10: *Medidas objetivas utilizadas para avaliar a qualidade média das imagens quantizadas com os dicionários convergentes e divergentes com $\epsilon = 0,01$.*

Partição	Janela	Dicionário	Quantização Convergente				Quantização Divergente			
			MSE	SSIM	PSNR	VIF	MSE	SSIM	PSNR	VIF
A	5x5	64	40,2625	0,7656	25,8691	0,2902	48,6514	0,7012	25,0507	0,2573
		128	37,3430	0,7989	26,6017	0,3249	43,2543	0,7436	25,7664	0,2883
		256	34,9772	0,8258	27,3195	0,3578	39,5365	0,7795	26,4481	0,3182
		512	32,9852	0,8479	28,0014	0,3885	36,9241	0,8064	27,0577	0,3457
	10x10	64	50,6017	0,6381	23,3138	0,1650	60,4931	0,5700	22,6246	0,1391
		128	47,7498	0,6704	23,8776	0,1908	55,6616	0,6030	23,0985	0,1589
		256	45,3592	0,6982	24,4228	0,2159	51,9045	0,6339	23,5805	0,1786
		512	43,3644	0,7234	24,9685	0,2415	49,2211	0,6595	24,0003	0,1973
	15x15	64	56,9124	0,5855	22,0866	0,1175	68,4287	0,5193	21,3469	0,0961
		128	53,7942	0,6112	22,5897	0,1363	63,1507	0,5427	21,7952	0,1093
		256	51,4671	0,6352	23,0989	0,1565	59,7461	0,5630	22,1735	0,1217
		512	49,4016	0,6590	23,6220	0,1771	57,2653	0,5799	22,4807	0,1328
	20x20	64	61,2851	0,5576	21,2825	0,0950	73,6585	0,4885	20,4811	0,0759
		128	58,4063	0,5792	21,7787	0,1097	69,1622	0,5054	20,8898	0,0848
		256	55,9494	0,6004	22,2701	0,1256	66,0986	0,5175	21,1957	0,0919
		512	53,7777	0,6240	22,8069	0,1437	63,6435	0,5253	21,4240	0,0983
	25x25	64	64,9950	0,5411	20,7187	0,0824	77,9068	0,4693	19,8415	0,0644
		128	61,7945	0,5599	21,2053	0,0946	73,6917	0,4798	20,2144	0,0699
		256	59,3621	0,5794	21,7006	0,1084	70,8739	0,4856	20,4765	0,0747
		512	57,0178	0,6050	22,2604	0,1262	68,4632	0,4891	20,6536	0,0781
	30x30	64	67,5145	0,5296	20,2843	0,0746	80,9700	0,4548	19,3328	0,0573
		128	64,6095	0,5457	20,7523	0,0846	77,1954	0,4581	19,6673	0,0607
		256	62,0619	0,5660	21,2610	0,0983	74,5476	0,4596	19,8865	0,0634
		512	59,5118	0,5951	21,8821	0,1182	72,5340	0,4620	20,0282	0,0657
B	5x5	64	40,5461	0,7646	25,8402	0,2891	49,1082	0,7051	25,0350	0,2562
		128	37,6707	0,7976	26,5721	0,3244	43,9256	0,7435	25,7020	0,2863
		256	35,2800	0,8254	27,2981	0,3571	40,3558	0,7785	26,3753	0,3155
		512	33,2564	0,8482	28,0092	0,3887	37,5952	0,8044	26,9707	0,3423
	10x10	64	50,7105	0,6399	23,3241	0,1656	61,3573	0,5696	22,5835	0,1381
		128	47,9559	0,6705	23,8966	0,1913	56,4840	0,6003	23,0588	0,1574
		256	45,5340	0,6993	24,4674	0,2172	52,7969	0,6303	23,5035	0,1753
		512	43,5004	0,7262	25,0589	0,2433	50,4133	0,6536	23,8743	0,1913
	15x15	64	56,8227	0,5860	22,1156	0,1178	69,2435	0,5117	21,2858	0,0945
		128	53,9311	0,6131	22,6521	0,1384	64,5405	0,5351	21,7011	0,1065
		256	51,5279	0,6382	23,1848	0,1583	61,0266	0,5528	22,0561	0,1171
		512	49,3138	0,6660	23,7858	0,1810	58,9506	0,5664	22,3132	0,1258
	20x20	64	61,2685	0,5585	21,3385	0,0956	74,7666	0,4794	20,4120	0,0739
		128	58,3440	0,5817	21,8655	0,1111	70,6896	0,4934	20,7760	0,0810
		256	55,9220	0,6069	22,4346	0,1290	67,3961	0,5039	21,0444	0,0872
		512	53,3134	0,6383	23,1045	0,1525	65,3315	0,5095	21,2250	0,0919
	25x25	64	64,6436	0,5428	20,7938	0,0832	78,2750	0,4602	19,7691	0,0623
		128	61,6210	0,5641	21,3265	0,0966	74,6990	0,4660	20,0813	0,0664
		256	58,9796	0,5903	21,9311	0,1147	72,2217	0,4691	20,2986	0,0696
		512	55,7974	0,6298	22,7413	0,1441	70,5750	0,4714	20,4250	0,0720
	30x30	64	67,2230	0,5309	20,3741	0,0755	81,5909	0,4418	19,2360	0,0537
		128	64,1056	0,5525	20,9135	0,0884	78,1506	0,4433	19,4995	0,0562
		256	61,4090	0,5821	21,5550	0,1082	76,0845	0,4443	19,6702	0,0584
		512	57,0275	0,6333	22,5732	0,1519	74,8306	0,4481	19,7626	0,0602
C	5x5	64	40,3999	0,7678	25,9368	0,2908	51,8834	0,6927	24,9500	0,2536
		128	37,5443	0,8002	26,6992	0,3269	46,0409	0,7334	25,6854	0,2837
		256	35,1030	0,8283	27,4509	0,3608	41,8474	0,7689	26,3413	0,3117
		512	33,0193	0,8514	28,1967	0,3932	38,8105	0,7959	26,9233	0,3374
	10x10	64	50,7447	0,6412	23,4221	0,1661	64,0872	0,5555	22,4772	0,1358
		128	47,7847	0,6745	24,0407	0,1937	59,8339	0,5851	22,9712	0,1539
		256	45,2525	0,7049	24,6581	0,2214	55,8814	0,6128	23,4138	0,1708
		512	43,0786	0,7336	25,3280	0,2496	52,6677	0,6318	23,7519	0,1850
	15x15	64	56,6209	0,5884	22,2315	0,1196	71,7598	0,4970	21,1524	0,0918
		128	53,6137	0,6177	22,8173	0,1411	67,3345	0,5175	21,5973	0,1026
		256	51,0693	0,6463	23,4268	0,1630	64,1874	0,5311	21,9013	0,1116
		512	48,5262	0,6792	24,1742	0,1905	61,8775	0,5442	22,1193	0,1190
	20x20	64	60,9337	0,5619	21,4765	0,0969	76,6423	0,4602	20,2809	0,0710
		128	57,9293	0,5879	22,0719	0,1147	72,9229	0,4687	20,6125	0,0767
		256	55,1490	0,6189	22,7325	0,1357	70,2858	0,4787	20,8520	0,0815
		512	51,6432	0,6629	23,6978	0,1724	68,4533	0,4843	20,9784	0,0857
	25x25	64	64,0902	0,5466	20,9478	0,0845	80,3814	0,4333	19,6010	0,0584
		128	61,0131	0,5728	21,5646	0,1019	77,3461	0,4369	19,8606	0,0621
		256	57,8396	0,6090	22,3278	0,1263	75,2390	0,4421	20,0486	0,0646
		512	52,3920	0,6686	23,5876	0,1835	73,8690	0,4466	20,0605	0,0671
	30x30	64	66,6481	0,5357	20,5332	0,0781	82,9013	0,4151	19,0400	0,0507
		128	63,4320	0,5637	21,1651	0,0951	80,3940	0,4157	19,2651	0,0525
		256	59,4118	0,6096	22,0615	0,1290	78,8229	0,4168	19,2982	0,0541
		512	51,5760	0,6850	23,6254	0,2120	77,6702	0,4212	19,2954	0,0561

Tabela B.11: *Medidas objetivas utilizadas para avaliar a qualidade média das imagens quantizadas com os dicionários convergentes e divergentes com $\epsilon = 0,001$.*

Partição	Janela	Dicionário	Quantização Convergente				Quantização Divergente			
			MSE	SSIM	PSNR	VIF	MSE	SSIM	PSNR	VIF
A	5x5	64	40,3980	0,7687	26,0137	0,2918	49,3965	0,7074	25,1692	0,2580
		128	37,4983	0,8010	26,7852	0,3262	43,4806	0,7501	25,9499	0,2900
		256	35,1093	0,8277	27,5035	0,3591	40,0565	0,7788	26,5890	0,3194
		512	33,0769	0,8497	28,1828	0,3898	37,4081	0,8048	27,1708	0,3457
	10x10	64	50,9265	0,6426	23,4324	0,1665	62,0890	0,5723	22,6570	0,1400
		128	47,8629	0,6745	24,0172	0,1929	56,4566	0,6079	23,2030	0,1605
		256	45,4785	0,7015	24,5636	0,2181	52,7662	0,6348	23,6625	0,1797
		512	43,4356	0,7261	25,0989	0,2432	49,5212	0,6594	24,0585	0,1971
	15x15	64	57,0116	0,5895	22,1901	0,1200	69,2403	0,5220	21,3758	0,0968
		128	54,0126	0,6151	22,7129	0,1389	64,5319	0,5432	21,8376	0,1100
		256	51,5584	0,6387	23,2154	0,1589	60,8026	0,5636	22,2160	0,1225
		512	49,4612	0,6620	23,7250	0,1790	57,7606	0,5790	22,5048	0,1330
	20x20	64	61,5381	0,5610	21,3848	0,0962	74,9739	0,4886	20,5055	0,0767
		128	58,4680	0,5828	21,8907	0,1116	70,0524	0,5067	20,9333	0,0853
		256	56,0198	0,6037	22,3764	0,1276	66,6723	0,5184	21,2351	0,0927
		512	53,7926	0,6267	22,8977	0,1453	64,2440	0,5268	21,4519	0,0986
	25x25	64	64,9435	0,5436	20,8038	0,0842	78,6760	0,4689	19,8527	0,0653
		128	61,8996	0,5625	21,2949	0,0963	74,6605	0,4784	20,2286	0,0704
		256	59,4023	0,5827	21,7863	0,1103	71,3687	0,4854	20,4906	0,0749
		512	57,0006	0,6081	22,3595	0,1281	68,8050	0,4902	20,6679	0,0784
	30x30	64	67,4964	0,5313	20,3504	0,0753	81,0162	0,4544	19,3717	0,0566
		128	64,4561	0,5490	20,8368	0,0865	77,6893	0,4586	19,6979	0,0606
		256	61,9940	0,5686	21,3392	0,0997	74,7591	0,4591	19,9048	0,0634
		512	59,3707	0,5979	21,9650	0,1202	72,6712	0,4626	20,0371	0,0656
B	5x5	64	40,7753	0,7676	25,9850	0,2907	50,4106	0,7061	25,1055	0,2563
		128	37,8443	0,8002	26,7549	0,3255	44,6016	0,7462	25,8469	0,2873
		256	35,3950	0,8274	27,4741	0,3587	41,0032	0,7762	26,4691	0,3158
		512	33,3388	0,8499	28,1827	0,3899	38,2300	0,8037	27,0480	0,3416
	10x10	64	50,9776	0,6431	23,4349	0,1668	62,1635	0,5719	22,6222	0,1385
		128	48,0508	0,6743	24,0258	0,1936	57,0370	0,6038	23,1256	0,1578
		256	45,6357	0,7023	24,5937	0,2192	53,3994	0,6288	23,5530	0,1757
		512	43,5114	0,7286	25,1769	0,2452	51,1299	0,6503	23,8936	0,1913
	15x15	64	56,9808	0,5899	22,2141	0,1199	69,4879	0,5163	21,3381	0,0949
		128	53,9695	0,6168	22,7524	0,1396	64,9602	0,5393	21,7520	0,1068
		256	51,4926	0,6419	23,2983	0,1609	61,4623	0,5529	22,0919	0,1177
		512	49,2797	0,6687	23,8946	0,1834	59,5632	0,5645	22,3136	0,1262
	20x20	64	61,4318	0,5615	21,4277	0,0970	75,0208	0,4820	20,4348	0,0742
		128	58,3906	0,5848	21,9613	0,1127	71,0745	0,4939	20,8059	0,0808
		256	55,7911	0,6098	22,5332	0,1312	67,5390	0,5031	21,0726	0,0873
		512	53,2509	0,6408	23,2145	0,1548	65,4827	0,5094	21,2291	0,0919
	25x25	64	64,6266	0,5459	20,8799	0,0846	78,2940	0,4634	19,7866	0,0619
		128	61,5922	0,5669	21,4126	0,0983	75,1632	0,4671	20,0971	0,0663
		256	59,0552	0,5921	21,9972	0,1155	72,6779	0,4677	20,2962	0,0696
		512	55,5877	0,6329	22,8467	0,1475	70,8297	0,4714	20,4186	0,0722
	30x30	64	67,0006	0,5344	20,4569	0,0757	81,4420	0,4424	19,2418	0,0533
		128	64,0215	0,5548	20,9955	0,0893	78,7290	0,4431	19,5081	0,0564
		256	61,2710	0,5852	21,6415	0,1096	76,7056	0,4431	19,6731	0,0584
		512	56,7121	0,6366	22,6800	0,1548	74,9752	0,4444	19,7640	0,0606
C	5x5	64	40,5737	0,7702	26,0910	0,2927	52,4385	0,6973	25,0453	0,2539
		128	37,6344	0,8033	26,8725	0,3282	46,7048	0,7364	25,7815	0,2846
		256	35,2026	0,8302	27,6221	0,3621	42,8155	0,7677	26,4122	0,3122
		512	33,0069	0,8532	28,3688	0,3953	39,7311	0,7938	26,9675	0,3369
	10x10	64	50,8449	0,6465	23,5444	0,1681	65,0797	0,5583	22,5063	0,1360
		128	47,8802	0,6779	24,1620	0,1955	60,4866	0,5875	23,0232	0,1542
		256	45,3866	0,7075	24,7813	0,2235	56,7189	0,6102	23,4302	0,1706
		512	43,1062	0,7357	25,4486	0,2515	53,5106	0,6303	23,7639	0,1844
	15x15	64	56,5928	0,5927	22,3392	0,1221	71,4723	0,4985	21,2134	0,0921
		128	53,6128	0,6213	22,9241	0,1431	67,6274	0,5181	21,6276	0,1027
		256	51,0896	0,6492	23,5291	0,1655	64,7928	0,5299	21,9116	0,1119
		512	48,4117	0,6823	24,2902	0,1936	62,4138	0,5425	22,1218	0,1191
	20x20	64	60,9078	0,5645	21,5584	0,0979	76,5339	0,4619	20,3154	0,0705
		128	57,8189	0,5912	22,1497	0,1158	73,1672	0,4702	20,6412	0,0770
		256	55,1795	0,6208	22,8105	0,1374	70,7000	0,4772	20,8617	0,0817
		512	51,3580	0,6648	23,7826	0,1748	69,0602	0,4846	20,9981	0,0855
	25x25	64	64,0646	0,5502	21,0426	0,0861	80,6414	0,4346	19,5896	0,0586
		128	61,0525	0,5761	21,6501	0,1036	77,6508	0,4368	19,8620	0,0621
		256	57,7968	0,6120	22,4186	0,1287	75,3814	0,4418	20,0302	0,0648
		512	52,0667	0,6709	23,6978	0,1881	74,3051	0,4475	20,1216	0,0674
	30x30	64	66,6177	0,5388	20,6153	0,0790	83,1417	0,4164	19,0396	0,0503
		128	63,3912	0,5663	21,2318	0,0961	80,4751	0,4143	19,2443	0,0522
		256	59,3292	0,6122	22,1586	0,1319	78,8539	0,4178	19,3850	0,0542
		512	51,1179	0,6875	23,8195	0,2194	77,6543	0,4221	19,4638	0,0560

Tabela B.12: *Medidas objetivas utilizadas para avaliar a qualidade média das imagens quantizadas com os dicionários convergentes e divergentes com $\epsilon = 0,0001$.*

Partição	Janela	Dicionário	Quantização Convergente				Quantização Divergente			
			MSE	SSIM	PSNR	VIF	MSE	SSIM	PSNR	VIF
A	5x5	64	40,5795	0,7690	26,0955	0,2925	50,8967	0,7106	25,2001	0,2581
		128	37,6571	0,8018	26,8702	0,3268	45,3793	0,7486	25,9556	0,2894
		256	35,2507	0,8278	27,5812	0,3588	40,8173	0,7774	26,6250	0,3184
		512	33,1721	0,8499	28,2532	0,3891	37,6775	0,8050	27,2197	0,3448
	10x10	64	50,9711	0,6444	23,4962	0,1666	63,1613	0,5772	22,6846	0,1395
		128	48,0212	0,6757	24,0745	0,1926	57,8464	0,6079	23,2265	0,1597
		256	45,6053	0,7023	24,6168	0,2181	53,4954	0,6326	23,6677	0,1791
		512	43,5408	0,7267	25,1483	0,2431	50,3664	0,6552	24,0563	0,1962
	15x15	64	57,2318	0,5899	22,2392	0,1195	70,5100	0,5243	21,3734	0,0958
		128	54,1284	0,6161	22,7585	0,1390	65,1255	0,5455	21,8691	0,1095
		256	51,6463	0,6395	23,2569	0,1590	61,3540	0,5627	22,2314	0,1223
		512	49,5329	0,6629	23,7681	0,1795	58,2114	0,5777	22,5144	0,1326
	20x20	64	61,6460	0,5610	21,4184	0,0964	75,1737	0,4901	20,5120	0,0757
		128	58,5801	0,5831	21,9164	0,1118	70,9797	0,5052	20,9384	0,0848
		256	56,0944	0,6047	22,4064	0,1279	67,1319	0,5152	21,2347	0,0924
		512	53,8913	0,6274	22,9296	0,1455	64,6653	0,5237	21,4496	0,0985
	25x25	64	65,0232	0,5439	20,8417	0,0844	79,5173	0,4670	19,8078	0,0652
		128	61,9558	0,5631	21,3238	0,0966	75,4474	0,4764	20,2055	0,0705
		256	59,4179	0,5832	21,8146	0,1107	72,0546	0,4837	20,4817	0,0750
		512	57,0282	0,6086	22,3850	0,1285	69,4581	0,4878	20,6413	0,0785
	30x30	64	67,7667	0,5319	20,3918	0,0759	82,1636	0,4532	19,3226	0,0568
		128	64,5949	0,5492	20,8647	0,0866	78,7389	0,4553	19,6518	0,0606
		256	62,0700	0,5699	21,3703	0,1003	75,4038	0,4587	19,8861	0,0635
		512	59,4216	0,5991	22,0101	0,1211	73,2171	0,4619	20,0335	0,0655
B	5x5	64	40,8514	0,7691	26,0582	0,2917	51,0775	0,7127	25,1509	0,2569
		128	37,9192	0,8016	26,8343	0,3263	45,7598	0,7471	25,8697	0,2872
		256	35,5062	0,8280	27,5515	0,3588	41,4566	0,7774	26,5154	0,3152
		512	33,3979	0,8501	28,2492	0,3896	38,4213	0,8022	27,0843	0,3408
	10x10	64	51,1879	0,6455	23,4907	0,1673	63,7175	0,5736	22,6094	0,1381
		128	48,1218	0,6758	24,0821	0,1937	57,7635	0,6041	23,1504	0,1578
		256	45,7247	0,7032	24,6433	0,2197	54,5940	0,6263	23,5611	0,1756
		512	43,5942	0,7291	25,2216	0,2455	51,5240	0,6484	23,9027	0,1909
	15x15	64	57,1160	0,5913	22,2599	0,1198	70,6244	0,5152	21,3198	0,0941
		128	53,9800	0,6177	22,7959	0,1406	65,5597	0,5361	21,7461	0,1066
		256	51,5852	0,6426	23,3324	0,1614	62,2425	0,5506	22,0638	0,1174
		512	49,3259	0,6696	23,9380	0,1838	59,8351	0,5630	22,2982	0,1258
	20x20	64	61,4332	0,5629	21,4724	0,0968	75,5372	0,4835	20,4298	0,0739
		128	58,4834	0,5854	21,9925	0,1129	71,8713	0,4925	20,7916	0,0811
		256	55,8985	0,6105	22,5578	0,1312	67,9798	0,5022	21,0730	0,0869
		512	53,2465	0,6419	23,2578	0,1551	65,4440	0,5106	21,2519	0,0915
	25x25	64	64,8022	0,5464	20,9127	0,0847	79,2974	0,4608	19,7748	0,0615
		128	61,7341	0,5672	21,4251	0,0980	75,4089	0,4651	20,0915	0,0659
		256	59,0839	0,5936	22,0245	0,1160	72,8803	0,4669	20,2829	0,0693
		512	55,5446	0,6346	22,9017	0,1488	70,9684	0,4741	20,4286	0,0719
	30x30	64	67,1118	0,5345	20,4721	0,0764	82,2357	0,4413	19,2234	0,0536
		128	64,0545	0,5565	21,0177	0,0904	79,1143	0,4425	19,5013	0,0563
		256	61,2100	0,5862	21,6675	0,1107	76,8366	0,4438	19,6757	0,0584
		512	56,7360	0,6377	22,7367	0,1567	74,8538	0,4464	19,7764	0,0600
C	5x5	64	40,7787	0,7714	26,1537	0,2930	53,3000	0,7027	25,0517	0,2533
		128	37,7425	0,8039	26,9437	0,3285	47,0402	0,7380	25,7989	0,2837
		256	35,2758	0,8304	27,6858	0,3619	43,7414	0,7659	26,4168	0,3111
		512	33,0860	0,8532	28,4298	0,3946	40,1868	0,7921	26,9815	0,3359
	10x10	64	50,9777	0,6461	23,5882	0,1678	65,2473	0,5612	22,5327	0,1359
		128	47,9402	0,6785	24,2013	0,1958	60,7030	0,5872	23,0358	0,1542
		256	45,4359	0,7077	24,8167	0,2230	56,7484	0,6110	23,4520	0,1706
		512	43,1694	0,7355	25,4848	0,2513	53,8917	0,6296	23,7647	0,1841
	15x15	64	56,7791	0,5938	22,3796	0,1216	72,6599	0,4982	21,1561	0,0919
		128	53,7477	0,6212	22,9473	0,1428	67,8197	0,5185	21,6148	0,1021
		256	51,1157	0,6495	23,5504	0,1651	64,8832	0,5298	21,8969	0,1114
		512	48,4327	0,6822	24,3162	0,1933	62,7841	0,5424	22,1090	0,1190
	20x20	64	61,0992	0,5658	21,5968	0,0986	77,4448	0,4601	20,2948	0,0709
		128	58,0192	0,5918	22,1813	0,1163	73,8482	0,4698	20,6381	0,0769
		256	55,1220	0,6223	22,8502	0,1380	70,6488	0,4781	20,8706	0,0816
		512	51,4070	0,6655	23,8399	0,1756	69,1259	0,4838	20,9962	0,0854
	25x25	64	64,2277	0,5507	21,0709	0,0873	80,8218	0,4355	19,5920	0,0588
		128	61,0870	0,5764	21,6742	0,1035	78,0547	0,4393	19,8525	0,0620
		256	57,7332	0,6131	22,4595	0,1298	75,7359	0,4435	20,0427	0,0646
		512	51,9563	0,6721	23,7912	0,1910	74,1696	0,4478	20,1464	0,0670
	30x30	64	66,7337	0,5384	20,6451	0,0793	83,6893	0,4167	19,0216	0,0503
		128	63,4829	0,5672	21,2891	0,0970	80,6912	0,4140	19,2657	0,0527
		256	59,3082	0,6125	22,2184	0,1338	79,1011	0,4164	19,3742	0,0541
		512	50,8013	0,6879	23,8724	0,2239	77,7378	0,4213	19,4596	0,0562

Tabela B.13: *Medidas objetivas utilizadas para avaliar a qualidade média das imagens quantizadas com os dicionários convergentes e divergentes com $\epsilon = 0,00001$.*

Partição	Janela	Dicionário	Quantização Convergente				Quantização Divergente				
			MSE	SSIM	PSNR	VIF	MSE	SSIM	PSNR	VIF	
A	5x5	64	40,3814	0,7693	26,0687	0,2927	50,033	0,7108	25,2077	0,2588	
		128	37,5861	0,8018	26,8506	0,3268	44,7543	0,7491	25,9682	0,29	
		256	35,2314	0,8278	27,5657	0,3588	40,5229	0,7788	26,624	0,3185	
		512	33,1566	0,8499	28,2417	0,3894	37,6838	0,8045	27,2028	0,3448	
	10x10	64	50,934	0,6447	23,4842	0,1663	63,167	0,5757	22,6509	0,1392	
		128	48,0116	0,675	24,0625	0,193	57,8883	0,6085	23,22	0,16	
		256	45,5621	0,7022	24,6066	0,2184	53,3921	0,6324	23,6647	0,1794	
		512	43,509	0,7267	25,1391	0,2432	50,3055	0,6561	24,0507	0,1966	
	15x15	64	57,0442	0,5908	22,2336	0,1187	70,3357	0,5213	21,3741	0,0956	
		128	54,137	0,6159	22,7491	0,139	64,8795	0,5445	21,8687	0,1095	
		256	51,6293	0,6395	23,2512	0,159	61,0638	0,5623	22,2252	0,1221	
		512	49,5167	0,6625	23,759	0,1794	58,1537	0,578	22,5151	0,1327	
	20x20	64	61,5881	0,5613	21,424	0,0962	75,6917	0,4894	20,5173	0,0763	
		128	58,5512	0,583	21,9174	0,1116	71,165	0,5041	20,9129	0,0844	
		256	56,0679	0,6044	22,4067	0,1279	67,2834	0,5163	21,2227	0,092	
		512	53,852	0,6275	22,9328	0,1457	64,591	0,5267	21,4454	0,0982	
	25x25	64	65,0274	0,5438	20,8365	0,0842	79,0667	0,468	19,8409	0,0652	
		128	61,8838	0,5635	21,3206	0,0966	74,9381	0,4774	20,2173	0,0704	
		256	59,364	0,5831	21,809	0,1108	71,6279	0,4843	20,4845	0,0748	
		512	57,0126	0,6085	22,3796	0,1284	69,1917	0,4903	20,6435	0,0783	
	30x30	64	67,5447	0,5327	20,3917	0,0751	81,8504	0,453	19,3327	0,0563	
		128	64,5395	0,5497	20,8641	0,087	78,31	0,4575	19,6776	0,0605	
		256	62,0189	0,57	21,3748	0,1002	75,2477	0,4582	19,9	0,0636	
		512	59,3932	0,5992	22,0113	0,1208	73,2891	0,4612	20,0193	0,0656	
	B	5x5	64	40,8633	0,7691	26,0436	0,2911	50,7801	0,7116	25,1476	0,2566
			128	37,8973	0,8015	26,8202	0,3261	45,2084	0,7475	25,8737	0,2874
			256	35,4517	0,828	27,5407	0,3591	41,3601	0,7774	26,5121	0,3156
			512	33,3596	0,8502	28,2356	0,39	38,5176	0,8021	27,0736	0,341
		10x10	64	51,061	0,6443	23,4709	0,1669	63,0471	0,5711	22,6099	0,1382
			128	48,0841	0,6756	24,0618	0,1935	57,9317	0,6021	23,1246	0,1576
			256	45,6782	0,7032	24,6287	0,2193	53,9563	0,6276	23,5487	0,1754
			512	43,5784	0,7293	25,2118	0,2459	51,4399	0,65	23,9003	0,1909
		15x15	64	57,1727	0,5908	22,248	0,1203	70,9635	0,5141	21,2863	0,0943
			128	53,9977	0,6173	22,7885	0,1405	65,5696	0,5357	21,7421	0,1064
			256	51,5655	0,6423	23,3269	0,1611	61,9742	0,5507	22,0763	0,117
			512	49,3253	0,6693	23,9267	0,1835	59,9129	0,5632	22,3061	0,1257
20x20		64	61,3899	0,5624	21,4613	0,0966	75,3813	0,4834	20,4464	0,0736	
		128	58,397	0,5859	21,9988	0,1132	71,5492	0,494	20,8149	0,0808	
		256	55,8298	0,6103	22,5481	0,1309	67,833	0,5023	21,0765	0,0872	
		512	53,2652	0,6418	23,2485	0,1552	65,775	0,5094	21,2397	0,0919	
25x25		64	64,8475	0,5455	20,91	0,0845	79,4273	0,4606	19,7461	0,0615	
		128	61,754	0,5672	21,4427	0,0986	75,7051	0,4664	20,0807	0,0665	
		256	59,0475	0,593	22,0275	0,1156	73,0211	0,4686	20,2918	0,0693	
		512	55,5346	0,6343	22,9067	0,1486	71,0179	0,4725	20,4135	0,072	
30x30		64	67,1337	0,5339	20,4707	0,0764	81,9613	0,4416	19,2323	0,0536	
		128	64,0664	0,5561	21,0147	0,09	78,7846	0,4426	19,5107	0,0563	
		256	61,2209	0,5858	21,6682	0,1107	76,8133	0,4442	19,6817	0,0582	
		512	56,6358	0,637	22,722	0,1561	74,9254	0,4463	19,7801	0,0604	
C		5x5	64	40,5893	0,7714	26,1326	0,2927	52,2274	0,7011	25,0824	0,2544
			128	37,7113	0,8037	26,9238	0,3283	47,2708	0,7367	25,7769	0,2838
			256	35,2678	0,8303	27,6696	0,362	43,3035	0,767	26,416	0,3111
			512	33,0608	0,8535	28,4143	0,395	39,9636	0,7932	26,9844	0,336
		10x10	64	50,9123	0,6474	23,584	0,1683	65,2072	0,5629	22,5369	0,1354
			128	47,9533	0,6786	24,1931	0,1961	60,672	0,5885	23,0236	0,1542
			256	45,3614	0,7083	24,8123	0,2232	56,688	0,6111	23,4388	0,1705
			512	43,1395	0,7362	25,4811	0,2516	53,7859	0,6301	23,7585	0,1845
		15x15	64	56,6804	0,5939	22,3717	0,1211	72,4547	0,4977	21,1523	0,0916
			128	53,6775	0,6218	22,9525	0,1432	67,8413	0,5174	21,6048	0,1027
			256	51,1174	0,6496	23,5655	0,1655	64,5664	0,5315	21,9227	0,112
			512	48,4963	0,6821	24,3248	0,1934	62,9012	0,5405	22,0959	0,1191
	20x20	64	60,9328	0,5654	21,5849	0,0986	76,6849	0,4619	20,3049	0,0702	
		128	57,97	0,5916	22,166	0,1161	73,4329	0,4703	20,6299	0,0763	
		256	55,1473	0,6222	22,8358	0,1381	70,7421	0,477	20,8659	0,0815	
		512	51,3399	0,6661	23,8147	0,176	69,0652	0,4841	21,0042	0,0853	
	25x25	64	64,2332	0,55	21,0565	0,0863	81,0081	0,4339	19,5857	0,0586	
		128	60,8695	0,5763	21,6716	0,1032	77,9893	0,4372	19,8513	0,0621	
		256	57,7167	0,6126	22,4373	0,1289	75,7859	0,4416	20,0344	0,065	
		512	51,8826	0,6724	23,7676	0,19	74,2589	0,4477	20,1365	0,0676	
	30x30	64	66,6642	0,538	20,6227	0,0794	83,5684	0,4142	19,0099	0,0508	
		128	63,3919	0,567	21,2639	0,0967	80,6873	0,4164	19,2608	0,0528	
		256	59,1689	0,6125	22,1794	0,133	78,8918	0,4167	19,3747	0,0542	
		512	50,9714	0,6882	23,8498	0,2213	77,637	0,4228	19,4736	0,0562	

Tabela B.14: Desempenho do conjunto de testes da partição A utilizando compressores.

ID	Part.	Conv.	Comp.	W	Acurácia				Precision				Recall				F1 score				Validação	Gener.
					Máximo	Mínimo	Média	Desvio	Máximo	Mínimo	Média	Desvio	Máximo	Mínimo	Média	Desvio	Máximo	Mínimo	Média	Desvio		
1				7	0,8543	0,7971	0,8237	0,0200	0,8639	0,8294	0,8498	0,0121	0,8543	0,7971	0,8237	0,0200	0,8526	0,7915	0,8202	0,0219	0,9171	-0,0934
2			<i>bzip2</i>	8	0,7475	0,6975	0,7208	0,0175	0,6741	0,6375	0,6526	0,0116	0,7475	0,6975	0,7208	0,0175	0,7007	0,6500	0,6727	0,0179	0,8025	-0,0817
3				9	0,6644	0,6200	0,6407	0,0156	0,5417	0,5076*	0,5285	0,0090	0,6644	0,6200	0,6407	0,0156	0,5862	0,5483	0,5663	0,0135	0,7133	-0,0727
6				6	0,8900	0,8433	0,8660	0,0174	0,8964	0,8570	0,8777	0,0152	0,8900	0,8433	0,8660	0,0174	0,8888	0,8424	0,8651	0,0176	0,9300	-0,0640
7				7	0,8457	0,8057	0,8197	0,0143	0,8509	0,8246	0,8380	0,0091	0,8457	0,8057	0,8197	0,0143	0,8417	0,8059	0,8197	0,0124	0,8829	-0,0631
8				8	0,7700	0,7125	0,7488	0,0159	0,7977	0,7558	0,7835	0,0120	0,7700	0,7125	0,7488	0,0159	0,7671	0,7299	0,7517	0,0117	0,8225	-0,0738
9			<i>gzip</i>	9	0,7044	0,6222	0,6840	0,0235	0,7497	0,6895	0,7216	0,0169	0,7044	0,6222	0,6840	0,0235	0,7069	0,6344	0,6850	0,0201	0,7511	-0,0671
10				10	0,6060	0,5020	0,5558	0,0298	0,6946	0,5434	0,6534	0,0466	0,6060	0,5020	0,5558	0,0298	0,5722	0,4615	0,5279	0,0322	0,6820	-0,1262
11				11	0,5491	0,4673	0,5104	0,0248	0,6638	0,5278	0,6079	0,0361	0,5491	0,4673	0,5104	0,0248	0,5207	0,4331	0,4835	0,0279	0,6218	-0,1115
15		0,1		7	0,8971	0,8257	0,8577	0,0277	0,9027	0,8393	0,8749	0,0217	0,8971	0,8257	0,8577	0,0277	0,8975	0,8245	0,8586	0,0272	0,9086	-0,0509
16			<i>LZW+H</i>	8	0,8300	0,7800	0,8093	0,0159	0,8503	0,8177	0,8327	0,0097	0,8300	0,7800	0,8093	0,0159	0,8301	0,7764	0,8101	0,0163	0,8700	-0,0608
17				9	0,7822	0,7200	0,7542	0,0159	0,8106	0,7547	0,7816	0,0145	0,7822	0,7200	0,7542	0,0159	0,7832	0,7124	0,7547	0,0183	0,8267	-0,0724
18				6	0,8900	0,8433	0,8660	0,0174	0,8964	0,8570	0,8777	0,0152	0,8900	0,8433	0,8660	0,0174	0,8888	0,8424	0,8651	0,0176	0,9300	-0,0640
19				7	0,8457	0,8057	0,8197	0,0143	0,8509	0,8246	0,8380	0,0091	0,8457	0,8057	0,8197	0,0143	0,8417	0,8059	0,8197	0,0124	0,8829	-0,0631
20				8	0,7700	0,7125	0,7488	0,0159	0,7977	0,7558	0,7835	0,0120	0,7700	0,7125	0,7488	0,0159	0,7671	0,7299	0,7517	0,0117	0,8225	-0,0738
21			<i>zip</i>	9	0,7044	0,6222	0,6840	0,0235	0,7497	0,6895	0,7216	0,0169	0,7044	0,6222	0,6840	0,0235	0,7069	0,6344	0,6850	0,0201	0,7511	-0,0671
22				10	0,6060	0,5020	0,5558	0,0298	0,6946	0,5434	0,6534	0,0466	0,6060	0,5020	0,5558	0,0298	0,5722	0,4615	0,5279	0,0322	0,6820	-0,1262
23				11	0,5491	0,4673	0,5104	0,0248	0,6638	0,5278	0,6079	0,0361	0,5491	0,4673	0,5104	0,0248	0,5207	0,4331	0,4835	0,0279	0,6218	-0,1115
24				7	0,9286**	0,8771	0,9043	0,0154	0,9369**	0,8852	0,9102	0,0159	0,9286**	0,8771	0,9043	0,0154	0,9294**	0,8763	0,9042	0,0161	0,9171	-0,0129
25				8	0,9000**	0,8575	0,8745	0,0126	0,9172**	0,8740	0,8902	0,0122	0,9000**	0,8575	0,8745	0,0126	0,9012**	0,8577	0,8746	0,0131	0,8850	-0,0105
26			<i>bzip2</i>	9	0,8311	0,7689	0,8024	0,0179	0,8428	0,7827	0,8116	0,0167	0,8311	0,7689	0,8024	0,0179	0,8310	0,7676	0,7992	0,0183	0,8400	-0,0376
27				10	0,7820	0,7200	0,7580	0,0170	0,8032	0,7493	0,7784	0,0174	0,7820	0,7200	0,7580	0,0170	0,7767	0,7147	0,7508	0,0178	0,7920	-0,0340
28				11	0,7473	0,7000	0,7253	0,0150	0,7777	0,7409	0,7649	0,0113	0,7473	0,7000	0,7253	0,0150	0,7510	0,7006	0,7258	0,0145	0,7291	-0,0038
34				6	0,8867	0,8500	0,8710	0,0107	0,8898	0,8530	0,8748	0,0106	0,8867	0,8500	0,8710	0,0107	0,8865	0,8479	0,8699	0,0115	0,9300	-0,0590
35				7	0,8686	0,8143	0,8411	0,0186	0,8787	0,8296	0,8488	0,0165	0,8686	0,8143	0,8411	0,0186	0,8686	0,8129	0,8403	0,0188	0,8914	-0,0503
36				8	0,7275	0,6575	0,7030	0,0222	0,7806	0,7164	0,7466	0,0197	0,7275	0,6575	0,7030	0,0222	0,7146	0,6497	0,6942	0,0230	0,8525	-0,1495
37			<i>gzip</i>	9	0,6933	0,6089	0,6656	0,0287	0,7421	0,6484	0,7041	0,0314	0,6933	0,6089	0,6656	0,0287	0,6863	0,6061	0,6594	0,0287	0,8111	-0,1456
38				10	0,7340	0,6980	0,7158	0,0135	0,7614	0,7108	0,7392	0,0154	0,7340	0,6980	0,7158	0,0135	0,7275	0,6857	0,7092	0,0146	0,7680	-0,0522
39				11	0,6891	0,6618	0,6736	0,0106	0,7236	0,6822	0,7045	0,0150	0,6891	0,6618	0,6736	0,0106	0,6853	0,6516	0,6691	0,0121	0,7273	-0,0536
40				12	0,6317	0,6100	0,6222	0,0081	0,6625	0,6160	0,6415	0,0139	0,6317	0,6100	0,6222	0,0081	0,6198	0,5957	0,6050	0,0086	0,6733	-0,0512
41				7	0,8629	0,8143	0,8343	0,0134	0,8704	0,8266	0,8424	0,0138	0,8629	0,8143	0,8343	0,0134	0,8626	0,8153	0,8339	0,0135	0,9143	-0,0800
42			<i>LZW+H</i>	8	0,8150	0,7600	0,7898	0,0169	0,8260	0,7828	0,8046	0,0130	0,8150	0,7600	0,7898	0,0169	0,8164	0,7633	0,7907	0,0174	0,8800	-0,0903
43				9	0,7600	0,7067	0,7344	0,0204	0,7871	0,7478	0,7733	0,0135	0,7600	0,7067	0,7344	0,0204	0,7557	0,6907	0,7254	0,0244	0,8356	-0,1011
44				10	0,7020	0,6300	0,6702	0,0207	0,7497	0,6944	0,7246	0,0193	0,7020	0,6300	0,6702	0,0207	0,6868	0,6199	0,6527	0,0219	0,7820	-0,1118
45				11	0,6673	0,5964	0,6276	0,0186	0,7016	0,6569	0,6800	0,0154	0,6673	0,5964	0,6276	0,0186	0,6532	0,5866	0,6129	0,0193	0,7345	-0,1069
46				6	0,8867	0,8500	0,8710	0,0107	0,8898	0,8530	0,8748	0,0106	0,8867	0,8500	0,8710	0,0107	0,8865	0,8479	0,8699	0,0115	0,9300	-0,0590
47				7	0,8686	0,8143	0,8411	0,0186	0,8787	0,8296	0,8488	0,0165	0,8686	0,8143	0,8411	0,0186	0,8686	0,8129	0,8403	0,0188	0,8914	-0,0503
48				8	0,7275	0,6575	0,7030	0,0222	0,7806	0,7164	0,7466	0,0197	0,7275	0,6575	0,7030	0,0222	0,7146	0,6497	0,6942	0,0230	0,8525	-0,1495
49			<i>zip</i>	9	0,6933	0,6089	0,6656	0,0287	0,7421	0,6484	0,7041	0,0314	0,6933	0,6089	0,6656	0,0287	0,6863	0,6061	0,6594	0,0287	0,8111	-0,1456
50				10	0,7340	0,6980	0,7158	0,0135	0,7614	0,7108	0,7392	0,0154	0,7340	0,6980	0,7158	0,0135	0,7275	0,6857	0,7092	0,0146	0,7680	-0,0522
51				11	0,6891	0,6618	0,6736	0,0106	0,7236	0,6822	0,7045	0,0150	0,6891	0,6618	0,6736	0,0106	0,6853	0,6516	0,6691	0,0121	0,7273	-0,0536
52				12	0,6317	0,6100	0,6222	0,0081	0,6625	0,6160	0,6415	0,0139	0,6317	0,6100	0,6222	0,0081	0,6198	0,5957	0,6050	0,0086	0,6733	-0,0512
53				7	0,8371	0,7857	0,8217	0,0157	0,8619	0,8252	0,8438	0,0123	0,8371	0,7857	0,8217	0,0157	0,8387	0,7855	0,8202	0,0160	0,9229	-0,1011
54			<i>LZW+H</i>	8	0,8000	0,7300	0,7740	0,0190	0,8120	0,7604	0,7875	0,0156	0,8000	0,7300	0,7740	0,0190	0,7990	0,7230	0,7698	0,0203	0,8850	-0,1110
55				9	0,7578	0,6867	0,7182	0,0222	0,7697	0,7065	0,7306	0,0190	0,7578	0,6867	0,7182	0,0222	0,7535	0,6805	0,7102	0,0237	0,8533	-0,1351
56				10	0,6900	0,6240	0,6568	0,0211	0,7062	0,6495	0,6787	0,0187	0,6900	0,6240	0,6568	0,0211	0,6840	0,6064	0,6748	0,0239	0,7860	-0,1292
57				11	0,6255	0,5491	0,6031	0,0215	0,6836	0,6293	0,6558	0,0151	0,6255	0,5491	0,6031	0,0215	0,61					

Tabela B.15: Desempenho do conjunto de testes das partições B e C utilizando compressores.

ID	Part.	Conv.	Comp.	W	Acurácia				Precision				Recall				F1 score				Validação	Gener.
					Máximo	Mínimo	Média	Desvio	Máximo	Mínimo	Média	Desvio	Máximo	Mínimo	Média	Desvio	Máximo	Mínimo	Média	Desvio		
113					0.5840	0.4987	0.5540	0.0273	0.6404	0.5570	0.5991	0.0242	0.5840	0.4987	0.5540	0.0273	0.5858	0.4839	0.5489	0.0308	0.9120	-0.3580
114			<i>bzip2</i>	11	0.5600	0.4788	0.5229	0.0263	0.6346	0.5361	0.5725	0.0292	0.5600	0.4788	0.5229	0.0263	0.5548	0.4667	0.5198	0.0288	0.8873	-0.3644
118				8	0.5667	0.4950*	0.5232	0.0214	0.6106	0.5331*	0.5754	0.0237	0.5667	0.4950*	0.5232	0.0214	0.5582	0.4946*	0.5175	0.0216	0.9300	-0.4068
119				9	0.5200	0.4622*	0.4865	0.0168	0.6062	0.5132	0.5611	0.0309	0.5200	0.4622*	0.4865	0.0168	0.5068	0.4566*	0.4801	0.0152	0.8978	-0.4113
120			<i>gzip</i>	10	0.5053	0.4333	0.4805	0.0217	0.6159	0.5420	0.5722	0.0218	0.5053	0.4333	0.4805	0.0217	0.5250	0.4441	0.4916	0.0226	0.8680	-0.3875
121				11	0.4982	0.4376	0.4745	0.0177	0.6014	0.5545	0.5730	0.0153	0.4982	0.4376	0.4745	0.0177	0.5177	0.4529	0.4877	0.0181	0.8400	-0.3655
122				12	0.4767	0.4222	0.4530	0.0157	0.5953	0.5440	0.5625	0.0160	0.4767	0.4222	0.4530	0.0157	0.4889	0.4392	0.4651	0.0146	0.8067	-0.3537
123				8	0.7183	0.6367	0.6742	0.0280	0.7352	0.6748	0.6995	0.0203	0.7183	0.6367	0.6742	0.0280	0.7162	0.6275	0.6692	0.0293	0.9150	-0.2408
124				9	0.7659	0.7052	0.7314	0.0189	0.7765	0.7337	0.7525	0.0157	0.7659	0.7052	0.7314	0.0189	0.7608	0.7003	0.7260	0.0183	0.8889	-0.1575
125			<i>LZW+H</i>	10	0.7227	0.6467	0.6880	0.0226	0.7287	0.6821	0.7083	0.0178	0.7227	0.6467	0.6880	0.0226	0.7197	0.6407	0.6830	0.0221	0.8400	-0.1520
126				11	0.6230	0.5576	0.5928	0.0174	0.6269	0.5659	0.5994	0.0176	0.6230	0.5576	0.5928	0.0174	0.6107	0.5337	0.5798	0.0206	0.8000	-0.2072
127				12	0.5989	0.5311	0.5749	0.0211	0.6200	0.5525	0.5865	0.0182	0.5989	0.5311	0.5749	0.0211	0.5883	0.5103	0.5632	0.0242	0.7533	-0.1784
128				8	0.5667	0.4950	0.5232	0.0214	0.6106	0.5331	0.5754	0.0237	0.5667	0.4950	0.5232	0.0214	0.5582	0.4946	0.5175	0.0216	0.9300	-0.4068
129				9	0.5200	0.4622	0.4865	0.0168	0.6062	0.5132	0.5611	0.0309	0.5200	0.4622	0.4865	0.0168	0.5068	0.4566	0.4801	0.0152	0.8978	-0.4113
130			<i>zip</i>	10	0.5053	0.4333	0.4805	0.0217	0.6159	0.5420	0.5724	0.0217	0.5053	0.4333	0.4805	0.0217	0.5250	0.4443	0.4916	0.0225	0.8680	-0.3875
131				11	0.4982	0.4376	0.4745	0.0177	0.6014	0.5545	0.5731	0.0153	0.4982	0.4376	0.4745	0.0177	0.5177	0.4530	0.4877	0.0181	0.8400	-0.3655
132				12	0.4767	0.4222	0.4530	0.0157	0.5953	0.5440	0.5626	0.0161	0.4767	0.4222	0.4530	0.0157	0.4889	0.4393	0.4651	0.0146	0.8067	-0.3537
133				10	0.5547	0.4773	0.5135	0.0244	0.6338	0.5746	0.6070	0.0205	0.5547	0.4773	0.5135	0.0244	0.5460	0.4777	0.5040	0.0261	0.9160	-0.4025
134			<i>bzip2</i>	11	0.5624	0.4970	0.5238	0.0190	0.6040	0.5247	0.5761	0.0240	0.5624	0.4970	0.5238	0.0190	0.5573	0.4955	0.5191	0.0206	0.8982	-0.3744
135				12	0.5522	0.4867	0.5160	0.0179	0.6007	0.5234	0.5685	0.0213	0.5522	0.4867	0.5160	0.0179	0.5487	0.4879	0.5114	0.0193	0.8800	-0.3640
139				9	0.6133	0.5452	0.5812	0.0228	0.6398	0.5982	0.6178	0.0128	0.6133	0.5452	0.5812	0.0228	0.6144	0.5489	0.5797	0.0206	0.9067	-0.3255
140			<i>gzip</i>	10	0.5400	0.4720	0.5099	0.0221	0.5933	0.5058	0.5343	0.0206	0.5400	0.4720	0.5099	0.0221	0.5317	0.4614	0.5023	0.0225	0.8840	-0.3741
141				11	0.5261	0.4727	0.4977	0.0202	0.5549	0.5089	0.5259	0.0157	0.5261	0.4727	0.4977	0.0202	0.5183	0.4640	0.4899	0.0194	0.8618	-0.3641
142				8	0.8150	0.7550	0.7907	0.0165	0.8377	0.7989	0.8106	0.0107	0.8150	0.7550	0.7907	0.0165	0.8152	0.7546	0.7901	0.0170	0.9150	-0.1243
143				9	0.7541	0.7022	0.7348	0.0167	0.7846	0.7398	0.7547	0.0150	0.7541	0.7022	0.7348	0.0167	0.7472	0.6965	0.7282	0.0180	0.8800	-0.1452
144			<i>LZW+H</i>	10	0.6720	0.5973	0.6369	0.0262	0.6910	0.6203	0.6445	0.0231	0.6720	0.5973	0.6369	0.0262	0.6710	0.5944	0.6319	0.0251	0.8480	-0.2111
145				11	0.6170	0.5600	0.5898	0.0178	0.6486	0.6030	0.6266	0.0183	0.6170	0.5600	0.5898	0.0178	0.6080	0.5415	0.5749	0.0213	0.8255	-0.2356
146				12	0.5867	0.5178	0.5558	0.0216	0.6339	0.5605	0.5966	0.0227	0.5867	0.5178	0.5558	0.0216	0.5714	0.4931	0.5396	0.0248	0.7867	-0.2309
147				9	0.6133	0.5452	0.5812	0.0228	0.6398	0.5982	0.6178	0.0128	0.6133	0.5452	0.5812	0.0228	0.6144	0.5489	0.5797	0.0206	0.9067	-0.3255
148			<i>zip</i>	10	0.5400	0.4720	0.5099	0.0221	0.5933	0.5058	0.5343	0.0206	0.5400	0.4720	0.5099	0.0221	0.5317	0.4614	0.5023	0.0225	0.8840	-0.3741
149				11	0.5261	0.4727	0.4977	0.0202	0.5549	0.5089	0.5259	0.0157	0.5261	0.4727	0.4977	0.0202	0.5183	0.4640	0.4899	0.0194	0.8618	-0.3641
150			<i>bzip2</i>	11	0.5952	0.5309	0.5484	0.0207	0.6142	0.5579	0.5749	0.0167	0.5952	0.5309	0.5484	0.0207	0.5970	0.5176	0.5398	0.0229	0.9055	-0.3571
151				12	0.5489	0.4778	0.5031	0.0204	0.5790	0.5293	0.5482	0.0168	0.5489	0.4778	0.5031	0.0204	0.5461	0.4711	0.4985	0.0231	0.8900	-0.3869
156				9	0.5526	0.5007	0.5302	0.0192	0.6076	0.5466	0.5865	0.0190	0.5526	0.5007	0.5302	0.0192	0.5475	0.4913	0.5214	0.0210	0.9156	-0.3853
157				10	0.5173	0.4653	0.4892	0.0147	0.5597	0.4885	0.5337	0.0197	0.5173	0.4653	0.4892	0.0147	0.5097	0.4540	0.4764	0.0150	0.8840	-0.3948
158			<i>gzip</i>	11	0.4776	0.4242	0.4474	0.0170	0.5321	0.4554	0.5022	0.0213	0.4776	0.4242	0.4474	0.0170	0.4739	0.4033	0.4356	0.0189	0.8473	-0.3999
159				12	0.4378	0.3889	0.4112	0.0159	0.5279	0.4274	0.4887	0.0348	0.4378	0.3889	0.4112	0.0159	0.4211	0.3613	0.3900	0.0165	0.7967	-0.3854
160			<i>LZW+H</i>	9	0.7526	0.6519	0.7099	0.0274	0.7642	0.6755	0.7256	0.0245	0.7526	0.6519	0.7099	0.0274	0.7541	0.6466	0.7071	0.0295	0.9022	-0.1923
161				10	0.6867	0.6013	0.6573	0.0230	0.6943	0.6320	0.6691	0.0171	0.6867	0.6013	0.6573	0.0230	0.6848	0.5989	0.6525	0.0230	0.8800	-0.2227
162				9	0.5526	0.5007	0.5302	0.0192	0.6076	0.5466	0.5865	0.0190	0.5526	0.5007	0.5302	0.0192	0.5475	0.4913	0.5214	0.0210	0.9156	-0.3853
163				10	0.5173	0.4653	0.4892	0.0147	0.5597	0.4885	0.5337	0.0197	0.5173	0.4653	0.4892	0.0147	0.5097	0.4540	0.4764	0.0150	0.8840	-0.3948
164			<i>zip</i>	11	0.4776	0.4242	0.4474	0.0170	0.5321	0.4554	0.5022	0.0213	0.4776	0.4242	0.4474	0.0170	0.4739	0.4033	0.4356	0.0189	0.8473	-0.3999
165				12	0.4378	0.3889	0.4112	0.0159	0.5279	0.4274	0.4887	0.0348	0.4378	0.3889	0.4112	0.0159	0.4211	0.3613	0.3900	0.0165	0.7967	-0.3854
166				11	0.5636	0.4691	0.5144	0.0288	0.5853	0.5373	0.5577	0.0153	0.5636	0.4691	0.5144	0.0288	0.5529	0.4516	0.5059	0.0303	0.9018	-0.3874
167			<i>bzip2</i>	12	0.5367	0.4556	0.4944	0.0263	0.5917	0.5146	0.5453	0.0241	0.5367	0.4556	0.4944	0.0263	0.5318	0.4403	0.4884	0.0270	0.8833	-0.3889
172			<i>gzip</i>	10	0.5387	0.4213	0.4915	0.0393	0.5945	0.5279	0.5635	0.0211	0.5387	0.4213	0.4915	0.0393	0.5478	0.4257	0.4944	0.0303	0.9000	-0.4085
173			<i>LZW+H</i>	9	0.6785	0.6074	0.6473	0.0242	0.6999													

Tabela B.16: Desempenho do conjunto de testes utilizando entro-pia.

ID	Part.	Conv.	W	Acurácia				Precision				Recall				F1 score				Validação	Gener.
				Máximo	Mínimo	Média	Desvio														
4			7	0,9029	0,8057	0,8683	0,0266	0,9040	0,8295	0,8749	0,0208	0,9029	0,8057	0,8683	0,0266	0,9030	0,8055	0,8675	0,0266	0,9143	-0,0460
5			8	0,8700	0,7825	0,8328	0,0248	0,8756	0,8071	0,8429	0,0212	0,8700	0,7825	0,8328	0,0248	0,8696	0,7817	0,8308	0,0251	0,8750	-0,0422
6		0,1	9	0,8511**	0,7356	0,7862	0,0344	0,8597	0,7725	0,8117	0,0243	0,8511**	0,7356	0,7862	0,0344	0,8504**	0,7317	0,7828	0,0371	0,8378	-0,0516
7			10	0,7900	0,6920	0,7454	0,0298	0,8038	0,7161	0,7657	0,0270	0,7900	0,6920	0,7454	0,0298	0,7847	0,6819	0,7391	0,0323	0,7960	-0,0506
8			11	0,7164	0,6291	0,6769	0,0269	0,6738	0,6013	0,6426	0,0227	0,7164	0,6291	0,6769	0,0269	0,6800	0,5909	0,6410	0,0286	0,7236	-0,0467
29			7	0,8800	0,7714	0,8374	0,0345	0,9074	0,8340	0,8709	0,0209	0,8800	0,7714	0,8374	0,0345	0,8826	0,7731	0,8393	0,0344	0,9200	-0,0826
30			8	0,8675	0,7375	0,7940	0,0393	0,8787	0,7988	0,8228	0,0260	0,8675	0,7375	0,7940	0,0393	0,8686	0,7412	0,7946	0,0393	0,8925	-0,0985
31		0,01	9	0,8333	0,6956	0,7607	0,0392	0,8529	0,7658	0,7997	0,0255	0,8333	0,6956	0,7607	0,0392	0,8358	0,6942	0,7592	0,0416	0,8600	-0,0993
32			10	0,7980	0,6640	0,7222	0,0388	0,8147	0,7268	0,7623	0,0266	0,7980	0,6640	0,7222	0,0388	0,7990	0,6614	0,7189	0,0421	0,8140	-0,0918
33			11	0,6982	0,6055	0,6651	0,0311	0,7385	0,6525	0,7005	0,0330	0,6982	0,6055	0,6651	0,0311	0,6931	0,5947	0,6606	0,0333	0,7564	-0,0913
59			7	0,8771	0,8029	0,8497	0,0280	0,8881	0,8373	0,8655	0,0179	0,8771	0,8029	0,8497	0,0280	0,8775	0,7984	0,8494	0,0293	0,9257	-0,0760
60			8	0,8575	0,7900	0,8253	0,0201	0,8608	0,7969	0,8333	0,0201	0,8575	0,7900	0,8253	0,0201	0,8570	0,7846	0,8241	0,0216	0,8875	-0,0623
61		0,001	9	0,8311	0,7511	0,7891	0,0259	0,8340	0,7647	0,8054	0,0211	0,8311	0,7511	0,7891	0,0259	0,8297	0,7444	0,7850	0,0289	0,8511	-0,0620
62			10	0,8120	0,7040	0,7470	0,0302	0,8285	0,7486	0,7804	0,0253	0,8120	0,7040	0,7470	0,0302	0,8090	0,6951	0,7408	0,0332	0,8160	-0,0690
63			11	0,7145	0,6436	0,6825	0,0195	0,7455	0,7148	0,7297	0,0104	0,7145	0,6436	0,6825	0,0195	0,7058	0,6406	0,6736	0,0187	0,7709	-0,0884
75			8	0,8650	0,7650	0,8300	0,0352	0,8782	0,8038	0,8538	0,0245	0,8650	0,7650	0,8300	0,0352	0,8669	0,7596	0,8269	0,0393	0,9200	-0,0900
76			9	0,8467	0,7556	0,8171	0,0279	0,8609	0,7898	0,8378	0,0194	0,8467	0,7556	0,8171	0,0279	0,8479	0,7512	0,8141	0,0312	0,8933	-0,0762
77		0,0001	10	0,8220**	0,7180	0,7638	0,0290	0,8303**	0,7792	0,8008	0,0150	0,8220**	0,7180	0,7638	0,0290	0,8212**	0,7092	0,7618	0,0306	0,8480	-0,0842
78			11	0,7727**	0,6818	0,7185	0,0275	0,7863**	0,7381	0,7573	0,0156	0,7727**	0,6818	0,7185	0,0275	0,7738**	0,6734	0,7179	0,0296	0,8127	-0,0918
89			8	0,8825	0,8025	0,8503	0,0267	0,8886	0,8335	0,8627	0,0165	0,8825	0,8025	0,8503	0,0267	0,8827	0,7994	0,8493	0,0279	0,9175	-0,0673
90			9	0,8400	0,7844	0,8173	0,0222	0,8627**	0,8190	0,8423	0,0147	0,8400	0,7844	0,8173	0,0222	0,8427	0,7815	0,8178	0,0231	0,8822	-0,0649
91		0,00001	10	0,7900	0,7200	0,7622	0,0267	0,8144	0,7621	0,7903	0,0198	0,7900	0,7200	0,7622	0,0267	0,7936	0,7238	0,7637	0,0275	0,8480	-0,0858
92			11	0,7364	0,6764	0,7078	0,0223	0,7551	0,7122	0,7342	0,0155	0,7364	0,6764	0,7078	0,0223	0,7332	0,6747	0,7048	0,0211	0,8073	-0,0995
93			12	0,6967**	0,6400	0,6698	0,0188	0,7260**	0,6759	0,6982	0,0157	0,6967**	0,6400	0,6698	0,0188	0,6912**	0,6342	0,6643	0,0184	0,7633	-0,0935
115			9	0,6741	0,5822	0,6230	0,0258	0,6749	0,5958	0,6404	0,0204	0,6741	0,5822	0,6230	0,0258	0,6662	0,5528	0,6061	0,0318	0,9156	-0,2926
116		0,1	10	0,6160	0,5360	0,5755	0,0245	0,6256	0,5532	0,5976	0,0223	0,6160	0,5360	0,5755	0,0245	0,6123	0,5082	0,5597	0,0309	0,8760	-0,3005
117			11	0,5709	0,4970	0,5377	0,0263	0,6008	0,5534	0,5747	0,0135	0,5709	0,4970	0,5377	0,0263	0,5683	0,4637	0,5202	0,0342	0,8436	-0,3059
136			9	0,6622	0,5733	0,6126	0,0259	0,6766	0,6090	0,6436	0,0197	0,6622	0,5733	0,6126	0,0259	0,6630	0,5793	0,6129	0,0247	0,9022	-0,2896
137		0,01	10	0,6200	0,5373	0,5677	0,0243	0,6307	0,5589	0,5969	0,0212	0,6200	0,5373	0,5677	0,0243	0,6211	0,5342	0,5666	0,0250	0,8800	-0,3123
138			11	0,5794	0,5164	0,5444	0,0217	0,6124	0,5361	0,5720	0,0234	0,5794	0,5164	0,5444	0,0217	0,5761	0,5017	0,5376	0,0244	0,8582	-0,3138
152			9	0,7289	0,6607	0,7036	0,0219	0,7477	0,6992	0,7289	0,0171	0,7289	0,6607	0,7036	0,0219	0,7288	0,6442	0,6958	0,0260	0,9200	-0,2164
153		0,001	10	0,6480	0,5693	0,6028	0,0234	0,6659	0,5823	0,6230	0,0226	0,6480	0,5693	0,6028	0,0234	0,6444	0,5657	0,5969	0,0251	0,8960	-0,3240
154			11	0,6279	0,5491	0,5858	0,0218	0,6415	0,5569	0,6032	0,0232	0,6279	0,5491	0,5858	0,0218	0,6242	0,5453	0,5794	0,0237	0,8691	-0,2833
155			12	0,5756	0,5033	0,5373	0,0200	0,6080	0,4760	0,5346	0,0450	0,5756	0,5033	0,5373	0,0200	0,5518	0,4825	0,5129	0,0214	0,8033	-0,2660
168			9	0,8311	0,6637	0,7636	0,0441	0,8432	0,7917	0,8064	0,0150	0,8311	0,6637	0,7636	0,0441	0,8334	0,6611	0,7629	0,0456	0,9333	-0,1698
169			10	0,6027	0,5253	0,5720	0,0270	0,6466	0,5933	0,6252	0,0162	0,6027	0,5253	0,5720	0,0270	0,6033	0,5207	0,5744	0,0286	0,8960	-0,3240
170		0,0001	11	0,5709	0,4848	0,5415	0,0312	0,6092	0,5516	0,5849	0,0200	0,5709	0,4848	0,5415	0,0312	0,5683	0,4770	0,5401	0,0333	0,8691	-0,3276
171			12	0,6000	0,5200	0,5558	0,0270	0,6735	0,6152	0,6530	0,0201	0,6000	0,5200	0,5558	0,0270	0,6025	0,5232	0,5555	0,0268	0,8333	-0,2776
178			9	0,7926	0,7230	0,7599	0,0258	0,8155	0,7599	0,7867	0,0187	0,7926	0,7230	0,7599	0,0258	0,7902	0,7165	0,7570	0,0270	0,9067	-0,1468
179			10	0,7520	0,6667	0,7079	0,0318	0,7559	0,6975	0,7284	0,0209	0,7520	0,6667	0,7079	0,0318	0,7447	0,6542	0,6994	0,0328	0,8640	-0,1561
180		0,00001	11	0,6848	0,6012	0,6429	0,0294	0,6810	0,6321	0,6540	0,0164	0,6848	0,6012	0,6429	0,0294	0,6576	0,5719	0,6196	0,0297	0,8145	-0,1716
181			12	0,6222	0,5544	0,5849	0,0241	0,6396	0,5565	0,5991	0,0284	0,6222	0,5544	0,5849	0,0241	0,5814	0,5200	0,5486	0,0225	0,7500	-0,1651
192			10	0,5647	0,4824	0,5260	0,0278	0,5966	0,5559	0,5744	0,0131	0,5647	0,4824	0,5260	0,0278	0,5628	0,4663	0,5229	0,0325	0,9600	-0,4340
193		0,1	11	0,5465	0,4663	0,5024	0,0301	0,5961	0,5319	0,5668	0,0206	0,5465	0,4663	0,5024	0,0301	0,5418	0,4528	0,4987	0,0345	0,9394	-0,4370
194			12	0,4902	0,4235	0,4572	0,0199	0,4962	0,4352	0,4642	0,0183	0,4902	0,4235	0,4572	0,0199	0,4564	0,3896	0,4228	0,0210	0,8611	-0,4040
201			11	0,5444	0,4770	0,5149	0,0232	0,5890	0,5201	0,5506	0,0234	0,5444	0,4770	0,5149	0,0232	0,5416	0,4509	0,5054	0,0294	0,9576	-0,4427
202			12	0,5137	0,4627	0,4908	0,0177	0,5672	0,4901	0,5194	0,0233	0,5137	0,4627	0,4908	0,0177	0,5100	0,4409	0,4777	0,0233	0,9333	-0,4425
211			10	0,5847	0,5235	0,5514	0,0217	0,6143	0,5473	0,5817	0,0197	0,5847	0,5235	0,5514	0,0217	0,5754	0,5120	0,5383	0,0211	0,9600	-0,4086
212		0,001	11	0,5679	0,4952	0,5303	0,0228	0,5768	0,5246	0,5608	0,0156	0,5679	0,4952	0,5303	0,0228	0,5516	0,4813	0,5136	0,0213	0,9333	-0,4031
213			12	0,4892	0,4039	0,4454	0,0270	0,5755	0,4666	0,5264	0,0315	0,4892	0,4039	0,4454	0,0270	0,4755	0,3961	0,4320	0,0271	0,8500	-0,4046
224			10	0,5447	0,4529	0,5129	0,0338	0,5948	0,5542	0,5778	0,0155	0,5447	0,4529	0,5129	0,0338	0,5449	0,4500	0,5111	0,0361	0,9667	-0,4537
225		0,0001	11	0,5337	0,4513	0,5005	0,0310	0,5754	0,5271	0,5581	0,0166	0,5337	0,4513	0,5005	0,0310	0,5313	0,4345	0,4974	0,0334	0,9455	-0,4449
226			12	0,5020	0,4196	0,4647	0,0278	0,5631	0,5009												

Tabela B.17: *R-values* entre as imagens de teste, com as maiores e menores diferenças observadas, e as classes do melhor cenário utilizando compressão LZW+Huffman

Caso	Diferença	Parâmetros <i>R-value</i>	Classes											
			avião 30x30 256	praia 30x30 512	construção 15x15 128	carro 20x20 128	campo 30x30 512	fogos de artifício 30x30 512	flor 25x25 512	corrida de motos 20x20 64	moto 25x25 64	montanha 25x25 256	pôr do sol 30x30 512	árvore 5x5 512
Máxima 0,209607 moto_7165	k=3, θ=1		0,202020	0,119632	0,438623	0,429907	0,138037	0,116564	0,149148	0,450550	0,660156	0,263393	0,127301	0,313441
	k=5, θ=2		0,204545	0,121166	0,381737	0,394860	0,141104	0,113497	0,143466	0,406593	0,609375	0,250000	0,122699	0,211032
	k=7, θ=3		0,199495	0,121166	0,357784	0,369159	0,136503	0,118098	0,142045	0,359890	0,558594	0,252232	0,121166	0,174322
	k=9, θ=4		0,196970	0,121166	0,362275	0,352804	0,131902	0,118098	0,142045	0,357143	0,503906	0,245536	0,119632	0,158697
	k=11, θ=5		0,199495	0,119632	0,350299	0,338785	0,130368	0,118098	0,144886	0,359890	0,507813	0,243304	0,121166	0,145520
	k=13, θ=6		0,196970	0,119632	0,338323	0,327103	0,128834	0,119632	0,144886	0,346154	0,488281	0,243304	0,119632	0,139495
	k=15, θ=7		0,196970	0,119632	0,329341	0,338785	0,130368	0,116564	0,143466	0,343407	0,468750	0,245536	0,119632	0,132718
	k=17, θ=8		0,191919	0,116564	0,324850	0,329439	0,131902	0,118098	0,144886	0,337912	0,468750	0,245536	0,119632	0,129518
	k=19, θ=9		0,196970	0,116564	0,327844	0,331776	0,130368	0,118098	0,146307	0,337912	0,445313	0,241071	0,119632	0,127447
	k=21, θ=10		0,196970	0,118098	0,332335	0,338785	0,130368	0,118098	0,144886	0,340659	0,460938	0,238839	0,119632	0,123682
Acerto	k=3, θ=1		0,535354	0,380368	0,399701	0,469626	0,343558	0,263804	0,326705	0,450549	0,312500	0,482143	0,309816	0,169239
	k=5, θ=2		0,472222	0,329755	0,267964	0,331776	0,311350	0,263804	0,303977	0,285714	0,257813	0,426339	0,280675	0,106363
	k=7, θ=3		0,431818	0,323620	0,251497	0,292056	0,299080	0,260736	0,274148	0,255495	0,261719	0,392857	0,274540	0,095633
	k=9, θ=4		0,404040	0,317485	0,239521	0,285047	0,280675	0,251534	0,281250	0,252747	0,238281	0,383929	0,268405	0,090550
	k=11, θ=5		0,378788	0,308282	0,232036	0,261682	0,262270	0,246933	0,279830	0,252747	0,250000	0,359375	0,265337	0,088479
	k=13, θ=6		0,383838	0,303681	0,227545	0,273364	0,259202	0,250000	0,288352	0,244505	0,246094	0,357143	0,254601	0,086785
	k=15, θ=7		0,366162	0,303681	0,224551	0,282710	0,263804	0,251534	0,281250	0,236264	0,242188	0,366071	0,251534	0,085655
	k=17, θ=8		0,373737	0,299080	0,220060	0,282710	0,256135	0,253067	0,278409	0,236264	0,246094	0,361607	0,242331	0,084149
	k=19, θ=9		0,371212	0,309816	0,220060	0,278037	0,256135	0,254601	0,275568	0,219780	0,250000	0,361607	0,245399	0,084526
	k=21, θ=10		0,391414	0,294479	0,214072	0,271028	0,256135	0,253067	0,282670	0,197802	0,250000	0,366071	0,243865	0,086596
Máxima 0,498963 arvore_3743	k=3, θ=1		0,527778	0,259202	0,782934	0,754673	0,226994	0,406442	0,288352	0,782967	0,582031	0,542411	0,233129	0,530120
	k=5, θ=2		0,469697	0,222393	0,681138	0,707944	0,210123	0,337423	0,264205	0,642857	0,320313	0,493304	0,217791	0,247929
	k=7, θ=3		0,429293	0,216258	0,525449	0,644860	0,210123	0,299080	0,255682	0,500000	0,234375	0,455357	0,208589	0,145896
	k=9, θ=4		0,411616	0,211656	0,384731	0,539720	0,208589	0,280675	0,255682	0,351648	0,222656	0,448661	0,208589	0,112011
	k=11, θ=5		0,419192	0,207055	0,284431	0,436916	0,208589	0,276074	0,254261	0,255495	0,226563	0,448661	0,208589	0,101845
	k=13, θ=6		0,416667	0,208589	0,241018	0,362150	0,210123	0,280675	0,259943	0,214286	0,242188	0,446429	0,211656	0,096574
	k=15, θ=7		0,401515	0,208589	0,208084	0,310748	0,208589	0,277607	0,262784	0,200549	0,246094	0,444196	0,213190	0,095068
	k=17, θ=8		0,396465	0,207055	0,203593	0,292056	0,210123	0,268405	0,262784	0,189560	0,246094	0,444196	0,208589	0,094691
	k=19, θ=9		0,373737	0,210123	0,202096	0,287383	0,211656	0,277607	0,261364	0,189560	0,250000	0,450893	0,210123	0,094880
	k=21, θ=10		0,371212	0,210123	0,199102	0,289720	0,211656	0,271472	0,262784	0,184066	0,250000	0,457589	0,210123	0,094691
Erro	k=3, θ=1		0,164141	0,095092	0,167665	0,163551	0,090491	0,107362	0,093750	0,197802	0,175781	0,131696	0,085890	0,136860
	k=5, θ=2		0,108586	0,081288	0,124251	0,130841	0,084356	0,101227	0,083807	0,153846	0,132813	0,100446	0,076687	0,108998
	k=7, θ=3		0,111111	0,082822	0,116766	0,123832	0,088957	0,098160	0,086648	0,140110	0,113281	0,100446	0,076687	0,100339
	k=9, θ=4		0,111111	0,088957	0,118263	0,114486	0,088957	0,110429	0,092330	0,134615	0,117188	0,104911	0,072086	0,095821
	k=11, θ=5		0,113636	0,090491	0,118263	0,119159	0,092025	0,111963	0,095170	0,129121	0,117188	0,116071	0,082822	0,092620
	k=13, θ=6		0,126263	0,095092	0,118263	0,119159	0,093558	0,116564	0,099432	0,129121	0,128906	0,125000	0,087423	0,089420
	k=15, θ=7		0,128788	0,099693	0,116766	0,121495	0,096626	0,121166	0,102273	0,137363	0,140625	0,131696	0,088957	0,085655
	k=17, θ=8		0,128788	0,099693	0,119760	0,121495	0,099693	0,124233	0,100852	0,142857	0,148438	0,136161	0,092025	0,084714
	k=19, θ=9		0,141414	0,099693	0,121257	0,130841	0,104294	0,122699	0,102273	0,145604	0,148438	0,136161	0,090491	0,084902
	k=21, θ=10		0,143939	0,102761	0,122754	0,130841	0,101227	0,122699	0,103693	0,151099	0,152344	0,138393	0,093558	0,084526

Tabela B.18: *R-values* entre as imagens de teste, com as maiores e menores diferenças observadas, e as classes do melhor cenário utilizando entropia

Caso	Diferença	Parâmetros <i>R-value</i>	Classes											
			avião	praia	construção	carro	campo	fogos de artifício		corrida de motos		montanha	pôr do sol	árvore
			30x30 512	5x5 128	5x5 256	25x25 512	5x5 128	30x30 256	25x25 256	10x10 256	15x15 512	15x15 256	5x5 128	5x5 256
Acerto	Máxima 0,293234 fogos_artificio_36	k=3, $\theta=1$	0,144172	0,164773	0,197191	0,254261	0,175325	0,603535	0,308036	0,287088	0,237643	0,310302	0,176339	0,195807
		k=5, $\theta=2$	0,105828	0,120130	0,130736	0,214489	0,119927	0,474747	0,236607	0,250687	0,200570	0,257538	0,104302	0,121044
		k=7, $\theta=3$	0,101227	0,097200	0,089399	0,186080	0,092127	0,452020	0,216518	0,225275	0,175856	0,239950	0,085836	0,095530
		k=9, $\theta=4$	0,096626	0,083807	0,073378	0,178977	0,077516	0,414141	0,212054	0,211538	0,166350	0,234925	0,069805	0,084059
		k=11, $\theta=5$	0,092025	0,071834	0,066851	0,174716	0,065544	0,398990	0,207589	0,187500	0,160646	0,232412	0,062500	0,076741
		k=13, $\theta=6$	0,085890	0,062703	0,060127	0,170455	0,059659	0,361111	0,212054	0,179945	0,155894	0,233668	0,054383	0,068829
		k=15, $\theta=7$	0,088957	0,053369	0,056369	0,167614	0,056412	0,338384	0,205357	0,175824	0,157795	0,238693	0,050528	0,064676
		k=17, $\theta=8$	0,087423	0,048498	0,053402	0,166193	0,050731	0,340909	0,205357	0,168956	0,158745	0,229899	0,043831	0,061313
		k=19, $\theta=9$	0,090491	0,043425	0,050831	0,164773	0,046672	0,333333	0,198661	0,164148	0,162548	0,224874	0,039570	0,058742
	k=21, $\theta=10$	0,087423	0,039773	0,050435	0,161932	0,042411	0,335859	0,198661	0,164835	0,167300	0,226131	0,036932	0,054391	
	Mínima 0,002378 flor_3175	k=3, $\theta=1$	0,317485	0,127638	0,126978	0,353693	0,114448	0,325758	0,558036	0,385302	0,274715	0,479899	0,137175	0,127967
		k=5, $\theta=2$	0,256135	0,041193	0,049446	0,237216	0,034091	0,267677	0,388393	0,204670	0,213878	0,335427	0,036932	0,057358
		k=7, $\theta=3$	0,194785	0,029424	0,046084	0,215909	0,025771	0,247475	0,314732	0,160714	0,193916	0,286432	0,025568	0,050040
		k=9, $\theta=4$	0,202454	0,024959	0,044897	0,194602	0,022930	0,227273	0,292411	0,147665	0,198669	0,266332	0,025365	0,048259
		k=11, $\theta=5$	0,191718	0,023539	0,045095	0,186080	0,022727	0,219697	0,258929	0,142170	0,197719	0,251256	0,024351	0,047666
		k=13, $\theta=6$	0,177914	0,023539	0,045688	0,181818	0,023133	0,224747	0,256696	0,138049	0,199620	0,242462	0,024756	0,047864
		k=15, $\theta=7$	0,171779	0,023539	0,046084	0,167614	0,022930	0,222222	0,254464	0,138736	0,200570	0,242462	0,024756	0,047468
		k=17, $\theta=8$	0,164110	0,023539	0,046282	0,169034	0,023133	0,217172	0,243304	0,135302	0,199620	0,239950	0,024756	0,048259
k=19, $\theta=9$		0,154908	0,023742	0,046875	0,171875	0,023133	0,209596	0,241071	0,135989	0,201521	0,238693	0,025162	0,048457	
k=21, $\theta=10$	0,161043	0,023945	0,047271	0,174716	0,023133	0,212121	0,254464	0,141484	0,201521	0,239950	0,025568	0,048062		
Erro	Máxima 0,633303 praia_4855	k=3, $\theta=1$	0,484663	0,155641	0,162184	0,561080	0,138190	0,436869	0,633929	0,480769	0,390684	0,788945	0,160714	0,143987
		k=5, $\theta=2$	0,384969	0,035106	0,070411	0,484375	0,035714	0,366162	0,542411	0,280907	0,299430	0,624372	0,045049	0,070609
		k=7, $\theta=3$	0,328221	0,027800	0,060127	0,440341	0,029221	0,333333	0,506696	0,232143	0,275665	0,545226	0,033076	0,061116
		k=9, $\theta=4$	0,300613	0,026786	0,054786	0,406250	0,028206	0,310606	0,488839	0,205357	0,267110	0,473618	0,030844	0,057951
		k=11, $\theta=5$	0,282209	0,026177	0,053995	0,409091	0,026583	0,310606	0,450893	0,185440	0,255703	0,447236	0,029627	0,055775
		k=13, $\theta=6$	0,276074	0,025974	0,053204	0,384943	0,026380	0,308081	0,421875	0,175137	0,260456	0,432161	0,028003	0,053995
		k=15, $\theta=7$	0,269939	0,025974	0,052809	0,352273	0,025974	0,308081	0,406250	0,177198	0,257605	0,402010	0,026989	0,052611
		k=17, $\theta=8$	0,256135	0,025974	0,052017	0,335227	0,025974	0,295455	0,390625	0,175824	0,260456	0,386935	0,026583	0,052413
		k=19, $\theta=9$	0,254601	0,025974	0,052017	0,321023	0,025974	0,295455	0,370536	0,175137	0,262357	0,376884	0,026583	0,052413
	k=21, $\theta=10$	0,246933	0,025974	0,051622	0,322443	0,025974	0,282828	0,375000	0,175824	0,269962	0,364322	0,026177	0,052017	
	Mínima 0,001670 corrida_motos_7387	k=3, $\theta=1$	0,256135	0,123782	0,193829	0,365057	0,124594	0,414141	0,470982	0,568681	0,387833	0,570352	0,156250	0,197587
		k=5, $\theta=2$	0,233129	0,049716	0,111155	0,340909	0,050731	0,383838	0,426339	0,361264	0,301331	0,433417	0,048295	0,117089
		k=7, $\theta=3$	0,211656	0,041802	0,092366	0,325284	0,041802	0,371212	0,433036	0,294643	0,271863	0,378141	0,039367	0,096123
		k=9, $\theta=4$	0,219325	0,039976	0,080498	0,316761	0,037135	0,358586	0,437500	0,264423	0,251901	0,325377	0,036120	0,083663
		k=11, $\theta=5$	0,214724	0,036729	0,075158	0,315341	0,035308	0,363636	0,455357	0,252747	0,244297	0,296482	0,034091	0,076147
		k=13, $\theta=6$	0,213190	0,033279	0,072191	0,308239	0,033279	0,363636	0,439732	0,235577	0,235741	0,282663	0,031047	0,071005
		k=15, $\theta=7$	0,211656	0,033482	0,069225	0,295455	0,032670	0,368687	0,444196	0,223214	0,234791	0,272613	0,031047	0,068631
		k=17, $\theta=8$	0,208589	0,034497	0,065665	0,292614	0,032265	0,373737	0,437500	0,225275	0,234791	0,265075	0,030844	0,064873
k=19, $\theta=9$		0,210123	0,033076	0,061907	0,301136	0,031250	0,378788	0,433036	0,219780	0,238593	0,260050	0,030032	0,064082	
k=21, $\theta=10$	0,202454	0,033076	0,059138	0,303977	0,031453	0,381313	0,446429	0,217720	0,232890	0,256281	0,029830	0,060522		