Do Macro ao Micro: Abordagens em Reconhecimento e Segmentação de Objetos em Imagens

Anderson Santos

Do Macro ao Micro: Abordagens em Reconhecimento e Segmentação de Objetos em Imagens[1]

Anderson Santos

Orientador: *Profº Drº Wesley Nunes Gonçalves*
Coorientador: *Profº Drº José Marcato Junior*

Tese apresentada à Faculdade de Computação - Facom-UFMS como parte dos requisitos necessários à obtenção do título de Doutor em Ciência da Computação.

UFMS - Campo Grande
Agosto/2025

[1] O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Código de Financiamento 001

A todos que
me acompanharam
nessa jornada

Agradecimentos

Gostaria de agradecer a minha família, pelo apoio financeiro no início desses anos de estudo. Agradeço também aos amigos que, direta ou indiretamente, ajudaram a tornar esta trajetória mais leve.

Ao meu orientador, professor Wesley Nunes Gonçalves, expresso gratidão pelo suporte, direcionamento e acompanhamento ao longo de todo o doutorado. Sua paciência e orientação foram essenciais para a realização deste trabalho.

Aos professores da UFMS e à Faculdade de Computação (FACOM) pela disponibilização da infraestrutura necessária para o desenvolvimento da pesquisa, deixo meus agradecimentos.

Por fim, agradeço à CAPES pela bolsa de doutorado.

Abstract

Detecting small objects in high-resolution images is a significant challenge in computer vision, especially in agricultural scenarios, where the size of insects tends to occupy only a few pixels relative to the entire image. Despite the advances brought by Convolutional Neural Networks (CNNs), the standard use of these architectures presents limitations, as the information associated with small objects tends to be lost due to downsampling and pooling operations. In this context, this thesis proposes strategies to improve the detection and segmentation of small objects. Initially, images are divided into smaller patches, and each patch is used individually to train and validate the models. For inference and prediction on the test set, the patches are overlapped, ensuring that in at least one of them, the object remains uncut. To address the issue of objects that end up being divided among different patches, techniques are presented to filter out-of-pattern predictions, those with low confidence scores, or with redundant and overlapping regions. To allow these techniques to also be applied to segmentation approaches, masks were generated from the original annotations, enabling the evaluation of both detection and segmentation models. Considering the limitations of the traditional Intersection over Union (IoU) metric for small objects, especially due to its sensitivity to minor spatial inaccuracies, this thesis also proposes an alternative metric based on the distance between the centers of the bounding boxes. The experimental results demonstrate that the proposed approaches contribute to the localization of small objects in high-resolution images, showing that both detection and segmentation techniques can be effective, as long as the data are properly processed before and after being input into the model.

Resumo

Detectar objetos pequenos em imagens de alta resolução é um desafio relevante na visão computacional, especialmente em cenários agrícolas, nos quais o tamanho dos insetos tende a ocupar poucos pixels em relação à imagem inteira. Apesar dos avanços proporcionados pelas *Convolutional Neural Networks* (CNNs), o uso padrão dessas arquiteturas apresenta limitações, pois as informações associadas a objetos pequenos tendem a se perder devido às operações de *downsampling* e *pooling*. Considerando esse contexto, esta tese propõe estratégias para aprimorar a detecção e segmentação de objetos pequenos. Inicialmente, as imagens são divididas em recortes menores, e cada recorte é utilizado individualmente para treinar e validar os modelos. Para a inferência e predição no conjunto de teste, os recortes são sobrepostos, garantindo que, em pelo menos um deles, o objeto não seja recortado. Assim, para contornar o problema de objetos que acabam sendo divididos entre diferentes recortes, são apresentadas técnicas para filtrar predições fora do padrão, com baixa pontuação de confiança ou com regiões redundantes e sobrepostas. Para permitir que as técnicas também fossem aplicadas em abordagens de segmentação, foram geradas máscaras a partir das anotações originais, possibilitando a avaliação tanto de modelos detectores quanto segmentadores. Considerando as limitações da métrica tradicional de *Intersection over Union* (IoU) para objetos pequenos, especialmente devido à sensibilidade a pequenas imprecisões espaciais, esta tese também propõe uma métrica alternativa baseada na distância entre os centros das caixas delimitadoras. Os resultados experimentais demonstram que as abordagens propostas contribuem para a localização de objetos pequenos em imagens de alta resolução, mostrando que tanto técnicas de detecção quanto de segmentação podem ser eficazes, desde que os dados sejam processados adequadamente antes e depois de passarem pelo modelo.

Sumário

Lista de Abreviaturas

ADCSPDarknet53 *Advanced Downsampling Cross Stage Partial Darknet-53*

AEFNet *Attention Enhancement and Fusion Network*

AP *Average Precision*

ARPN *Adaptive Region Proposal Network*

AUC *Area Under the Curve*

Cascade R-CNN *Cascade Region-based Convolutional Neural Network*

CIoU *Complete Intersection over Union*

CNN *Convolutional Neural Network*

COWC *Car Overhead with Context*

CBFF-SSD *Context-Based Feature Fusion SSD*

DIOR *DetectIon in Optical Remote sensing images*

DLR-3K *German Aerospace Center 3K Vehicle Dataset*

DOTA *Dataset for Object deTection in Aerial images*

ECA *Efficient Channel Attention*

EESRGAN *Edge-Enhanced Super-Resolution Generative Adversarial Network*

EfficientNet *Efficient Network*

EfficientNetB7 *Efficient Network – Model B7*

EIoU *Efficient Intersection over Union*

Faster R-CNN *Faster Region-based Convolutional Neural Network*

FCOS *Fully Convolutional One-Stage Object Detection*

FEBlock *Feature Enhancement Block*

FPN *Feature Pyramid Networks*

FPS Frames por Segundo

FSD *Feature Skyscraper Detector*

GANs *Generative Adversarial Networks*

GC-YOLO *Ghost Convolution and Centralized Feature Pyramid You Only Look Once*

GHOST *Guided Hybrid Quantization with One-to-One Self-Teaching*

GWHD *Global Wheat Head Detection Dataset*

Grad-CAM *Gradient-weighted Class Activation Mapping*

HRTP-Net *High-Resolution Transformer-embedding Parallel detection Network*

HRRSD *High-Resolution Remote Sensing Detection*

HRSC2016 *High Resolution Ship Collection 2016*

HSSCenterNet *Hierarchical Scale Sensitive CenterNet*

InceptionV3 *Inception Network – Version 3*

IoU *Intersection over Union*

LC-YOLO *Laplace Bottleneck and Cross-Layer Attention Upsampling You Only Look Once*

LIIF *Local Implicit Image Function*

LPSW *Local Perception Swin Transformer*

mAP *mean Average Precision*

MASATI *Maritime SATellite Imagery*

Mask R-CNN *Mask Region-based Convolutional Neural Network*

MCFN *Multi-Component Fusion Network*

MdrlEcf *Model with Deep Reinforcement Learning and Efficient Convolution Feature learning*

MSCCA *Multiscale Context and Enhanced Channel Attention*

NMS *Non-Maximum Suppression*

NWPU VHR-10 *Northwestern Polytechnical University Very High Resolution 10*

OGST *Oil and Gas Storage Tank Dataset*

OIRDS *Overhead Imagery Research Data Set*

PR *Precision-Recallc*

PeleeNet *Pelee Network*

RGB *Red, Green, Blue*

RSOD *Remote Sensing Object Detection*

ResNet101 *Residual Neural Network with 101 layers*

ResNet50 *Residual Neural Network with 50 layers*

RetinaNet *Retina Network*

R^2-CNN *Remote sensing Region-based Convolutional Neural Network*

SCEP *Self-Characteristic Expansion Plate*

SCFPN *Scene-Contextual Feature Pyramid Network*

SODCNN *Small Object Detection Convolutional Neural Network*

SSD *Single Shot MultiBox Detector*

TickIDNet *Tick Identification Network*

UCAS-AOD *University of Chinese Academy of Sciences – Aerial Object Detection*

VANT Veículo Aéreo Não Tripulado

VDNET-RSI *Vehicle Detection Network based on Remote Sensing Images*

VEDAI *VEhicle Detection in Aerial Imagery*

YOLO *You Only Look Once*

YOLOv3 *You Only Look Once version 3*

YOLOv4 *You Only Look Once version 4*

YOLOv5 *You Only Look Once version 5*

YOLOv5s *You Only Look Once version 5 – small*

YOLOv7 *You Only Look Once version 7*

Capítulo 1

Introdução

Detectar objetos em imagens tem se tornado uma tarefa cada vez mais comum nas áreas de visão computacional [34, 69, 74]. Arquiteturas de redes neurais profundas têm desempenhado papel central nesse avanço. Inicialmente, CNNs obtiveram excelentes resultados na detecção de objetos, devido à sua grande capacidade de extrair características das imagens [74, 84]. Mais recentemente, modelos baseados em *Transformers* têm se destacado em diferentes cenários, introduzindo mecanismos de atenção capazes de capturar dependências globais e alcançar um bom desempenho em diversas tarefas de detecção [13, 3].

Embora estas arquiteturas apresentem bons resultados na detecção de objetos grandes e médios em conjuntos de dados tradicionais [69], a identificação de objetos pequenos continua sendo um desafio [91]. Arquiteturas amplamente utilizadas, *Faster Region-based Convolutional Neural Network* (Faster R-CNN) [60], *Single Shot MultiBox Detector* (SSD) [39] e *You Only Look Once* (YOLO) [57], por exemplo, demonstram dificuldades quando aplicadas à detecção de alvos de pequenas dimensões.

Entre os fatores que agravam essa dificuldade, destaca-se a variação de escala. Nem sempre um objeto de interesse está em boas condições na imagem, e problemas como iluminação e oclusão também comprometem a detecção. Além disso, as características de grandes objetos podem ser extraídas mais facilmente, em comparação com as de objetos pequenos, uma vez que estas últimas possuem baixa resolução e são ruidosas [23, 24, 41].

Diversos autores propõem definições específicas para objetos pequenos. A definição relativa considera pequenos objetos cuja largura e altura correspondem a menos de 10% das dimensões da imagem original [93, 18]. Na definição absoluta, objetos pequenos possuem resolução inferior a 32×32 pixels [93, 22].

Com o objetivo de compreender como a detecção de objetos pequenos em imagens tem sido abordada, é fundamental analisar a literatura recente. Nesse contexto, a próxima seção apresenta um panorama dos principais trabalhos correlatos, destacando as estratégias, técnicas e limitações apontadas por diferentes autores.

Além disso, é importante destacar que os Capítulos 2 e 3 desta tese apresentam estudos iniciais que, embora tenham sido conduzidos em contextos diferentes, contribuíram para o amadurecimento metodológico da pesquisa. Esses trabalhos permitiram entender as limitações e definir estratégias para a detecção de objetos pequenos, servindo como etapas importantes que conduziram às propostas apresentadas no Capítulo 4.

1.1 Trabalhos Correlatos

Abordagens de detecção de objetos em imagens com CNNs enfrentam desafios recorrentes relacionados à variação de escala [8, 75, 93], interferência com o fundo da imagem [18, 92, 35] e a perda de informações com operações de *down-sampling* e em camadas mais profundas da rede [35, 90]. Tais limitações tornam-se especialmente críticas na identificação de objetos pequenos em relação ao tamanho da imagem.

Com o desenvolvimento das tecnologias de satélite, bem como de Veículos Aéreos Não Tripulados (VANTs), a detecção de objetos tem se tornado importante em imagens aéreas. Imagens obtidas por estas tecnologias possuem características como amplas áreas de fundo, com uma pequena porção da imagem representando objetos de interesse [14, 74, 82], o que pode fazer com que as informações sejam insuficientes para representar os objetos devido aos seus tamanhos [84, 91].

Além de imagens de sensoriamento remoto [16, 55, 52] e imagens capturadas por drones [40, 37], as dificuldades mencionadas também são evidentes em domínios como detecção de defeitos e peças no geral [12, 80, 61], cenas marítimas [17, 87, 25], detecção de insetos e pragas agrícolas [66, 77], sistemas embarcados [48, 9, 96] e detecção de sinais de trânsito [88, 59, 28]. A diversidade de contextos ressalta a necessidade de soluções robustas e adaptáveis para mitigar as limitações das CNNs em cenários complexos.

Nesse contexto, diversos trabalhos propõem aprimoramentos estruturais em arquiteturas baseadas em CNNs [45], modificações no estágio de geração de propostas de regiões [2], bem como avanços em técnicas de *upsampling* [49], segmentação de imagens [81], métodos de destilação de conhecimento [89], divisão da imagem em pedaços com sobreposição [66] e estratégias para reduzir falsos positivos [1, 52].

Alguns estudos de revisão (*surveys*) apresentam um panorama abrangente sobre a detecção de objetos pequenos, resumindo avanços, desafios e soluções recorrentes nessa área. Por exemplo, uma análise [51] de métodos recentes discute aspectos como definições de objetos pequenos, aprimoramentos arquiteturais em redes convolucionais e *transformers*, técnicas de fusão de características, estratégias de aumento de dados e ajustes específicos para lidar com objetos de baixa resolução. De forma semelhante, uma outra pesquisa [7] aborda os desafios inerentes à detecção de objetos pequenos em larga escala, destacando conjuntos de dados, métricas de avaliação e *benchmarks* padronizados, propondo direções para pesquisas futuras.

Em domínios como sensoriamento remoto óptico, um estudo [29] descreve métodos voltados à detecção de objetos em imagens de alta resolução, com ênfase em aplicações como monitoramento ambiental, inspeção de infraestrutura e vigilância. Este estudo ressalta a importância de adaptar arquiteturas e pré-processamentos para lidar com variações de escala, alta densidade de alvos e interferências de fundo, fatores frequentemente presentes em cenários agrícolas e urbanos. Esses *surveys* resumem o conhecimento existente, mapeando lacunas, oportunidades e oferecem um guia para o desenvolvimento de abordagens relacionadas com detecção de objetos pequenos.

Esta seção está organizada em subseções que destacam as estratégias dos trabalhos relacionados. A Subseção 1.1.1 apresenta de forma resumida estudos sobre detecção de objetos pequenos em imagens de sensoriamento remoto. A Subseção 1.1.2 resume trabalhos relacionados com drones e imagens aéreas. A Subseção 1.1.3 descreve técnicas adotadas para detectar insetos, aracnídeos e pragas agrícolas em plantações.

1.1.1 Objetos pequenos em imagens de sensoriamento remoto

A detecção de objetos pequenos em imagens de sensoriamento remoto é uma tarefa desafiadora, especialmente devido à alta resolução das imagens e a proporção reduzida entre o tamanho dos objetos e a cena.

Métodos de detecção, como *You Only Look Once version 3* (YOLOv3) [58], SSD [39] e Faster R-CNN [60], foram comparados em um estudo [95] cujo objetivo é identificar pequenas aeronaves em imagens do *Google Earth* e do conjunto *Dataset for Object deTection in Aerial images* (DOTA) [76]. Os resultados mostraram que, além de apresentar maior velocidade, YOLOv3 também obteve melhor desempenho médio de detecção comparada às demais arquiteturas.

Para superar limitações de modelos clássicos, diversas abordagens propõem modificações estruturais. Por exemplo, o *Context-Based Feature Fusion SSD* (CBFF-SSD) [31] integra unidades de fusão de características e mapas de detecção para melhorar a identificação de objetos pequenos. Experimentos no conjunto *Northwestern Polytechnical University Very High Resolution 10* (NWPU VHR-10) [6] demonstraram ganhos relevantes de precisão em relação ao SSD tradicional.

Outras abordagens buscam enriquecer a extração de características multiescala. Um exemplo é a aplicação da *Mask Region-based Convolutional Neural Network* (Mask R-CNN) [19] com *Residual Neural Network with 101 layers* (ResNet101) [20] adaptada com *Feature Pyramid Networks* (FPN), que auxilia na detecção objetos em diferentes escalas. A proposta [16] foi avaliada em conjuntos como DOTA e *Remote Sensing Object Detection* (RSOD) [78], demonstrando resultados promissores para detectar classes de “aviões” e “navios”.

A utilização de mecanismos de atenção e fusão de contexto também tem se mostrado eficaz. O modelo *Multiscale Context and Enhanced Channel Attention* (MSCCA) [55] combina o *backbone Pelee Network* (PeleeNet) [73] com blocos *Efficient Channel Attention* (ECA), obtendo $80,4\%$ de *mean Average Precision* (mAP) no DOTA e $94,4\%$ no NWPU VHR-10, equilibrando velocidade de detecção e economia de recursos computacionais.

Além dos mecanismos de atenção, a preservação de resolução apresenta bons resultados na detecção de objetos pequenos em fundos complexos. Nesse contexto, a abordagem *High-Resolution Transformer-embedding Parallel detection Network* (HRTP-Net) [90] propõe módulos que preservam a alta resolução espacial de objetos pequenos e distinguem seus pixels dos do fundo por meio de mecanismos de atenção. Avaliado nos conjuntos *Maritime SATellite Imagery* (MASATI) [15], *VEhicle Detection in Aerial Imagery* (VEDAI) [56] e DOTA, o modelo superou métodos tradicionais.

Limitações computacionais são comuns em dispositivos como satélites e drones. Neste sentido, o modelo *Guided Hybrid Quantization with One-to-One Self-Teaching* (GHOST) [89] utiliza distilação guiada para preservar detalhes importantes e detectar objetos pequenos, diminui os custos computacionais e aumenta a precisão em comparação com métodos tradicionais de quantização. Avaliado nos conjuntos VEDAI, DOTA, NWPU VHR-10 e *DetectIon in Optical Remote sensing images* (DIOR) [30], GHOST se destacou em relação a outros detectores.

No contexto de imagens de grande escala (por exemplo, 20000×20000 pixels), a *Remote sensing Region-based Convolutional Neural Network* (R^2-CNN) [52], baseada em Tiny-Net, se destaca por seu baixo consumo de memória e por apresentar mAP de $96,04\%$. Essa rede treina em conjunto um classificador e um detector, processando pedaços de imagem sobrepostos para reduzir falsos positivos e aumentar a precisão da localização.

Cenários complexos com objetos sobrepostos e fundos confusos requerem soluções com maior sensibilidade contextual. A *Scene-Contextual Feature Pyramid Network* (SCFPN) [4] utiliza normalização por grupo e melhora a detecção de objetos pequenos em múltiplas escalas. O modelo foi avaliado no conjunto de dados DOTA e demonstrou desempenho superior aos métodos de referência nas métricas de $IoU \geq 0.7$.

Propostas ainda mais robustas incluem arquiteturas compostas por múltiplos componentes. A *Multi-Component Fusion Network* (MCFN) [35] combina três blocos distintos, sendo eles, fusão de pirâmides, seleção de regiões baseada em interseção relativa e incorporação de contexto. Essa estrutura melhora significativamente a detecção em cenários complexos, superando Faster R-CNN, YOLOv3 e SSD.

Considerando a baixa resolução ou ruídos em imagens, a *Edge-Enhanced Super-Resolution Generative Adversarial Network* (EESRGAN) [54] utiliza uma abordagem híbrida com *Generative Adversarial Networks* (GANs) para aprimoramento de bordas e super-resolução. Testes nos conjuntos *Car Overhead with Context* (COWC) [50] e *Oil and Gas Storage Tank Dataset* (OGST) [53] indicaram que preservar detalhes estruturais é fundamental para detectar objetos pequenos.

Abordagens recentes exploram o potencial de arquiteturas híbridas. A *Local Perception Swin Transformer* (LPSW) [81] incorpora elementos do *Swin Transformer* [42] com técnicas de atenção espacial para aprimorar a acurácia na segmentação. Com base em conjuntos como DIOR, *High-Resolution Remote Sensing Detection* (HRRSD) [94] e NWPU VHR-10, a abordagem demonstrou uma inferência mais rápida e resultados superiores em segmentação.

Além disso, propostas específicas como a *Hierarchical Scale Sensitive CenterNet* (HSSCenterNet) [21] focam na detecção de embarcações, integrando vetores de direção para prever caixas delimitadoras inclinadas. Já o modelo *Model with Deep Reinforcement Learning and Efficient Convolution Feature learning* (MdrlEcf) [38] incorpora aprendizado por reforço para melhorar a localização e classificação de objetos pequenos, destacando-se na detecção em imagens marítimas e urbanas.

Ainda no contexto de objetos inclinados, algumas técnicas [71, 72] utilizam módulos de rotação de regiões de interesse e razão entre a largura e altura dos objetos para estimar o ângulo de inclinação. Experimentos realizados nos conjuntos NWPU VHR-10, DOTA, *University of Chinese Academy of Sciences – Aerial Object Detection* (UCAS-AOD) [97], *High Resolution Ship Collection 2016* (HRSC2016) [43] e *German Aerospace Center 3K Vehicle Dataset* (DLR-3K) [36] mostraram que as técnicas propostas superam métodos tradicionais de representação de objetos, além de serem mais rápidas e precisas na inferência.

Outra proposta de destaque é a *Vehicle Detection Network based on Remote Sensing Images* (VDNET-RSI) [98], uma rede em duas etapas que combina preservação de bordas por meio do *Local Implicit Image Function* (LIIF), super-resolução, módulos de detecção e atenção. Avaliada no conjunto DIOR, a abordagem superou modelos como *You Only Look Once version 5* (YOLOv5) [26], Faster R-CNN e *Fully Convolutional One-Stage Object Detection* (FCOS) [68], demonstrando potencial para aplicações em sistemas de transporte inteligente.

Essas abordagens refletem a diversidade de estratégias empregadas na detecção de objetos pequenos em imagens de sensoriamento remoto, combinando eficiência computacional, precisão e robustez.

1.1.2 Objetos pequenos em imagens aéreas e de drones

A detecção de objetos pequenos em imagens aéreas representa um desafio significativo na visão computacional, especialmente em contextos com recursos limitados e cenários visuais complexos. Detectores de objetos convencionais são eficazes para alvos de dimensões médias ou grandes, mas apresentam dificuldades quando aplicados à identificação de objetos pequenos. Esta seção resume abordagens propostas para lidar com estas limitações.

A detecção de defeitos em isoladores elétricos, caracterizados como objetos pequenos em fundos complexos, motivou a proposta da *Ghost Convolution and Centralized Feature Pyramid You Only Look Once* (GC-YOLO) [12], uma otimização da YOLOv5. Enquanto convoluções fantasmas extraem características de forma mais eficiente, mecanismos de atenção coordenada destacam regiões relevantes da imagem. Avaliada em um conjunto com 1600 imagens e 5375 anotações, a GC-YOLO superou arquiteturas tradicionais.

Uma extensão [40] do YOLOv5 introduz módulos como *Feature Enhancement Block* (FEBlock), *Self-Characteristic Expansion Plate* (SCEP) e camadas adicionais de detecção para lidar com objetos pequenos em cenários densos e com ruído de fundo. Avaliado no conjunto VisDrone2021 [99], o modelo melhorou significativamente o desempenho, aumentando o $mAP@0.5$ de $42,5\%$ para $54,4\%$ ao utilizar resolução de 1024×1024. Os resultados foram promissores em condições como ruas noturnas e variações de iluminação.

Uma variação [37] da YOLOv3 incorpora blocos residuais modificados e uma estrutura multiescala para previsão em diferentes resoluções. A rede foi treinada com um conjunto de dados que possui 4406 imagens categorizadas por distância e ruído de fundo. Estratégias como classificação prévia dos dados e re-treinamento proporcionaram um mAP de $90,88\%$.

Para lidar com limitações computacionais de dispositivos embarcados, uma proposta introduziu a *Laplace Bottleneck and Cross-Layer Attention Upsampling You Only Look Once* (LC-YOLO) [9]. A arquitetura incorpora módulos que reforçam detalhes nas camadas superficiais por meio de filtros de realce e fundem características rasas e profundas com atenção cruzada em nível de pixel. Avaliado no conjunto UCAS-AOD, o modelo alcançou um $mAP@0.5$ de $94{,}96\%$, superando versões mais robustas da YOLO.

Visando a detecção de objetos pequenos em missões com VANT, uma proposta [96] modificou a arquitetura da *You Only Look Once version 4* (YOLOv4), introduzindo uma nova função de perda e o *backbone Advanced Downsampling Cross Stage Partial Darknet-53* (ADCSPDarknet53). O modelo incorpora técnicas de aumento de dados e um método de classificação baseado em métricas de distância. Avaliado com imagens aéreas de objetos pequenos, o detector alcançou $mAP@0.5$ de $61{,}00\%$ com 77 Frames por Segundo (FPS).

No mesmo contexto, *Small Object Detection Convolutional Neural Network* (SODCNN) [47], uma variação da *You Only Look Once version 7* (YOLOv7) [70], foi proposta com diversas otimizações estruturais. Entre as melhorias estão a remoção do módulo de detecção de objetos grandes, aumento do número de âncoras e substituição da função de perda *Complete Intersection over Union* (CIoU) pela *Efficient Intersection over Union* (EIoU). Avaliado no conjunto VisDrone2019, o modelo alcançou $mAP@0.5$ de $54{,}03\%$ e superou outros modelos da categoria YOLO e *Cascade Region-based Convolutional Neural Network* (Cascade R-CNN).

Módulos de deconvolução, super-resolução e fusão de camadas rasas foram combinados para detectar objetos pequenos. O modelo [46] foi avaliado em conjuntos de dados que incluem imagens de gado e pedestres capturadas por drones, apresentando mAP de $79{,}12\%$ e $Recall$ de $94{,}10\%$, superando detectores tradicionais. O equilíbrio entre desempenho e acurácia mostrou-se adequado para aplicações de vigilância e agricultura de precisão.

Uma abordagem [5] alternativa explorou o uso de duas redes convolucionais para melhorar a detecção de veículos com múltiplas orientações e escalas. A primeira rede gera propostas de regiões orientadas com base em mapas de características hierárquicos, enquanto a segunda realiza a classificação dos objetos. Avaliado nos conjuntos VEDAI e *Overhead Imagery Research Data Set* (OIRDS) [67], o modelo apresentou superioridade quando comparado com arquiteturas tradicionais.

Modelos compactos também têm sido explorados para detectar objetos pequenos quando há restrições de *hardware*. Uma proposta [48] usa camadas pré-treinadas, concatena características de múltiplas escalas e aplica treinamento não supervisionado para extrair representações. A predição é realizada por classificadores leves e um modelo de regressão otimizado, equilibrando precisão, desempenho e baixo custo computacional.

Outra abordagem relevante é a *Attention Enhancement and Fusion Network*

(AEFNet) [17], proposta para detecção de objetos pequenos em cenas marítimas. A arquitetura une o *backbone* Swin-T [42] com módulos de autoatenção, destacando características em fundos complexos, e funde informações entre diferentes escalas para preservar detalhes de alvos pequenos. Avaliada no conjunto TinyPerson [86], a AEFNet mostrou bom desempenho em contextos com objetos pequenos e ruídos ao fundo.

Uma proposta [44] integrou o *CSWin Transformer* ao Mask R-CNN, complementado por um módulo híbrido que incorpora pedaços menores das imagens. Essa abordagem visa reforçar a detecção em múltiplas escalas, preservando detalhes como bordas e cantos, e melhorar a identificação de pequenos objetos sem aumentar a complexidade do modelo. Os resultados mostraram ganhos significicativos, especialmente em objetos pequenos.

Essas abordagens refletem a diversidade de estratégias propostas para a detecção de objetos pequenos em imagens aéreas, combinando eficiência computacional, preservação de detalhes em múltiplas escalas e mecanismos de atenção para lidar com as limitações impostas por alvos de baixa resolução, fundos complexos e restrições operacionais.

1.1.3 Insetos, aracnídeos, pragas agrícolas e plantações em geral

A detecção de insetos em imagens de platações apresenta desafios similares à detecção de objetos pequenos, principalmente devido ao tamanho reduzido das espécies e à semelhança entre indivíduos.

Um estudo [66] divide imagens em pedaços de 800×800 pixels com sobreposição para serem processadas pelo detector YOLOv4. Ao combinar a estratégia com *Efficient Network* (EfficientNet) na etapa de classificação, a precisão obtida foi de 89%. Essa abordagem demonstrou ser eficaz para diferenciar espécies pequenas e semelhantes, como *Phyllotreta striolata* e *Phyllotreta atra*.

Para detectar pragas de pequeno porte, foi desenvolvido o Yolo-Pest [77], com módulos que extraem características em cenários com poucas amostras e uma camada que amplia campos receptivos e reforça canais informativos. Avaliado em imagens de pragas agrícolas, o modelo alcançou $91,9\%$ de $mAP@0.5$, superando o *You Only Look Once version 5 – small* (YOLOv5s) em quase 8% com redução de parâmetros.

Uma abordagem [83] baseada em *Gradient-weighted Class Activation Mapping* (Grad-CAM) foi aplicada na YOLOv5 para detectar espigas de trigo. A arquitetura final remove a camada de larga escala, adiciona uma camada de microescala e reforça a extração de características na escala intermediária. Testes no conjunto *Global Wheat Head Detection Dataset* (GWHD) [10, 11] mostraram aumento da métrica *Average Precision* (AP) para $93,5\%$ em alta resolução com redução de parâmetros.

O reconhecimento de impurezas em grãos de milho também demanda atenção com objetos pequenos. Uma arquitetura [85] integra FasterRCNN com *Efficient Network – Model B7* (EfficientNetB7) para extrair características semânticas de múltiplas escalas e gera caixas delimitadoras com uma *Adaptive Region Proposal Network* (ARPN). O modelo supera as alternativas ResNet101 e EfficientNetB7, destacando-se na detecção de objetos pequenos.

A identificação automatizada de espécies de carrapatos em imagens foi viabilizada por meio do *Tick Identification Network* (TickIDNet) [27]. O modelo foi treinado em um conjunto de imagens com variações em relação a qualidade e tamanho dos objetos. Mesmo obtendo uma boa acurácia, o modelo foi afetado pelo tamanho relativo do carrapato e por características como estágio de vida e status alimentar.

Para diferenciar regiões normais e defeituosas em laranjas-baía, foi proposto o *Feature Skyscraper Detector* (FSD) [65]. A arquitetura utiliza conectividade densa e otimiza a extração de características de objetos pequenos, como manchas pretas, além de distinguir com precisão as extremidades do caule e da flor. Avaliado em um conjunto específico, o modelo superou detectores como YOLOv3 e SSD.

Os estudos analisados resumem a detecção de insetos, pragas e defeitos em cenários agrícolas, destacando os desafios associados à identificação de objetos pequenos, com alta similaridade visual e baixa representatividade nos dados.

1.2 Motivação

Conforme observado nos trabalhos correlatos, a detecção de objetos em imagens tem evoluído significativamente nos últimos anos, com arquiteturas de CNNs e, mais recentemente, modelos baseados em *Transformers*, principalmente em contextos em que os objetos são grandes em relação ao tamanho da imagem, favorecendo a extração de características pelos modelos.

Apesar dos avanços em cenários de maior escala, a detecção de objetos pequenos continua sendo um dos principais desafios da visão computacional, especialmente em contextos com resolução limitada, fundos complexos e com grande densidade de objetos. Embora avanços significativos tenham sido obtidos com o uso de arquiteturas especializadas, mecanismos de atenção e técnicas de que utilizam multiescala, parte dos estudos concentra-se em cenários urbanos, marítimos e de tráfego, com foco em veículos e embarcações.

A detecção de insetos, por sua vez, permanece como um campo menos explorado, ainda que compartilhe diversas dificuldades com os cenários mencionados, como o tamanho reduzido dos objetos e a semelhança visual entre classes. Um pequeno número de abordagens foi proposto para este domínio e algumas delas utilizam variações da família YOLO. Além disso, a falta de estratégias que exploram técnicas de pré e pós-processamento de imagens e refinamento dos resultados obtidos foi observada.

Esses desafios tornam-se ainda mais evidentes em situações em que as imagens são redimensionadas para serem processadas por CNNs, fazendo com que informações de objetos pequenos desapareçam [79]. Mesmo quando permanecem visíveis, à medida que as imagens passam por sucessivas convoluções e seus mapas de características se tornam menores, as informações associadas a esses objetos tendem a se perder ainda mais [14, 34, 64].

Além dos desafios mencionados, os conjuntos de dados bem conhecidos usados para o treinamento das CNNs, como ImageNet[1], MS COCO[2] e PASCAL VOC[3], consistem de imagens obtidas em visão de frente e com uma certa proximidade dos objetos. Desta maneira, as arquiteturas de CNNs desenvolvidas para realizar a detecção de objetos é mais apropriada para as características destes conjuntos de dados [34].

Outro problema que ocorre com a detecção de objetos pequenos é a falta de dados, pois a maioria dos conjuntos de dados possuem anotações em objetos de grande ou média escala. Para estes problemas específicos, os algoritmos de detecção de objetos, como as CNNs, podem não ser capazes de proporcionar bons resultados [69].

Os trabalhos correlatos mostraram que a estratégia de recortar as imagens auxilia na detecção de objetos pequenos, mas não há padronização quanto à sua aplicação. Alguns estudos adotam sobreposição entre as regiões recortadas, enquanto outros não utilizam esse recurso. Outro ponto pouco abordado é o procedimento para reunir novamente objetos que acabam sendo divididos durante o processo de recorte da imagem. Isso evidencia que as soluções atuais são, em geral, desenvolvidas para problemas bastante específicos, dificultando sua generalização para outros cenários.

Alguns experimentos realizados ao longo desta tese buscaram aplicar técnicas de detecção de objetos em cenários nos quais os alvos são visualizados de cima. O primeiro estudo desenvolvido, detalhado no Capítulo 2, avaliou a capacidade de diferentes arquiteturas de CNNs em detectar árvores da espécie Cumbaru em imagens aéreas adquiridas por VANTs.

No segundo trabalho, apresentado no Capítulo 3, investigou-se a aplicação de CNNs para a detecção de bueiros e poços de visita em imagens capturadas de ruas na cidade de Campo Grande-MS. Um dos principais desafios identifi-

[1] Disponível em: https://www.image-net.org/
[2] Disponível em: https://cocodataset.org/
[3] Disponível em: http://host.robots.ox.ac.uk/pascal/VOC/

cados foi a distância entre os objetos de interesse e a câmera, resultando em objetos extremamente pequenos nas imagens. Para mitigar esse problema, adotou-se a estratégia de recortar e utilizar apenas a parte inferior das imagens, descartando regiões com objetos menores e pouco visíveis.

Em muitos cenários, objetos pequenos são os alvos principais e descartar regiões pode levar a perda de informações importantes. Diante desse desafio, o terceiro artigo, apresentado no Capítulo 4, detecta insetos em imagens de folhas de soja. Vistos de cima, os insetos apresentam dimensões reduzidas em relação ao tamanho total da imagem.

Para lidar com essa dificuldade, foi adotada a estratégia de recortar as imagens em regiões menores com sobreposição, garantindo que a análise de todas as áreas da imagem e que nenhum objeto de interesse fosse descartado. Além disso, o trabalho final compara abordagens baseadas em detectores e segmentadores, avaliando o desempenho de ambos os tipos de modelos na identificação dos alvos.

1.3 Objetivos

O objetivo geral deste trabalho foi propor técnicas para detectar objetos pequenos em imagens de alta resolução, explorando abordagens de detecção e segmentação, especialmente em domínios pouco explorados, como a identificação de insetos. Para alcançar o objetivo geral, foram definidos e concluídos os seguintes objetivos específicos:

- Anotar um conjunto de dados com imagens de insetos, que originalmente foi desenvolvido para classificação de superpixels, contribuindo para pesquisas em detecção e segmentação de objetos pequenos;

- Desenvolver e padronizar técnicas de pré-processamento, incluindo métodos de recorte de imagens com sobreposição e conversão de caixas delimitadoras em máscaras de segmentação;

- Propor e implementar estratégias de pós-processamento, incluindo união de detecções que foram recortadas no pré-processamento e alternativas para contornar as limitações da métrica IoU.

- Avaliar e comparar abordagens de detecção e segmentação para a identificação de objetos pequenos, analisando vantagens e limitações dos métodos originais e das técnicas de pré e pós-processamento propostas neste trabalho.

1.4 Estrutura do texto

Esta tese está organizada em cinco capítulos. O presente capítulo introduziu os problemas enfrentados ao detectar objetos pequenos em imagens de alta resolução, resumiu trabalhos correlatos, destacou as lacunas encontradas e apresentou os objetivos deste estudo.

A tese é composta por uma coleção de artigos, inclusos nos Capítulos 2, 3 e 4, e por um capítulo de conclusões (5). No primeiro artigo (Capítulo 2), diversas arquiteturas de CNNs são avaliadas para detectar árvores da espécie cumbaru em imagens obtidas por VANTs. No segundo artigo (Capítulo 3), bueiros e poços de visita são detectados em imagens, e uma abordagem de recortar metade da imagem é avaliada para melhorar a detecção de objetos pequenos. Por fim, o terceiro artigo (Capítulo 4) apresenta uma abordagem de diversos recortes com sobreposição para encontrar insetos em folhas de soja.

Além dos artigos publicados, há um capítulo de conclusões (5), no qual são discutidas as principais contribuições, limitações e perspectivas para pesquisas futuras na identificação de objetos pequenos em imagens de alta resolução.

CAPÍTULO

2

Assessment of CNN-Based Methods for Individual Tree Detection on Images Captured by RGB Cameras Attached to UAVs

Este capítulo apresenta um estudo inicial da pesquisa, desenvolvido sob a forma de um artigo [63] publicado na revista *Sensors*[1], o qual avalia o desempenho de diferentes arquiteturas de CNNs para detectar árvores da espécie cumbaru em imagens *Red, Green, Blue* (RGB) capturadas por VANTs. Embora o foco deste trabalho não esteja diretamente na detecção de objetos pequenos, sua realização foi fundamental para o amadurecimento metodológico da tese, permitindo conhecer limitações, pensar em estratégias e avaliar arquiteturas que posteriormente auxiliaram nas investigações em objetos pequenos.

Para este trabalho, foram consideradas três arquiteturas de CNNs, sendo elas, YOLOv3 [58], Faster R-CNN [60] e *Retina Network* (RetinaNet) [33], as duas últimas usando *Residual Neural Network with 50 layers* (ResNet50) [20] como *backbone*. O conjunto de dados era composto por 392 imagens, com resolução de 5472×3648 pixels e capturadas em diferentes épocas do ano. Cada imagem possuía cerca de uma amostra de cumbaru, a qual foi anotada por um especialista.

Até o presente momento, o artigo contabiliza 196 citações pelo *Google Scholar*[2], mostrando a importância deste trabalho.

[1] Disponível em: https://www.mdpi.com/journal/sensors

[2] Disponível em: https://scholar.google.com/

sensors

Article

Assessment of CNN-Based Methods for Individual Tree Detection on Images Captured by RGB Cameras Attached to UAVs

Anderson Aparecido dos Santos [1], José Marcato Junior [2,*], Márcio Santos Araújo [2], David Robledo Di Martini [2], Everton Castelão Tetila [3], Henrique Lopes Siqueira [2], Camila Aoki [4], Anette Eltner [5], Edson Takashi Matsubara [1], Hemerson Pistori [1,3], Raul Queiroz Feitosa [6], Veraldo Liesenberg [7] and Wesley Nunes Gonçalves [1,*]

[1] Faculty of Computer Science, Federal University of Mato Grosso do Sul, Campo Grande 79070-900, Brazil
[2] Faculty of Engineering, Architecture and Urbanism and Geography, Federal University of Mato Grosso do Sul, Campo Grande 79070-900, Brazil
[3] Department of Computer Engineering, Dom Bosco Catholic University, Campo Grande 79117-900, Brazil
[4] CPAQ, Federal University of Mato Grosso do Sul, Aquidauana 79200-000, Brazil
[5] Institute of Photogrammetry and Remote Sensing, Technische Universität Dresden, 01062 Dresden, Germany
[6] Department of Electrical Engineering, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro 22451-900, Brazil
[7] Department of Forest Engineering, Santa Catarina State University, Lages 88520-000, Brazil
* Correspondence: jrmarcato@gmail.com (J.M.J.); wesley.goncalves@ufms.br (W.N.G.)

Received: 11 June 2019; Accepted: 19 July 2019; Published: 18 August 2019

Abstract: Detection and classification of tree species from remote sensing data were performed using mainly multispectral and hyperspectral images and Light Detection And Ranging (LiDAR) data. Despite the comparatively lower cost and higher spatial resolution, few studies focused on images captured by Red-Green-Blue (RGB) sensors. Besides, the recent years have witnessed an impressive progress of deep learning methods for object detection. Motivated by this scenario, we proposed and evaluated the usage of Convolutional Neural Network (CNN)-based methods combined with Unmanned Aerial Vehicle (UAV) high spatial resolution RGB imagery for the detection of law protected tree species. Three state-of-the-art object detection methods were evaluated: Faster Region-based Convolutional Neural Network (Faster R-CNN), YOLOv3 and RetinaNet. A dataset was built to assess the selected methods, comprising 392 RBG images captured from August 2018 to February 2019, over a forested urban area in midwest Brazil. The target object is an important tree species threatened by extinction known as *Dipteryx alata* Vogel (Fabaceae). The experimental analysis delivered average precision around 92% with an associated processing times below 30 miliseconds.

Keywords: object-detection; deep learning; remote sensing

1. Introduction

Preservation of sensitive tree species requires timely and accurate information on their distribution in the area under threat. Remote sensing techniques have been increasingly applied as alternatives to costly and time consuming field surveys for assessing forest resources. For this purpose, satellite, aerial and, more recently, Unmanned Aerial Vehicle (UAV) have been the most common platforms used for data collection.

Multispectral [1–5] and hyperspectral [6,7] imageries, Light Detection And Ranging (LiDAR) data [8–11], and also combinations of them [12–17] have been the preferred data source. Clark et al. [6] used airborne hyperspectral data (161 bands, 437–2434 nm) for the classification of seven tree species.

Linear discriminant analysis (LDA), maximum likelihood (ML) and spectral angle mapper (SAM) classifiers were tested. The authors reported accuracy of 88% with a ML classifier based on 60 bands. Dalponte et al. [7] investigated the use of hyperspectral sensors for the classification of tree species in a boreal forest. Accuracy around 80% was achieved, using Support Vector Machines (SVM) and Random Forest (RF) classifiers.

Immitzer et al. [3] applied RF to classify 10 tree species in an Austrian temperate forest upon WorldView-2 (8 spectral bands) multispectral data, having achieved an overall classification accuracy around 82%. In a later work, Immitzer et al. [4] used Sentinel-2 multispectral imagery to classify tree species in Germany with a RF classifier achieving accuracy around 65%. Franklin and Ahmed [5] reported 78% accuracy in the classification of deciduous tree species applying object-based and machine learning techniques to UAV multispectral images.

Voss and Sugumaran [12] combined hyperspectral and LiDAR data to classify tree species using an object-oriented approach. Accuracy improvements up to 19% were achieved when both data were combined. Dalponte et al. [15] investigated the combination of hyperspectral and multispectral images with LiDAR for the classification of tree species in Southern Alps. They achieved 76.5% accuracy in experiments using RF and SVM. Nevalainen et al. [18] combined UAV-based photogrammetric point clouds and hyperspectral data for tree detection and classification in boreal forests. RF and Multilayer Perceptron (MLP) provided 95% overall accuracy. Berveglieri et al. [19] developed a method based on multi-temporal Digital Surface Model (DSM) and superpixels for classifying successional stages and their evolution in tropical forest remnants in Brazil.

While numerous studies have been conducted on multispectral, hyperspectral, LiDAR and combinations of them, there are few studies relying on RGB images for tree detection/classification. Feng et al. [20] used RGB images for urban vegetation mapping. They used RF classifiers, and verified that the texture, derived from the RGB images, contributed significantly to improve the classification accuracy. However, tree species classification was not specifically addressed in this work.

In the last few years, approaches based on deep learning, such as Convolutional Neural Networks (CNNs) and their variants, gained popularity in many fields, including remote sensing data analysis. Mizoguchi et al. [11] applied CNN to terrestrial LiDAR data to classify tree species and achieved between 85% and 90% accuracy. Weinstein et al. [21] used semi-supervised deep learning neural networks for individual tree-crown detection in RGB airborne imagery. Barré et al. [22] developed a deep learning system for classifying plant species based on leaf images using CNN.

Regarding plant species classification and diseases detection based on leaf images, several works were developed [23–28]. Fuentes et al. [25] focused on the development of a deep-learning-based detector for real-time tomato plant diseases and pests recognition, considering three CNNs: Faster Region-based Convolutional Neural Network (Faster R-CNN), Region-based Fully Convolutional Network (R-FCN) and Single Shot Multibox Detector (SSD). However, tree detection was not the target application.

To the best of our knowledge, no study focused thus far on state-of-the-art deep learning-based methods for tree detection on images generated by RGB cameras on board of UAVs. The present study addressed this gap and presented an evaluation of deep learning-based methods for individual tree detection on UAV/RGB high resolution imagery. This study focused on a tree species known as *Dipteryx alata* Vogel (Fabaceae), popularly known as baru or cumbaru (henceforth cumbaru), which is threatened by extinction according to the IUCN (2019) (The International Union for Conservation of Nature's Red List of Threatened Species, https://www.iucnredlist.org/species/32984/9741012). This species has a high economic potential and a wide range of applications, mainly for the use of non-timber forest products. It is distributed over a large territory, being mostly associated to the Brazilian Savanna, although it also occurs in the wetlands [29] in midwest Brazil.

Our work hypothesis is that state-of-the-art deep learning-based methods are able to detect single tree species upon high-resolution RGB images with attractive cost, accuracy and computational load. The contribution of this work is twofold. First, we assessed the usage of high-resolution images produced by RGB cameras carried by UAVs for individual trees detection. Second, we compared three state-of-the-art CNN-based object detection methods, namely FasterRCNN, RetinaNet and YOLOv3, for the detection of cumbaru trees on said UAV/RGB imagery.

The rest of this paper is organized as follows. Section 2 presents the materials and methods adopted in this study. Section 3 presents and discusses the results obtained in the experimental analysis. Finally, Section 4 summarizes the main conclusions and points to future directions.

2. Materials and Methods

2.1. Overall Experimental Procedure

The experiments were conducted in four main steps (see Figure 1). First, the images were acquired in different periods of the year by a RGB camera on a UAV platform. The images were annotated by a specialist who delimited the cumbaru trees with a rectangle (bounding box). In the next step, the networks representing each method were trained to detect the cumbaru tree instances in a set of training images. In the third step, the images not used for training were submitted to the trained networks, which predicted the cumbaru tree occurrences, returning the detected bounding boxes. In the final step, the accuracy metrics were computed for each methods on the predicted results.

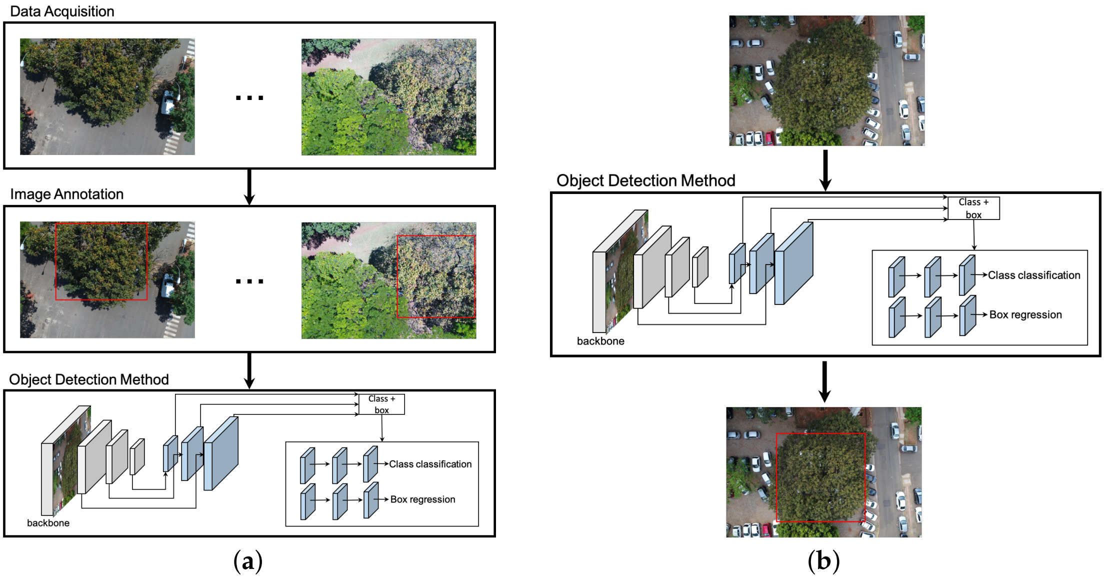


Figure 1. General processing chain: (**a**) UAV images at different seasons were captured and annotated by a specialist. A set of images were selected to train the detection network. (**b**) Once trained, the network was applied to detect cumbaru trees in test images. The object detection method in this figure corresponds to RetinaNet, although other methods (e.g., Faster-RCNN and YOLOv3) can be applied.

2.2. Data Acquisition

In total, 392 UAV images were acquired over seven months (from August 2018 to February 2019 in six missions). An advanced Phantom 4 UAV equipped with a 20-megapixel camera captured the images at flight heights of 20–40 m (see Table 1 for more details). Images with a mean Ground Sample Distance (GSD) of 0.82 cm (centimeter) were acquired in three study regions, with a total area of approximately 150,000.00 square meters, in the urban part of Campo Grande municipality, in the Brazilian state of Mato Grosso do Sul. Approximately 110 trees were imaged during the missions. Some tree samples are shown in Figure 2. Notice the variability in terms of appearance, scale and illumination.

Table 1. Aircraft and flight specifications.

Aircraft	Sensor	Field of View	Nominal Focal Length	Image Size	Mean GSD	Mean Flight Height
Phantom4 Advanced	1″ CMOS	84°	8.8 mm	5472 × 3648 (20 Mp)	0.82 cm	30 m

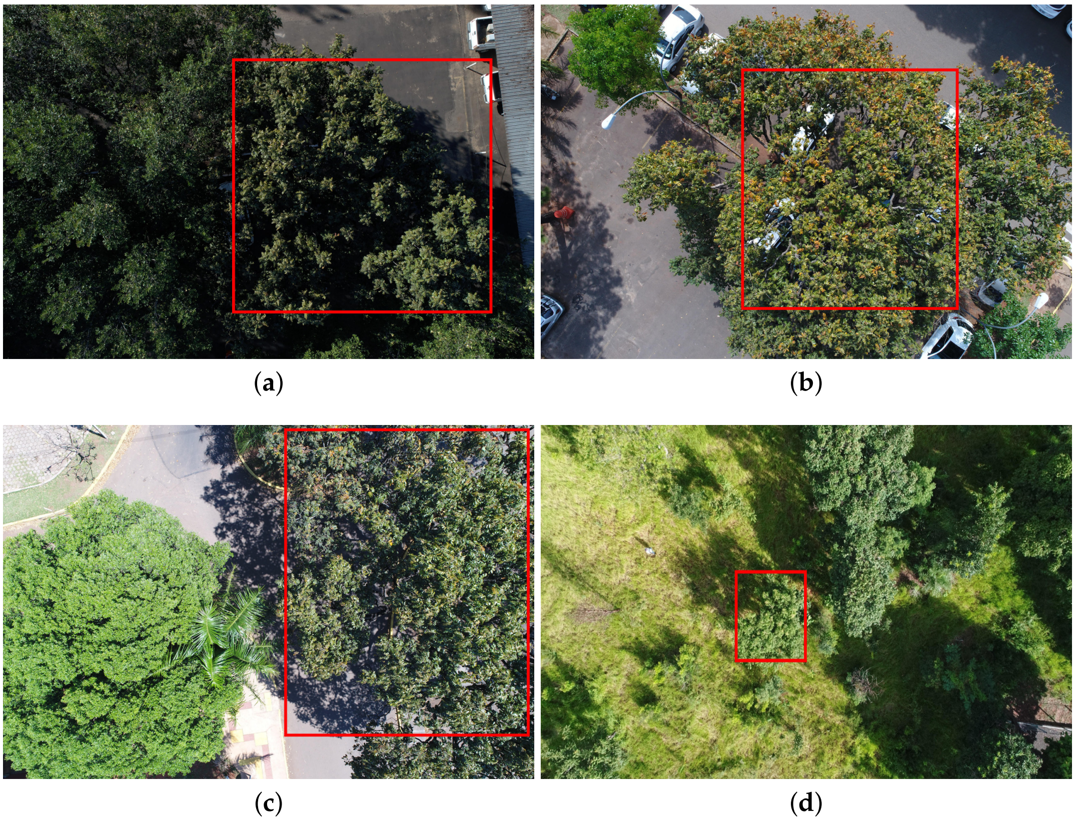


Figure 2. Examples from the dataset: images captured on: (**a**) 26 August 2018; (**b**) 21 September 2018; (**c**) 22 September 2018; and (**d**) 20 February 2019.

Each image was annotated by a specialist using LabelMe software (https://github.com/wkentaro/labelme). In this process, a bounding box specified by the top-left and bottom-right corners was annotated for each cumbaru tree sample in the image.

2.3. Object Detection Methods

The object detection methods compared in this study are briefly described in the following (the following source codes were used as a basis for our implementation: Faster-RCNN, https://github.com/yhenon/keras-frcnn; YOLOv3, https://github.com/qqwweee/keras-yolo3; and RetinaNet, https://github.com/fizyr/keras-retinanet).

- Faster-RCNN [30]: In this method, a feature map is initially produced by a ResNet50 [31]. Given the feature map, Faster-RCNN detects object instances in two stages. The first stage, called Region Proposal Network (RPN), receives the feature map and proposes candidate object bounding boxes. The second stage also accesses the feature map and extracts features from each candidate bounding box using a Region of Interest Pooling (RoIPoolRoIPool) layer. This operation is based on max pooling, and aims to obtain a fixed-size feature map, independent on the size of the candidate bounding box at its input. A softmax layer then predicts the class of the proposed regions as well as the offset values for their bounding boxes.

- YOLOv3 [32]: Unlike Faster-RCNN, which has a stage for region proposal, YOLOv3 addresses the object detection as a problem of direct regression from pixels to bounding box coordinates and class probabilities. The input image is divided into $S \times S$ tiles. For each tile, YOLOv3 predicts bounding boxes using dimension clusters as anchor boxes [33]. For each bounding box, an objectness score is predicted using logistic regression, which indicates the chance of the bounding box to have an object of interest. In addition, C class probabilities are estimated for each bounding box, indicating the classes that it may contain. In our case, each bounding box may contain the cumbaru species or background (uninteresting object). Thus, each prediction in YOLOv3 is composed of four parameters for the bounding box (coordinates), one objectness score and C class probabilities. To improve detection precision, YOLOv3 predicts boxes at three different scales using a similar idea to feature pyramid networks [34]. As a backbone, YOLOv3 uses Darknet-53 as it provides high accuracy and requires fewer operations compared to other architectures.
- RetinaNet [35]: Similar to YOLOv3, RetinaNet is a one-stage object detector but it addresses class imbalance by reducing the loss assigned to well-classified images. Class imbalance occurs when the number of background examples is much larger than examples of the object of interest (cumbaru trees). Using this new loss function, training focuses on hard examples and prevents the large number of background examples from hampering method learning. RetinaNet architecture consists of a backbone and two task-specific subnetworks (see Figure 1b). As the backbone, RetinaNet adopts the Feature Pyramid Network from [34], which is responsible for computing a feature map over an entire input image. The first subnet is responsible for predicting the probability of object's presence at each spatial position. This subnet is a small Fully Convolutional Network (five conv layers) attached to the backbone. The second subnet, which is parallel with the object classification subnet, performs bounding box regression. The design of this subnet is identical to the first one except that it estimates box coordinates for each spatial location at the end.

2.4. Experimental Setup

We adopted in our experiments a five-fold cross validation strategy. Thus, all images were randomly divided into five equal sized sets, called folds. One fold was separated for testing while the remaining four folds were used as training data. This procedure was repeated five times, each time with a different fold selected for testing. Part of the training set was used for validation. Thus, each round (or iteration) of the cross validation procedure was carried out on training, validation, and testing sets comprising 60%, 20% and 20% of the available images, respectively. To reduce the risk of overfitting, we tuned the learning rate and the number of epochs upon the validation set. The weights of the ResNet backbone were initialized with values pre-trained at ImageNet, a procedure known as transfer learning.

The Adam optimizer was used for training all object-detection methods. We set the learning rate to $0.01, 0.001, 0.0001$ and 0.00001. The networks were trained through a number of epochs until the loss stabilized both in the training and in the validation sets. After this tuning procedure, we adopted learning rates equal to 0.0001, 0.001 and 0.0001, and the number of epochs was set to 500, 600, and 250 for Faster-RCNN, YOLOv3, and RetinaNet, respectively.

The performance of each method is reported in the next section by precision–recall curves and the average precision (AP) [36,37]. To estimate the precision and recall, we calculated the Intersection over Union (IoU), which is given by the overlapping area between the predicted and the ground truth bounding boxes divided by the area of union between them. Following well-known competitions in object detection [36,37], a correct detection (True Positive, TP) is considered for IoU ≥ 0.5, and a wrong detection (False Positive, FP) for IoU < 0.5. A False Negative (FN) is assigned when no corresponding ground truth is detected. Given the above metrics, precision and recall are estimated using Equations (1) and (2), respectively.

$$P = \frac{TP}{TP + FP} \tag{1}$$

$$R = \frac{TP}{TP + FN} \tag{2}$$

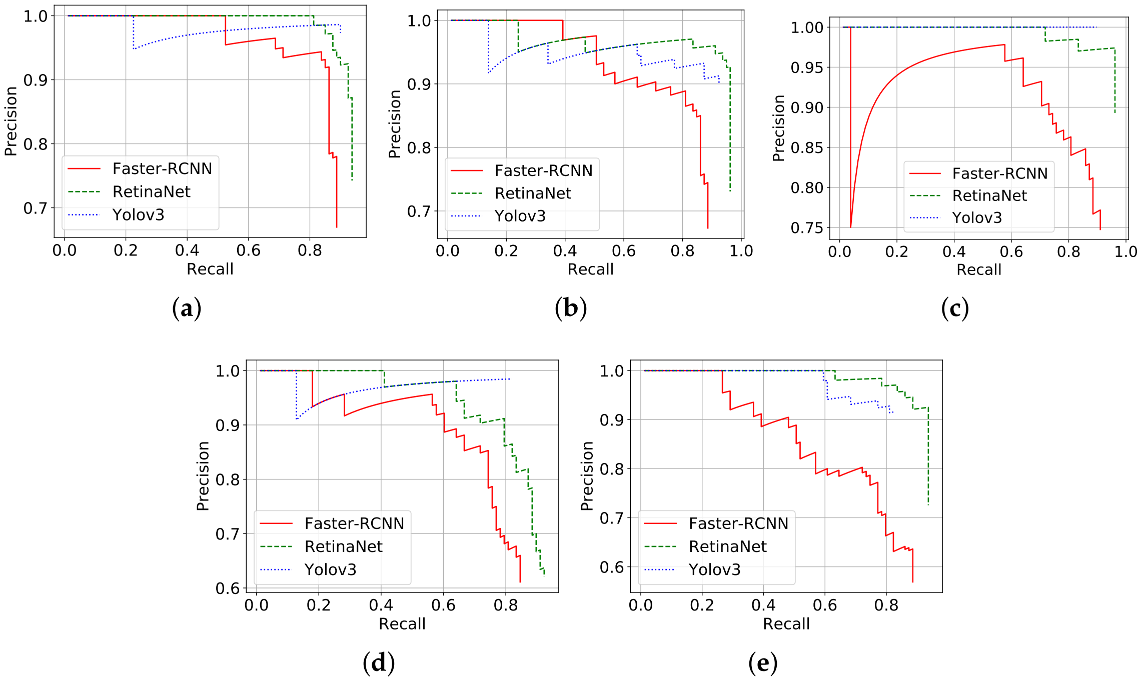
The average precision is given by the area under the precision–recall curve.

3. Results and Discussion

This section presents the results collected in our experiments in three ways. Section 3.1 reports the performance quantitatively in terms of average precision. Section 3.2 presents qualitative results. Finally, we discuss in Section 3.3 the computational costs.

3.1. Precision Results of Three CNN-Based Methods

Figure 3 presents the precision–recall curves of all tested variants for each cross validation round. RetinaNet delivered consistently the highest precision and recall among all tested methods. Despite the comparatively smaller IoUs, Faster-RCNN and YOLOv3 also achieved encouraging results considering the complexity of the problem, as the dataset contains many similar trees.



(a) (b) (c) (d) (e)

Figure 3. Precision–recall curves of detection methods in all five cross validation rounds (**a**–**e**).

The average precision (area under the precision–recall curve) of the detection methods in each cross validation round is shown in Table 2. RetinaNet presented the most accurate results, 92.64% (±2.61%) on average over all five rounds. Actually, RetinaNet consistently achieved the best results on all folds. YOLOv3 and Faster-RCNN came next, with average precision 85.88% (±4.03%) and 82.48% (±3.94%), respectively. In accordance with Figure 3, Table 2 indicates that RetinaNet outperformed its counterparts by about 7%, reaching 92.64% average precision. RetinaNet proposed a new loss function to focus learning on hard negative examples [35]. In this case, training focused on separating the cumbaru from similar trees (hard examples) contributed to greater precision. It is worth emphasizing that the dataset represents a wide variety of environmental conditions, such as flight height and lighting. Thus, the results support the hypothesis that high resolution UAV/RGB images might be a viable approach for detection of individual trees.

Table 2. Average precision (%) for cumbaru tree detection in five cross validation rounds (R1–R5).

Variant	R1	R2	R3	R4	R5	Mean (std)
Faster-RCNN	86.62	84.14	86.13	77.83	77.69	82.48 (±3.94)
YOLOv3	89.08	88.64	89.74	80.99	80.93	85.88 (±4.03)
RetinaNet	93.13	93.92	95.65	87.82	92.66	92.64 (±2.61)

3.2. Detection under Different Conditions

Figure 4 shows detection results in different seasons, as cumbaru trees have different color and overall appearance. The first row of images shows the cumbaru with chestnuts while the second row presents the cumbaru with greener leaves. The images were captured approximately five months apart from each other. In contrast to other detection approaches (e.g., [12]), all tested methods managed to perform well regardless of image acquisition dates. Previous work suggested periods of the year best suited for capturing images (e.g., September [38] and October [39]). Voss and Sugumaran [12] showed that methods trained in images captured in the fall present more consistent results to those trained with images captured in the summer. On the other hand, the methods used in this work do not need to be trained separately in each season and present consistent precision compared to the literature methods.

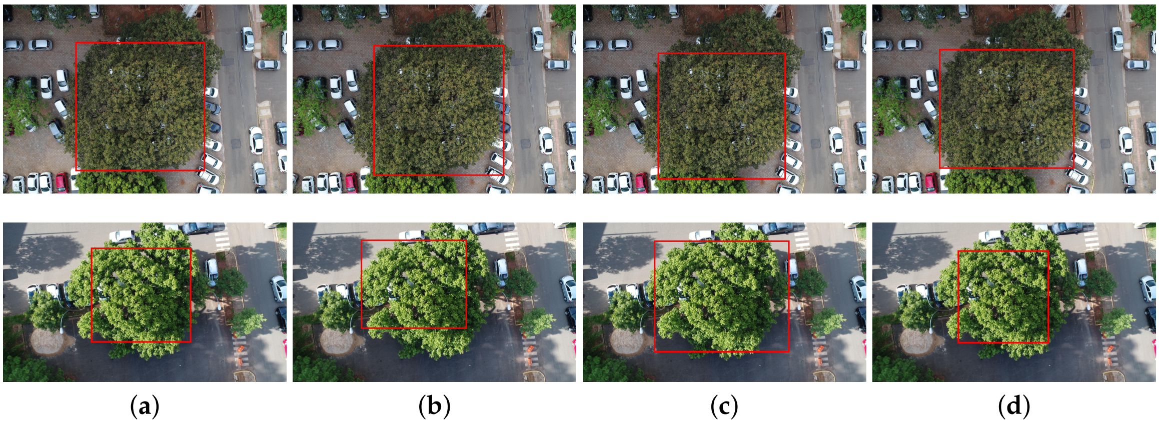


Figure 4. Examples of detection results in images captured in different seasons: (**a**) ground truth; (**b**) Faster-RCNN; (**c**) YOLOv3; and (**d**) RetinaNet.

The methods were able to detect cumbaru trees even on images captured under different lighting and scale conditions, as shown in Figure 5. The first column shows the ground truth while the three columns on the right present the results produced by the three detection methods.

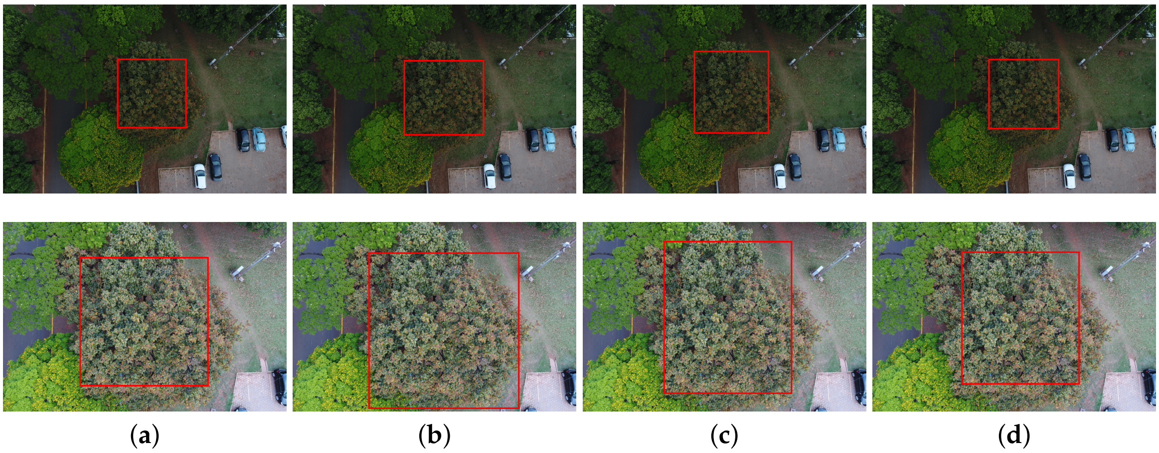


Figure 5. Examples of detection results in images captured for different lighting (average of 67.15 and 130.99 for the brightness channel of the HSB color space) and scale conditions (1:4000 and 1:2500): (**a**) ground truth; (**b**) Faster-RCNN; (**c**) YOLOv3; and (**d**) RetinaNet.

The UAV flight height directly influences the scale of a tree image. Generally, all tested methods were able to handle different scales and flight heights in the range represented in the dataset. Figure 5 illustrates how they behaved under this kind of variation.

Figure 6 shows results of the same tree captured from different view angles.

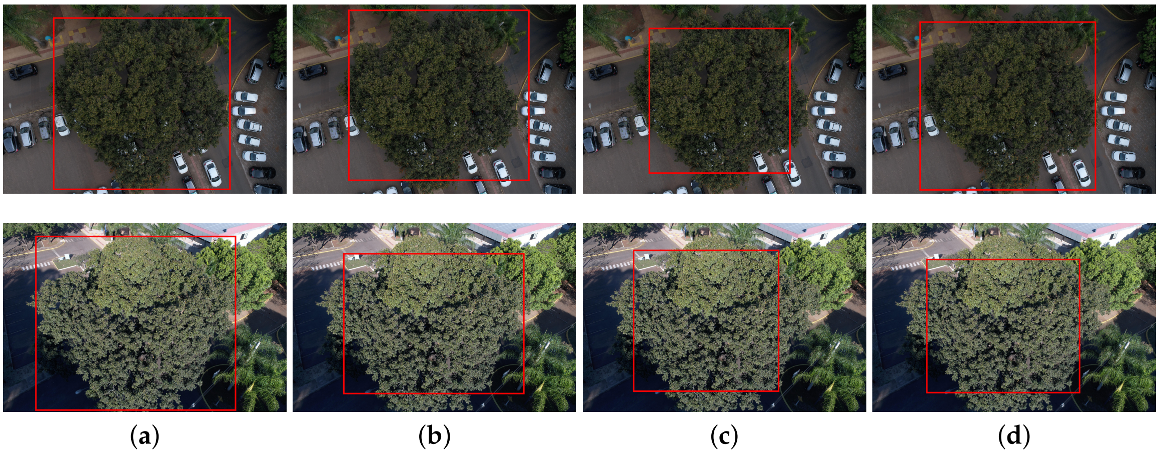


Figure 6. Examples of detection results in images with different capture angles (0° and 30°): (**a**) ground truth; (**b**) Faster-RCNN; (**c**) YOLOv3; and (**d**) RetinaNet.

3.3. Discussion on Computational Complexity

The models were trained and tested on a desktop computer with an Intel(R) Xeon(R) CPU E3-1270@3.80GHz, 64 GB memory, and NVIDIA Titan V graphics card (5120 Compute Unified Device Architecture (CUDA) cores and 12 GB graphics memory). The detection algorithms were coded using Keras-Tensorflow [40] on the Ubuntu 18.04 operating system.

Table 3 shows the mean and standard deviation of the runtime for a tree detection after the image image and trained network model have been loaded.

As expected, the Faster-RCNN variant had the highest computational cost, because it comprises two sequential stages, the first one to propose regions, followed by the second one that classifies the proposed regions. YOLOv3 and RetinaNet were approximately 6.3 and 2.5 times faster than Faster-RCNN, respectively, mainly because they handle object detection as a regression problem.

The results in Table 3 suggest that the methods meet real-time requirements and may be embedded in devices with comparatively low computational capacity.

Table 3. Computational cost of the proposed approach variants. The time is the average in seconds to execute the deep learning methods on an image.

Approach Variation	Time (s)
Faster-RCNN	0.163 (±0.066)
YOLOv3	0.026 (±0.001)
RetinaNet	0.067 (±0.001)

4. Conclusions

In this work, we proposed and evaluated an approach for the detection of tree species based on CNN and high resolution images captured by RGB cameras in an UAV platform. Three state-of-the-art CNN-based methods for object detection were tested: Faster R-CNN, YOLOv3 and RetinaNet. In the experiments carried out on a dataset comprising 392 images, RetinaNet achieved the most accurate results, having delivered 92.64% average precision. Regarding computational cost, YOLOv3 was faster than its counterparts. Faster RCNN was the least accurate and at the same time the most computationally demanding among the assessed detection methods.

The experimental results indicate that RGB cameras attached to UAVs and CNN-based detection algorithms constitute a promising approach towards the development of operational tools for population estimates of tree species, as well for demography monitoring, which is fundamental to integrate economic development and nature conservation. Future works will investigate the application of the proposed techniques considering other tree species. Real-time tree detection using embedded devices will also be investigated.

Author Contributions: Conceptualization, W.N.G. and J.M.J.; methodology, A.A.d.S., W.N.G. and E.T.M.; software, A.A.d.S. and W.N.G.; validation, A.A.d.S., W.N.G. and E.T.M.; formal analysis, A.A.d.S., W.N.G., H.P., E.C.T., M.S.A. and A.E.; investigation, A.A.d.S.; resources, H.L.S. and J.M.J.; data curation, M.S.A., H.L.S. and D.R.D.M.; writing—original draft preparation, A.A.d.S., W.N.G., C.A., R.Q.F. and J.M.J.; writing—review and editing, H.P., A.E., R.Q.F., V.L., E.C.T. and E.T.M.; supervision, W.N.G. and J.M.J.; project administration, W.N.G., H.P. and J.M.J.; and funding acquisition, W.N.G. and J.M.J.

Funding: This research was funded by CNPq (p: 433783/2018-4, 304173/2016-9, 313887/2018-7 and 314902/2018-0), CAPES Print (p: 88881.311850/2018-01) and GRSS (Geoscience and Remote Sensing Society) through the Grand Challenge grant.

Acknowledgments: The authors acknowledge NVIDIA for the donation of a Titan X and a Titan V.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Landenburger, L.; Lawrence, R.L.; Podruzny, S.; Schwartz, C.C. Mapping Regional Distribution of a Single Tree Species: Whitebark Pine in the Greater Yellowstone Ecosystem. *Sensors* **2008**, *8*, 4983–4994. doi:10.3390/s8084983. [CrossRef] [PubMed]
2. Sánchez-Azofeifa, A.; Rivard, B.; Wright, J.; Feng, J.L.; Li, P.; Chong, M.M.; Bohlman, S.A. Estimation of the Distribution of Tabebuia guayacan (Bignoniaceae) Using High-Resolution Remote Sensing Imagery. *Sensors* **2011**, *11*, 3831–3851. doi:10.3390/s110403831. [CrossRef] [PubMed]
3. Immitzer, M.; Atzberger, C.; Koukal, T. Tree Species Classification with Random Forest Using Very High Spatial Resolution 8-Band WorldView-2 Satellite Data. *Remote Sens.* **2012**, *4*, 2661–2693. doi:10.3390/rs4092661. [CrossRef]
4. Immitzer, M.; Vuolo, F.; Atzberger, C. First Experience with Sentinel-2 Data for Crop and Tree Species Classifications in Central Europe. *Remote Sens.* **2016**, *8*. doi:10.3390/rs8030166. [CrossRef]
5. Franklin, S.E.; Ahmed, O.S. Deciduous tree species classification using object-based analysis and machine learning with unmanned aerial vehicle multispectral data. *Int. J. Remote Sens.* **2018**, *39*, 5236–5245. [CrossRef]

6. Clark, M.L.; Roberts, D.A.; Clark, D.B. Hyperspectral discrimination of tropical rain forest tree species at leaf to crown scales. *Remote Sens. Environ.* **2005**, *96*, 375–398. doi:10.1109/TGRS.2012.2216272. [CrossRef]
7. Dalponte, M.; Ørka, H.O.; Gobakken, T.; Gianelle, D.; Næsset, E. Tree Species Classification in Boreal Forests With Hyperspectral Data. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 2632–2645. doi:10.1109/TGRS.2012.2216272. [CrossRef]
8. Heinzel, J.; Koch, B. Exploring full-waveform LiDAR parameters for tree species classification. *Int. J. Appl. Earth Obs. Geoinf.* **2011**, *13*, 152–160. doi:10.1016/j.jag.2010.09.010. [CrossRef]
9. Yao, W.; Krzystek, P.; Heurich, M. Tree species classification and estimation of stem volume and DBH based on single tree extraction by exploiting airborne full-waveform LiDAR data. *Remote Sens. Environ.* **2012**, *123*, 368–380. doi:10.1016/j.rse.2012.03.027. [CrossRef]
10. Garrido, M.; Perez-Ruiz, M.; Valero, C.; Gliever, C.J.; Hanson, B.D.; Slaughter, D.C. Active Optical Sensors for Tree Stem Detection and Classification in Nurseries. *Sensors* **2014**, *14*, 10783–10803. doi:10.3390/s140610783. [CrossRef]
11. Mizoguchi, T.; Ishii, A.; Nakamura, H.; Inoue, T.; Takamatsu, H. Lidar-based individual tree species classification using convolutional neural network. In *Videometrics, Range Imaging, and Applications XIV*; International Society for Optics and Photonics: Bellingham, WA, USA, 2017; Volume 10332. doi:10.1117/12.2270123.
12. Voss, M.; Sugumaran, R. Seasonal Effect on Tree Species Classification in an Urban Environment Using Hyperspectral Data, LiDAR, and an Object-Oriented Approach. *Sensors* **2008**, *8*, 3020–3036. doi:10.3390/s8053020. [CrossRef]
13. Puttonen, E.; Jaakkola, A.; Litkey, P.; Hyyppä, J. Tree Classification with Fused Mobile Laser Scanning and Hyperspectral Data. *Sensors* **2011**, *11*, 5158–5182. doi:10.3390/s110505158. [CrossRef]
14. Naidoo, L.; Cho, M.; Mathieu, R.; Asner, G. Classification of savanna tree species, in the Greater Kruger National Park region, by integrating hyperspectral and LiDAR data in a Random Forest data mining environment. *ISPRS J. Photogramm. Remote Sens.* **2012**, *69*, 167–179. doi:10.1016/j.isprsjprs.2012.03.005. [CrossRef]
15. Dalponte, M.; Bruzzone, L.; Gianelle, D. Tree species classification in the Southern Alps based on the fusion of very high geometrical resolution multispectral/hyperspectral images and LiDAR data. *Remote Sens. Environ.* **2012**, *123*, 258–270. doi:10.1016/j.rse.2012.03.013. [CrossRef]
16. Dalponte, M.; Orka, H.; Ene, L.T.; Gobakken, T.; Naesset, E. Tree crown delineation and tree species classification in boreal forests using hyperspectral and ALS data. *Remote Sens. Environ.* **2014**, *140*, 306–317. doi:10.1016/j.rse.2013.09.006. [CrossRef]
17. Alonzo, M.; Bookhagen, B.; Roberts, D.A. Urban tree species mapping using hyperspectral and lidar data fusion. *Remote Sens. Environ.* **2014**, *148*, 70–83. doi:10.1016/j.rse.2014.03.018. [CrossRef]
18. Nevalainen, O.; Honkavaara, E.; Tuominen, S.; Viljanen, N.; Hakala, T.; Yu, X.; Hyyppä, J.; Saari, H.; Pölönen, I.; Imai, N.N.; et al. Individual Tree Detection and Classification with UAV-Based Photogrammetric Point Clouds and Hyperspectral Imaging. *Remote Sens.* **2017**, *9*. doi:10.3390/rs9030185. [CrossRef]
19. Berveglieri, A.; Imai, N.N.; Tommaselli, A.M.; Casagrande, B.; Honkavaara, E. Successional stages and their evolution in tropical forests using multi-temporal photogrammetric surface models and superpixels. *ISPRS J. Photogramm. Remote Sens.* **2018**, *146*, 548–558. doi:10.1016/j.isprsjprs.2018.11.002. [CrossRef]
20. Feng, Q.; Liu, J.; Gong, J. UAV Remote Sensing for Urban Vegetation Mapping Using Random Forest and Texture Analysis. *Remote Sens.* **2015**, *7*, 1074–1094. doi:10.3390/rs70101074. [CrossRef]
21. Weinstein, B.G.; Marconi, S.; Bohlman, S.; Zare, A.; White, E. Individual Tree-Crown Detection in RGB Imagery Using Semi-Supervised Deep Learning Neural Networks. *Remote Sens.* **2019**, *11*. [CrossRef]
22. Barré, P.; Stover, B.C.; Muller, K.F.; Steinhage, V. LeafNet: A computer vision system for automatic plant species identification. *Ecol. Inform.* **2017**, *40*, 50–56. doi:10.1016/j.ecoinf.2017.05.005. [CrossRef]
23. Lee, S.H.; Chan, C.S.; Wilkin, P.; Remagnino, P. Deep-plant: Plant identification with convolutional neural networks. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015; pp. 452–456. doi:10.1109/ICIP.2015.7350839. [CrossRef]
24. Sladojevic, S.; Arsenovic, M.; Anderla, A.; Culibrk, D.; Stefanovic, D. Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. *Comput. Intell. Neurosci.* **2016**, *2016*, 3289801. doi:10.1155/2016/3289801. [CrossRef]

25. Fuentes, A.; Yoon, S.; Kim, S.C.; Park, D.S. A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. *Sensors* **2017**, *17*. doi:10.3390/s17092022. [CrossRef]
26. Pound, M.P.; Atkinson, J.A.; Townsend, A.J.; Wilson, M.H.; Griffiths, M.; Jackson, A.S.; Bulat, A.; Tzimiropoulos, G.; Wells, D.M.; Murchie, E.H.; et al. Deep machine learning provides state-of-the-art performance in image-based plant phenotyping. *GigaScience* **2017**, *6*. [CrossRef]
27. Lee, S.H.; Chan, C.S.; Mayo, S.J.; Remagnino, P. How deep learning extracts and learns leaf features for plant classification. *Pattern Recognit.* **2017**, *71*, 1–13. doi:10.1016/j.patcog.2017.05.015. [CrossRef]
28. Ferentinos, K.P. Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* **2018**, *145*, 311–318. doi:10.1016/j.compag.2018.01.009. [CrossRef]
29. Sano, S. *Baru: Biologia e Uso*; Documentos; Embrapa Cerrados: Brasília, Brazil, 2004.
30. Ren, S.; He, K.; Girshick, R.B.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2015**, arXiv:1506.01497.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
32. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
33. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. *arXiv* **2016**, arXiv:1612.08242.
34. Lin, T.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. doi:10.1109/CVPR.2017.106. [CrossRef]
35. Lin, T.; Goyal, P.; Girshick, R.B.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *arXiv* **2017**, arXiv:1708.02002.
36. Everingham, M.; Eslami, S.M.A.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
37. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*; Springer International Publishing: Cham, Switzerland, 2014; pp. 740–755. doi:10.1007/978-3-319-10602-1_48. [CrossRef]
38. Sugumaran, R.; Pavuluri, M.K.; Zerr, D. The use of high-resolution imagery for identification of urban climax forest species using traditional and rule-based classification approach. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 1933–1939. [CrossRef]
39. Key, T.; Warner, T.A.; McGraw, J.B.; Fajvan, M.A. A Comparison of Multispectral and Multitemporal Information in High Spatial Resolution Imagery for Classification of Individual Tree Species in a Temperate Hardwood Forest. *Remote. Sens. Environ.* **2001**, *75*, 100–112. doi:10.1016/S0034-4257(00)00159-0. [CrossRef]
40. Chollet, F. Keras. Available online: https://github.com/fchollet/keras (accessed on 22 July 2019).

CAPÍTULO
3

Storm-Drain and Manhole Detection Using the RetinaNet Method

Este capítulo apresenta um segundo estudo preliminar da pesquisa, desenvolvido sob a forma de um artigo [62] publicado na revista *Sensors*[1], no qual foram avaliadas arquiteturas de CNNs para detectar bueiros e poços de visita em ruas de diferentes regiões de Campo Grande-MS.

Assim como no Capítulo 2, o foco principal não estava diretamente na detecção de objetos pequenos, mas os desafios observados, como presença de sombras, objetos cortados pelos limites da imagem e alvos distantes da câmera, foram fundamentais para compreender limitações práticas e elaborar estratégias para a pesquisa em detecção de objetos pequenos.

Os resultados obtidos com a RetinaNet no artigo do capítulo anterior motivaram a avaliação desta arquitetura também neste estudo. Além da RetinaNet, experimentos com Faster R-CNN foram realizados. Em ambas as arquiteturas foram explorados os *backbones* ResNet50 e ResNet101 juntamente com FPN [32].

O conjunto de dados é composto por 297 imagens RGB, com resolução de 1280×720 pixels. Para os experimentos, as imagens foram cortadas em cerca de 50% da altura original, para eliminar a área do céu, que não apresentou relevância para o estudo. Desta maneira, a resolução das imagens utilizadas para treinamento foi de 1280×369 pixels.

O artigo atualmente contabiliza 38 citações de acordo com o *Google Scholar*[2], reforçando a relevância do trabalho.

[1] Disponível em: https://www.mdpi.com/journal/sensors
[2] Disponível em: https://scholar.google.com/

sensors

MDPI

Letter

Storm-Drain and Manhole Detection Using the RetinaNet Method

Anderson Santos [1], José Marcato Junior [2,*], Jonathan de Andrade Silva [1], Rodrigo Pereira [2], Daniel Matos [2], Geazy Menezes [1], Leandro Higa [1], Anette Eltner [3], Ana Paula Ramos [4], Lucas Osco [4] and Wesley Gonçalves [1,2]

[1] Faculty of Computer Science, Federal University of Mato Grosso do Sul, Campo Grande 79070900, MS, Brazil; anderson.asantos3@gmail.com (A.S.); jonathan.andrade@ufms.br (J.d.A.S.); geazyme01@gmail.com (G.M.); leandro.t.higa@gmail.com (L.H.); wesley.goncalves@ufms.br (W.G.)

[2] Faculty of Engineering, Architecture and Urbanism and Geography, Federal University of Mato Grosso do Sul, Campo Grande 79070900, MS, Brazil; rodrigoeamb@gmail.com (R.P.); daniel.matos@ufms.br (D.M.)

[3] Institute of Photogrammetry and Remote Sensing, Technische Universität Dresden, 01062 Dresden, Germany; anette.eltner@tu-dresden.de

[4] Graduate Program of Environment and Regional Development, University of Western São Paulo, Presidente Prudente 19067175, Brazil; anaramos@unoeste.br (A.P.R.); lucasosco@unoeste.br (L.O.)

[*] Correspondence: jose.marcato@ufms.br

Received: 29 June 2020; Accepted: 30 July 2020; Published: 10 August 2020

Abstract: As key-components of the urban-drainage system, storm-drains and manholes are essential to the hydrological modeling of urban basins. Accurately mapping of these objects can help to improve the storm-drain systems for the prevention and mitigation of urban floods. Novel Deep Learning (DL) methods have been proposed to aid the mapping of these urban features. The main aim of this paper is to evaluate the state-of-the-art object detection method RetinaNet to identify storm-drain and manhole in urban areas in street-level RGB images. The experimental assessment was performed using 297 mobile mapping images captured in 2019 in the streets in six regions in Campo Grande city, located in Mato Grosso do Sul state, Brazil. Two configurations of training, validation, and test images were considered. ResNet-50 and ResNet-101 were adopted in the experimental assessment as the two distinct feature extractor networks (i.e., backbones) for the RetinaNet method. The results were compared with the Faster R-CNN method. The results showed a higher detection accuracy when using RetinaNet with ResNet-50. In conclusion, the assessed DL method is adequate to detect storm-drain and manhole from mobile mapping RGB images, outperforming the Faster R-CNN method. The labeled dataset used in this study is available for future research.

Keywords: convolutional neural network; object detection; urban floods mapping

1. Introduction

According to the United Nations Office for Disaster Risk Reduction [1], floods were the most common type of natural disaster in the world for the period 1998–2017, affecting 2 billion people, causing 142,088 deaths and economic losses estimated at $656 billion. In this context, also urban floods need to be considered; according to the World Urbanization Prospects [2], 36.8% of the 633 largest cities in the world are exposed to flood risk, impacting over 660 million inhabitants. An increase in urban flood risks is expected due to climate change, as an intensification of extreme events of precipitation is predicted, potentially leading to a larger water intake into an urban basin [3]. Furthermore, according to [4], changes in land use are another main factor responsible for modifying the hydrological characteristics of urban basins due to the reduction of infiltration capacities and increased runoff. Thus, urbanization leads to increased flood risk because of the impervious surfaces in urban areas [3,5]. Municipalities adopt storm-drain networks to decrease the runoff rate from extreme events and impervious surfaces and thus reduce the impacts by urban floods [6]. One way to assess urban flood risks is to model the drainage system for these watersheds at specific hydrological conditions, and thus adapt the storm-drain network to mitigate the potential damage caused by such floods. It is an essential tool for the planning and management of storm-drain system infrastructures of urban watersheds [7]. Models, such as HEC-1 and Storm Water Management Model (SWMN), evaluate the interation between rainwater and drainage system. Inputs to these models include the size, quantity, and spatial distribution of storm-drains. However, municipal management does not always possess this data, especially in developing countries.

Various remote sensing approaches have been developed to find manholes and storm-drains in urban areas automatically. For instance, [8,9] tested the usage of laser scanning (LiDAR) data. However, when compared to image-based methods, LiDAR data are expensive in terms of equipment and computational costs. Therefore, another focus has been on machine learning algorithms applied to imagery because they can be a useful and robust form to analyze data [10]. These algorithms are widely combined with computer vision techniques to process image data [11,12]. For manhole detection in aerial images, different algorithms were designed with shallow structures [13–17], which need a careful feature extraction method involving pre-processing steps and classification algorithms to achieve good accuracy rates [18,19]. For example, in [15], the authors achieved manhole detection accuracies of 58%. Due to the variety of images datasets (with different illumination conditions, occlusions, noise, and scale), traditional machine learning methods have a low probability of being successful to detect manhole and storm-drain, especially in high dimensionality feature space. More recent, Deep Learning (DL) based-methods have shown higher performances in computer vision tasks because they can extract features while jointly performing classification (end-to-end learning) [18].

DL methods have been successfully used to object detection [20] in several applications, such as agriculture and environmental studies [21,22], urban infrastructure [23] and health analysis [24]. Thus far, solely few works have been developed to detect manholes using DL ([25] and [26]). Reference [25] perform manhole detection in aerial images. However, according to [26], there are two main limitations for using aerial images to detect manholes: (i) The images present resolutions of about 5–10 cm/pixel, which can be insufficient to identify details of the objects, and (ii) manholes can be hidden by trees and vehicles in these images. Therefore, in [26] the authors aimed to detect manhole and storm-drains in images captured from Google Street View API. They demonstrated that street-level imagery can provide useful information to identify obstructed objects, which were not appropriately detected in aerial images.

In this paper, the state-of-the-art DL method RetinaNet was investigated to automatically detect storm-drain and manhole covers in street-level images collected with a car-mounted camera. As an additional contribution, an analyzes of the influence of different feature extractor networks (i.e., backbones) was conducted at the detection accuracy of storm-drain and manhole different from [26], which used a Faster R-CNN architecture (two-stage network) with Resnet 101 as the backbone. The one-stage network RetinaNet was chosen as the network architecture because of its state-of-the-art performance in object detection tasks [27–29]. Furthermore, one-stage methodologies have lower computational processing costs than two-stage approaches [20,30]. One-stage methods typically use the VGG and ResNet as network backbone [31,32], which have shown good results even compared to the DenseNet backbone [23]. ResNet backbones (ResNet-50 and ResNet-101) are used to analyze the effect of their depth on the RetinaNet classification model. Another contribution is to make the labeled dataset publicly available to allow for further DL training in this object detection application. In summary, here are the main contributions:

- The state-of-the-art DL method RetinaNet is investigated to detect Storm-drain and Manhole;
- RetinaNet is compared to Faster R-CNN, which was used for the same purpose in previous research;
- ResNet-50 and ResNet-101 backbones were assessed and;

- The data set is publicly provided for future investigations in https://sites.google.com/view/geomatics-and-computer-vision/home/datasets.

This paper is organized as followed. In Section 2, materials and methods adopted in the study are described. Section 3 presents and discusses the results obtained in the experimental analysis, and Section 4 highlights the main conclusions.

2. Material and Methods

To achieve the aim of this work, initially terrestrial images were acquired in the streets of the Campo Grande city (Section 2.1). The image dataset is described with details in Section 2.2, including the organization in training, validation, and testing sets. The assessed object detection methods are presented in Section 2.3. Finnaly, the assessment metrics are presented in Section 2.4. The procedure steps are the same adopted in our previous work [22].

2.1. Study Area

The images were acquired in the streets of the Campo Grande city, in the state of Mato Grosso do Sul, Brazil (Figure 1). Several damages related to floods occurred in Campo Grande in the previous years, showing a real need for detailed hydrological modeling in its urban area. Accurately mapping storm-drains and manholes is a crucial step to contribute to this modeling. The black lines in Figure 1d highlight the streets considered in our experiments.

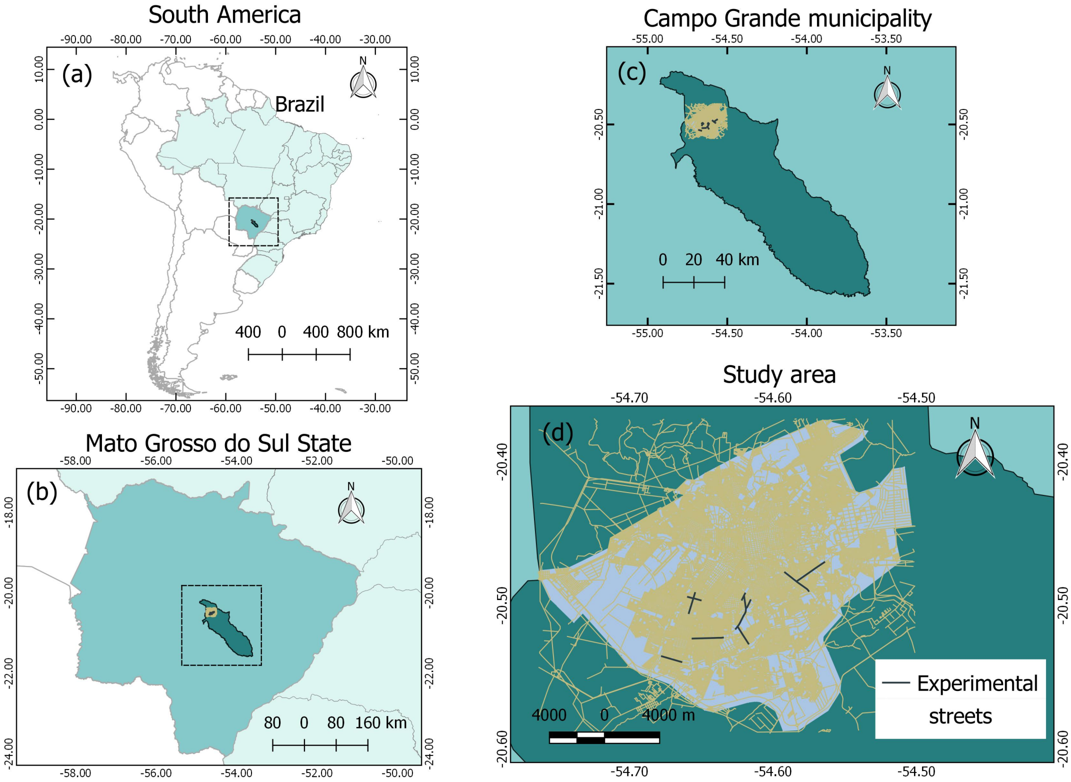


Figure 1. Study area in (**a**) South America and Brazil, (**b**) Mato Grosso do Sul, (**c**) Campo Grande, and (**d**) experimental streets. The black lines represent the streets used in the experiments

2.2. Image Dataset

Storm-drain and manhole samples are presented in Figure 2, showing that the images of the dataset possess variability in terms of appearance, position, scale, and illumination. The dataset is composed of 297 images with resolutions of 1280 × 720 pixels acquired with a GoPro HERO6 Black

RGB camera. This data set contains 166 manhole and 142 storm-drain objects. These images correspond to different regions of Campo Grande city. The images were cropped at 50% of the original width to remove the sky, as done by [26] and [25], resulting in images with resolutions of 1280 × 369 pixels.

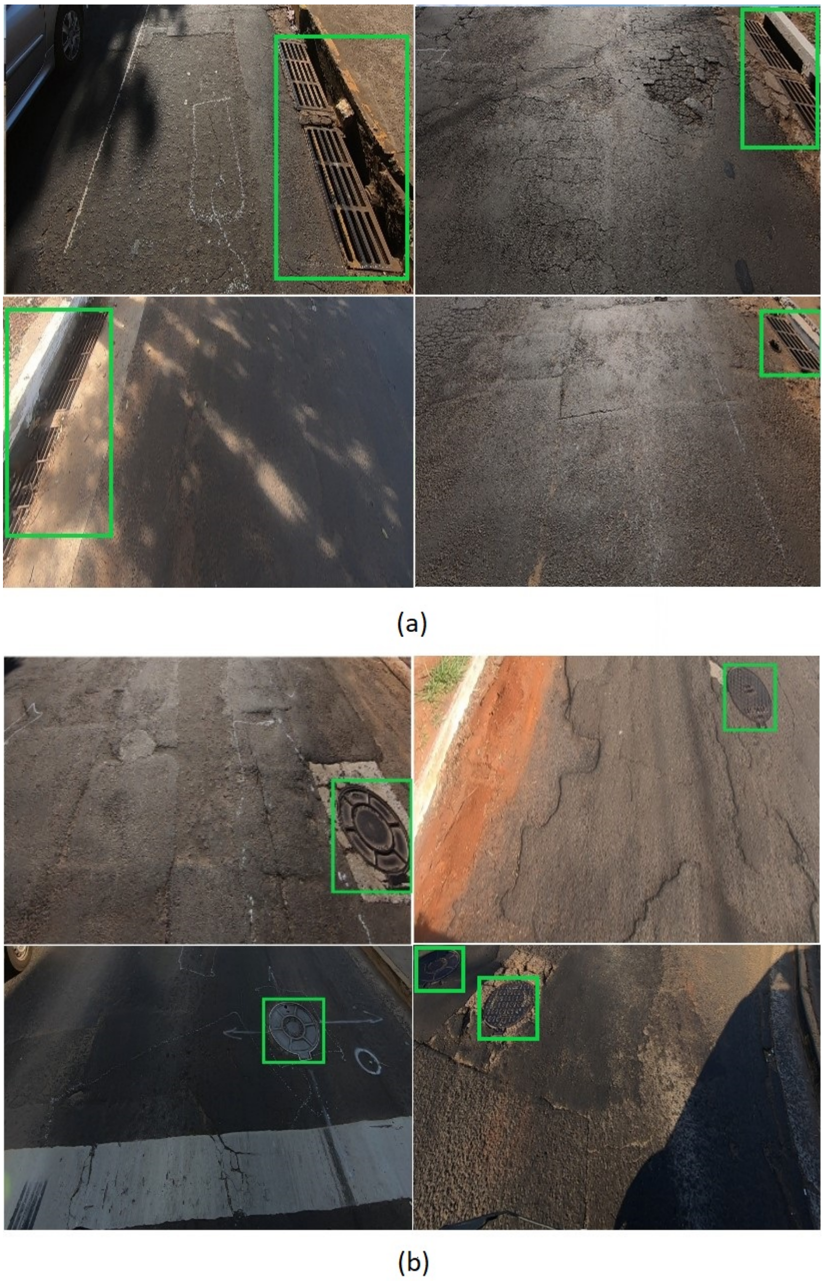


Figure 2. Eight examples of images containing (**a**) storm-drains and (**b**) manhole, both highlighted by green rectangles.

The images were manually annotated by marking the manhole and storm-drains objects with rectangles (bounding boxes) and labeling each rectangle to its corresponding class. Afterward, these images were divided into two groups of training, validation, and testing sets. The first group (named 76-12-12) has 76%, 12%, and 12%, respectively, for training, validation and testing sets. The second group (named 66-15-19) has 66% of training images, 15% of validation images, and 19% of testing images. These two groups were considered to assess the methods not only in one scenario, contributing to a more robust evaluation.

Images for training, validation, and test are from different regions of the city. The idea is to avoid similarity between images from validation and test sets with the training set images to achieve a well generalizing detection model. In Table 1, the main features of our data set summarized.

Table 1. Distribution of the number of (#) images and classes on training, validation and testing data-sets for the division 72-12-12 and 66-15-19.

Division	Set	# Images (%)	# Manholes	# Storm-Drains
76-12-12	Train	226 (76%)	120	113
	Validation	35 (12%)	25	10
	Train + Validation	261 (88%)	145	123
	Test	36 (12%)	21	19
66-15-19	Train	198 (66%)	104	100
	Validation	44 (15%)	25	20
	Train + Validation	226 (81%)	129	120
	Test	55 (19%)	37	22

2.3. Object Detection Method

For this study, the RetinaNet object detection method [33] was adopted. RetinaNet is a one-stage object detection method that considers class imbalance by reducing the loss assigned to images that are well-classified. Class imbalance happens when the number of background examples is larger than the examples of the object of interest, which, in this case, are storm-drains and manholes.

The training step focuses on hard-to-detect examples. RetinaNet architecture is composed of a backbone and two task-specific subnetworks. We adopted the ResNet-50 and ResNet-101 as the backbone and combined it with the Feature Pyramidal Network (FPN) [34], which represents objects at multiple scales that share high and low-level features. Two subnets are applied to the backbone's output to perform the classification and regression tasks.

The models' weights were initialized with weights from the same architecture pre-trained on the MS Coco dataset [35] to reduce the training time. We used the source code available on the Detectron2 toolbox [36] for our implementation. The model was trained and tested on a desktop computer with an Intel(R) Xeon(R) CPU E3-1270@3.80GHz, 64 GB memory, and an NVIDIA Titan V Graphics Card (5120 Compute Unified Device Architecture (CUDA) cores and 12 GB graphics memory) on the Ubuntu 18.04 operating system.

A learning rate of 0.01 was adopted. The number of iterations was set to 10,000 (as set in [25]). Moreover, a batch size of 4 images and 128 regions of interests was chosen for the RetinaNet and Faster R-CNN [37] methods. The results between both methods were compared because previous work on storm-drain and manhole detection [26] considered Faster R-CNN.

2.4. Method Assessment

The performance of RetinaNet was assessed by precision–recall curves and the average precision (AP) as adopted in [22]. To estimate the precision and recall, the Intersection over Union (IoU) was calculated. This metric is given by overlapping the area between the predicted and the ground truth bounding boxes divided by the area of union between them. Following well-known competitions in the object detection scene, a correct detection (True Positive, TP) was also considered for IoU ≥ 0.5, and a wrong detection (False Positive, FP) for IoU < 0.5. A False Negative (FN) is assigned when no corresponding ground truth is detected. Based on the above metrics, precision (P) and recall (R) are estimated using Equations (1) and (2), respectively. The average precision is estimated by the area under the precision–recall curve.

$$P = \frac{TP}{TP + FP} \tag{1}$$

$$R = \frac{TP}{TP + FN} \tag{2}$$

3. Results and Discussions

3.1. Learning Results of the Object Detection Method

The training of the methods was performed with different backbones and the loss curves are shown in Figures 3 and 4 for both groups, 76-12-12 and 66-12-19, respectively. These loss curves indicate that no overfitting occurred because the loss values for training and validation were similar and did not increase. Furthermore, the RetinaNet model converged at approximately 2000 iterations while the Faster R-CNN needed about 8000 iterations until the training loss curve remained flat. This was noted for both proposed divisions of training, validation, and testing sets.

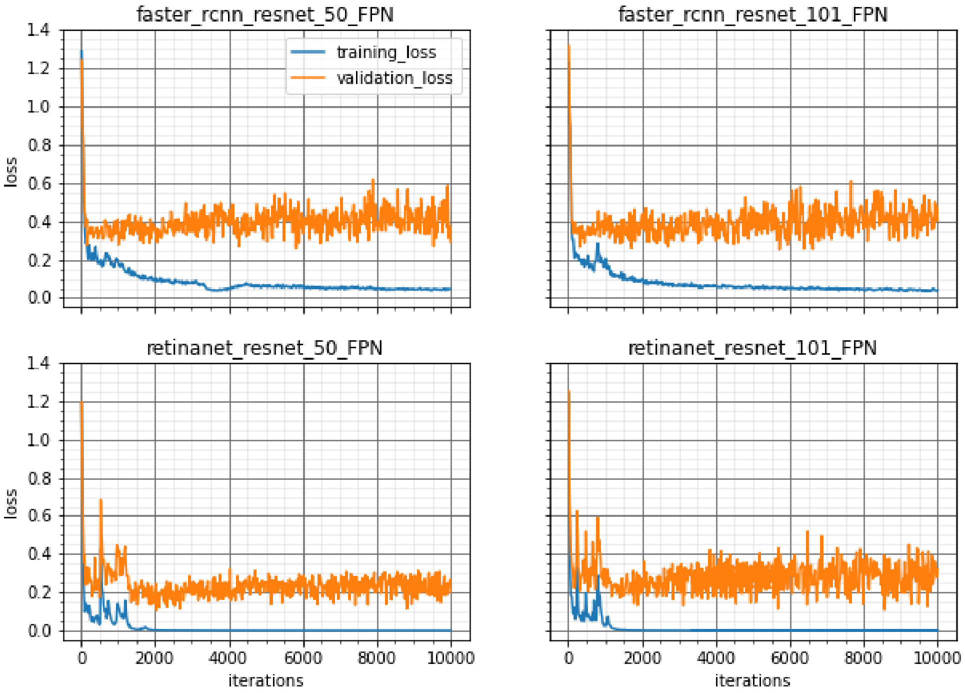


Figure 3. Training and Validation Loss values for all methods to the division 76-12-12 over 10,000 iterations of training model.

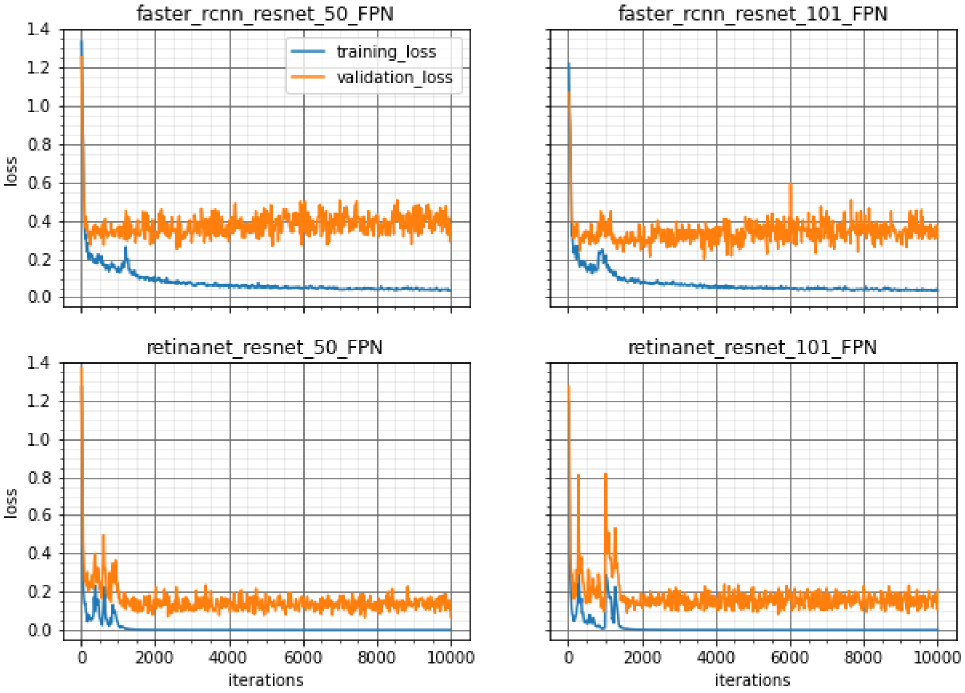


Figure 4. Training and Validation Loss values for all methods to the division 66-15-19 over 10,000 iterations of training model.

3.2. Inference Results of the Object Detection Method

The average precision (AP, %) and its mean values (mAP, %) obtained from the area under the curve are illustrated in Figures 5 and 6 and in Table 2. The results on Table 2 display the IoU cutoff at 0.5 (AP50) and the AP values to each class, manhole ($\mathtt{AP}_{mh}$) and storm-drain ($\mathtt{AP}_{sd}$). The best AP50 values are achieved by RetinaNet, compared to Faster R-CNN, for both datasets division (76-12-12 and 66-12-19). Furthermore, RetinaNet provides the best results for the storm-drain class, which is more challenging to identify when compared to the manhole class.

Table 2. Average precision values to AP50 and to classes manhole (**AP**$_{mh}$) and storm-drain (**AP**$_{sd}$).

Division	**Method**	**Backbone**	**AP50(%)**	**AP$_{mh}$(%)**	**AP$_{sd}$(%)**
76-12-12	Faster-RCNN	ResNet-50	88.30	95.24	71.93
		ResNet-101	86.32	95.24	71.15
	RetinaNet	ResNet-50	92.08	100.00	84.21
		ResNet-101	92.08	95.24	89.47
66-15-19	Faster-RCNN	ResNet-50	88.62	97.22	80.86
		ResNet-101	85.22	96.95	73.85
	RetinaNet	ResNet-50	88.85	94.01	84.42
		ResNet-101	89.69	94.22	85.93

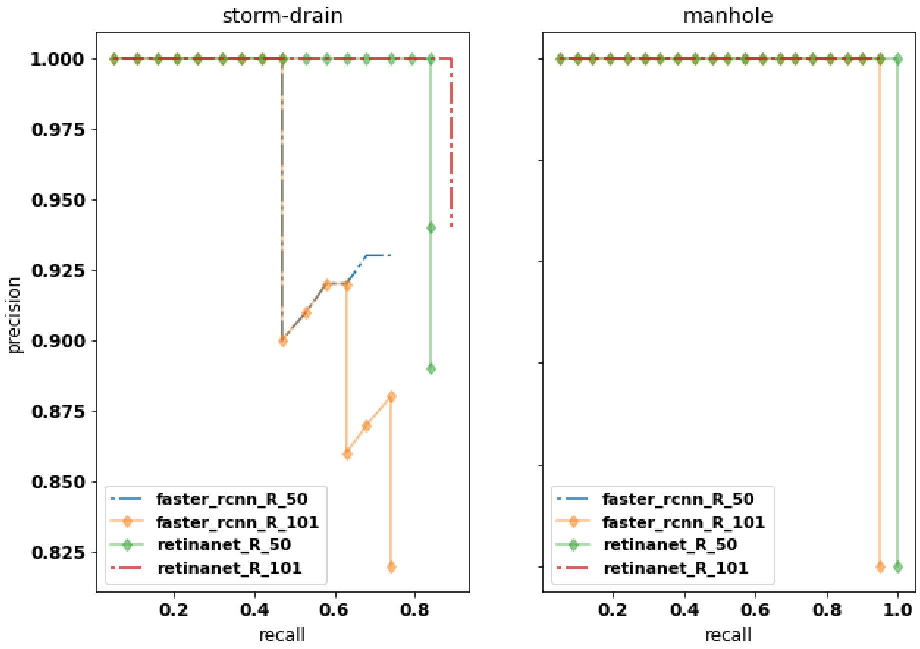


Figure 5. Precision–recall curves for all methods (R_50 and R_101 means ResNet-50 and ResNet-101, respectively) to the division 76-12-12. on IoU threshold at 0.5 (AP50).

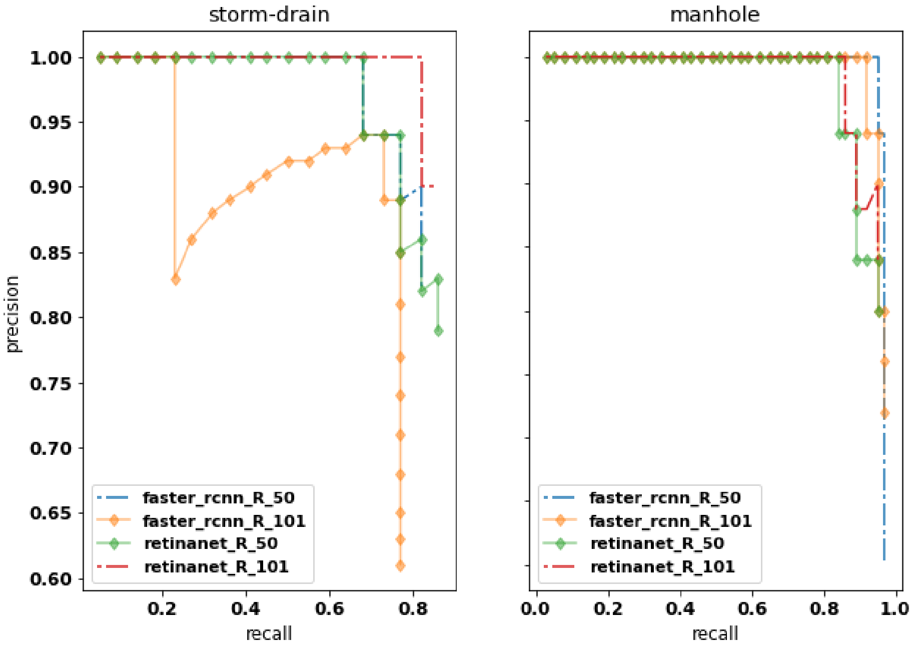


Figure 6. Precision–recall curves for all methods (R_50 and R_101 means ResNet-50 and ResNet-101, respectively) to the division 66-15-19 on IoU threshold at 0.5 (AP50).

Considering the images in Figure 7 it becomes obvious that not all predictions were made correctly by RetinaNet and Faster-RCNN. We found six situations of FNs (false-negative) for the division 76-12-12: Faster-RCNN (ResNet-101) achieved four FNs (Figure 7b–f); Faster-RCNN (ResNet-50) not only achieved the same FNs, but also did not detect the object of interest in Figure 7a; RetinaNet (ResNet50) and RetinaNet (ResNet101) provided only two FNs each one. The objects were not detected in Figure 7b,e when using RetinaNet (ResNet-50), while RetinaNet (ResNet-101) did not detect them in Figure 7c,d. These images were challenging for the trained network due to illumination and noise conditions (Figure 7f). Nevertheless, even in these conditions RetinaNet (ResNet-50) achieved an IoU value of 0.77 with a corresponding confidence (score) value of 0.99.

To examine the importance of our proposed framework, a discussion is presented with a selection of similar studies. A study by [38] achieved an F1-measure score of 0.95 using mobile laser scanning data and a random forest model to identify manholes. The approach, although showing high performance for a shallow learning method, is more expensive regarding data acquisition than RGB data imagery. Another approach by [25] detected manholes in aerial imagery with an accuracy of 99% and a positioning error below 0.7 m. In that study, a Single Shot multi-box Detector (SSD) method was developed and evaluated for images mostly captured from the nadir position. A paper by [25] evaluated different DL networks to detect manholes similar to the current study. However, they utilized aerial images. Their method faced the same conditions as the study by [25]; the high-resolution imagery from the nadir position returned lower accuracies (ranging from 0.67 to 0.89) than our approach. However, it is difficult to compare the results with the performance of our method because they evaluated images from a different point-of-view. The investigated DL-based approach identified hard-to-detect instances with proximal accuracy metrics, in different sizes, point-of-view, and positions, which demonstrates its versatility.

Based on the qualitative and quantitative analysis, RetinaNet outperformed Faster-RCNN, mainly due to more reliable detection in challenging situations. It is important to highlight that the RetinaNet method focuses on hard-to-detect examples in the training task. Furthermore, a higher performance was revealed for manhole detection compared to storm-drain, which confirms the previous work by [26]. Furthermore, only small differences was verified in the results obtained with different backbones. According to [26], results from deep models (like ResNet101) could deteriorate the detection's quality when using aerial images because the last layers of the model are not able to respond to too small objects, as shown in [25]. Thus, street-level images can provide a good alternative to detect manhole and storm-drain objects in images.

Previous work [22,39] showed the potential of RetinaNet in other remote sensing applications, which was also verified in the detection of manhole and storm-drain. However, additional experiments are still necessary to evaluate its effectiveness in other applications.

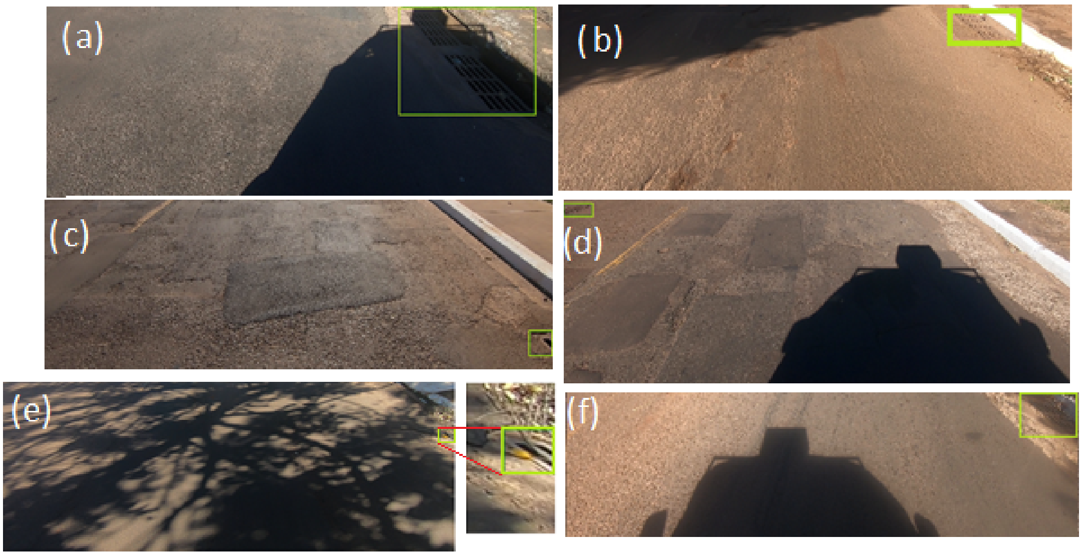


Figure 7. Examples of images that some models did not predict the bounding boxes to the division 76-12-12 considering (**a**) shadow presence, (**b**) small size objects, (**c**) small size objects truncated, (**d**) small size objects, (**e**) small size objects with shadow presence and (**f**) shadow presence.

4. Conclusions

The state-of-art deep network named RetinaNet was investigated to detect storm-drains and manholes in mobile mapping RGB images. RetinaNet was considered with a backbone composed of the ResNet-50 and the Resnet-101 models. THe approach revealed high accuracy in detecting both objects (with mAP higher than 90%). The RetinaNet method was suitable to detect storm-drains in terrestrial RGB imagery, and it outperformed the Faster R-CNN method.

In the future, the trained network will be able to be used to map entire urban catchments with the help of image-based mobile imagery to allow for the incorporation of manhole and storm-drain information into hydrologic and hydraulic modeling to better prevent and mitigate the impact of urban flood events. Other state-of-the-art methods should be proposed and tested to produce a more specific network, which is related to our previous work [21], that can handle this and similar tasks considering point annotation. We provide the labeled dataset used in this study and encourage future research to test the performance of new DL methods with this data. Because of the specific nature of this type of labeled data, it is usually not easily available, and hence it should benefit the training process for focused hydrological work in urban areas.

Author Contributions: Conceptualization, J.M.J., J.d.A.S. and W.G.; methodology, A.S., W.G., J.d.A.S.; software, J.d.A.S., W.G. and A.S.; formal analysis, J.M.J. and L.O.; resources, D.M. and J.M.J.; data curation, A.S., G.M., J.d.A.S.; writing—original draft preparation, J.d.A.S., J.M.J., L.O., A.P.R. and R.P.; writing—review and editing, W.G., L.H. and A.E.; supervision, project administration and funding acquisition, J.M.J. and D.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by CNPq (p: 433783/2018-4, 303559/2019-5 and 304052/2019-1) and CAPES Print (p: 88881.311850/2018-01). The authors acknowledge the support of the UFMS (Federal University of Mato Grosso do Sul) and CAPES (Finance Code 001).

Acknowledgments: The authors would like to acknowledge NVidia© for the donation of the Titan X graphics card used in the experiments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mizutor, M.; Guha-Sapir, D. *Economic Losses, Poverty & Disasters*; Centre for Research on the Epidemiology of Disasters (CRED): Brussels, Belgium; UN Office for Disaster Risk Reduction (UNISDR): Geneva, Switzerland, 2017.
2. Heilig, G.K. *World Urbanization Prospects: The 2011 Revision*; United Nations, Department of Economic and Social Affairs (DESA), Population Division, Population Estimates and Projections Section: New York, NY, USA, 2012.
3. Ahiablame, L.; Shakya, R. Modeling flood reduction effects of low impact development at a watershed scale. *J. Environ. Manag.* **2016**, *171*, 81–91. [CrossRef]
4. Shuster, W.D.; Bonta, J.; Thurston, H.; Warnemuende, E.; Smith, D.R. Impacts of impervious surface on watershed hydrology: A review. *Urban Water J.* **2005**, *2*, 263–275. [CrossRef]
5. Xie, J.; Chen, H.; Liao, Z.; Gu, X.; Zhu, D.; Zhang, J. An integrated assessment of urban flooding mitigation strategies for robust decision making. *Environ. Model. Softw.* **2017**, *95*, 143–155. [CrossRef]
6. Darabi, H.; Choubin, B.; Rahmati, O.; Haghighi, A.T.; Pradhan, B.; Kløve, B. Urban flood risk mapping using the GARP and QUEST models: A comparative study of machine learning techniques. *J. Hydrol.* **2019**, *569*, 142–154. [CrossRef]
7. Habibi, H.; Seo, D.J. Simple and modular integrated modeling of storm-drain network with gridded distributed hydrologic model via grid-rendering of storm-drains for large urban areas. *J. Hydrol.* **2018**, *567*, 637–653. [CrossRef]
8. Yu, Y.; Li, J.; Guan, H.; Wang, C.; Yu, J. Automated Detection of Road Manhole and Sewer Well Covers From Mobile LiDAR Point Clouds. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 1549–1553.
9. Wei, Z.; Yang, M.; Wang, L.; Ma, H.; Chen, X.; Zhong, R. Customized Mobile LiDAR System for Manhole Cover Detection and Identification. *Sensors* **2019**, *19*, 2422. [CrossRef] [PubMed]
10. Mitchell, T.M. *Machine Learning*, 1st ed.; McGraw-Hill, Inc.: New York, NY, USA, 1997.

11. Chaczko, Z.; Yeoh, L.A.; Mahadevan, V. A Preliminary Investigation on Computer Vision for Telemedicine Systems Using OpenCV. In Proceedings of the 2010 Second International Conference on Machine Learning and Computing, Bangalore, India, 9–11 February 2010; IEEE Computer Society: Washington, DC, USA, 2010; pp. 42–46.
12. Marengoni, M.; Stringhini, D. High Level Computer Vision Using OpenCV. In Proceedings of the 2011 24th SIBGRAPI Conference on Graphics, Patterns, and Images Tutorials, Alagoas, Brazil, 28–30 August 2011; pp. 11–24.
13. Timofte, R.; Van Gool, L. Multi-view manhole detection, recognition, and 3D localisation. In Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 6–13 November 2011; pp. 188–195.
14. Niigaki, H.; Shimamura, J.; Morimoto, M. Circular object detection based on separability and uniformity of feature distributions using Bhattacharyya Coefficient. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Tsukuba, Japan, 11–15 November 2012; pp. 2009–2012.
15. Pasquet, J.; Desert, T.; Bartoli, O.; Chaumont, M.; Delenne, C.; Subsol, G.; Derras, M.; Chahinian, N. Detection of manhole covers in high-resolution aerial images of urban areas by combining two methods. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *9*, 1802–1807. [CrossRef]
16. Ali, Z.; Wang, D.; Loya, M. SURF and LA with RGB Vector Space Based Detection and Monitoring of Manholes with an Application to Tri-Rotor UAS Images. *Int. J. Eng. Technol.* **2017**, *9*, 32–39.
17. Moy de Vitry, M.; Schindler, K.; Rieckermann, J.; Leitão, J.P. Sewer Inlet Localization in UAV Image Clouds: Improving Performance with Multiview Detection. *Remote Sens.* **2018**, *10*, 706. [CrossRef]
18. Guo, Y.; Liu, Y.; Oerlemans, A.; Lao, S.; Wu, S.; Lew, M.S. Deep learning for visual understanding: A review. *Neurocomputing* **2016**, *187*, 27–48. [CrossRef]
19. Wang, J.; Ma, Y.; Zhang, L.; Gao, R.X.; Wu, D. Deep learning for smart manufacturing: Methods and applications. *J. Manuf. Syst.* **2018**, *48*, 144–156. [CrossRef]
20. Zhao, Z.; Zheng, P.; Xu, S.; Wu, X. Object Detection With Deep Learning: A Review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [CrossRef] [PubMed]
21. Osco, L.P.; de Arruda, M.D.S.; Junior, J.M.; da Silva, N.B.; Ramos, A.P.M.; Moryia, É.A.S.; Imai, N.N.; Pereira, D.R.; Creste, J.E.; Matsubara, E.T.; et al. A convolutional neural network approach for counting and geolocating citrus-trees in UAV multispectral imagery. *ISPRS J. Photogramm. Remote Sens.* **2020**, *160*, 97–106. [CrossRef]
22. Santos, A.A.D.; Marcato Junior, J.; Araújo, M.S.; Di Martini, D.R.; Tetila, E.C.; Siqueira, H.L.; Aoki, C.; Eltner, A.; Matsubara, E.T.; Pistori, H.; et al. Assessment of CNN-Based Methods for Individual Tree Detection on Images Captured by RGB Cameras Attached to UAVs. *Sensors* **2019**, *19*, 3595. [CrossRef]
23. Ale, L.; Zhang, N.; Li, L. Road Damage Detection Using RetinaNet. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 5197–5200.
24. Guan, Q.; Huang, Y.; Zhong, Z.; Zheng, Z.; Zheng, L.; Yang, Y. Diagnose like a Radiologist: Attention Guided Convolutional Neural Network for Thorax Disease Classification. *arXiv* **2018**, arXiv:1801.09927.
25. Liu, W.; Cheng, D.; Yin, P.; Yang, M.; Li, E.; Xie, M.; Zhang, L. Small Manhole Cover Detection in Remote Sensing Imagery with Deep Convolutional Neural Networks. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 49. [CrossRef]
26. Boller, D.; de Vitry, M.M.; Wegner, J.D.; Leitão, J.P. Automated localization of urban drainage infrastructure from public-access street-level images. *Urban Water J.* **2019**, *16*, 480–493. [CrossRef]
27. Cui, Y.; Oztan, B. Automated firearms detection in cargo x-ray images using RetinaNet. In *Anomaly Detection and Imaging with X-rays (ADIX) IV*; International Society for Optics and Photonics: Bellingham, WA, USA, 2019; Volume 10999, pp. 105–115.
28. Sun, P.; Chen, G.; Guerdan, L.M.; Shang, Y. Salience Biased Loss for Object Detection in Aerial Images. *arXiv* **2018**, arXiv:1810.08103.
29. Sinkevych, O.; Berezhansky, D.; Matchyshyn, Z. On the Development of Object Detector Based on Capsule Neural Networks. In Proceedings of the 2019 XIth International Scientific and Practical Conference on Electronics and Information Technologies (ELIT), Lviv, Ukraine, 16–18 September 2019; pp. 159–162.
30. Jiao, L.; Zhang, F.; Liu, F.; Yang, S.; Li, L.; Feng, Z.; Qu, R. A Survey of Deep Learning-Based Object Detection. *IEEE Access* **2019**, *7*, 128837–128868. [CrossRef]
31. Han, J.; Zhang, D.; Cheng, G.; Liu, N.; Xu, D. Advanced Deep-Learning Techniques for Salient and Category-Specific Object Detection: A Survey. *IEEE Signal Process. Mag.* **2018**, *35*, 84–100. [CrossRef]

32. Luo, R.; Huang, H.; Wu, W. Salient object detection based on backbone enhanced network. *Image Vis. Comput.* **2020**, *95*, 103876. [CrossRef]
33. Lin, T.; Goyal, P.; Girshick, R.B.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *arXiv* **2017**, arXiv:1708.02002.
34. Lin, T.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
35. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 740–755.
36. Wu, Y.; Kirillov, A.; Massa, F.; Lo, W.Y.; Girshick, R. Detectron2. 2019. Available online: https://github.com/facebookresearch/detectron2 (accessed on 4 April 2020).
37. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
38. Yu, Y.; Li, J.; Guan, H.; Wang, C. Automated Extraction of Urban Road Facilities Using Mobile Laser Scanning Data. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2167–2181. [CrossRef]
39. Li, K.; Wan, G.; Cheng, G.; Meng, L.; Han, J. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 296–307. [CrossRef]

CAPÍTULO

4

Finding Small Objects: A Patch-Based and Distance-Based Evaluation Method

Este artigo propõe uma abordagem para a detecção de objetos pequenos em imagens de alta resolução, utilizando como principal aplicação a identificação de insetos em folhas de soja.

Inicialmente, um conjunto de dados anotado com superpixels foi adaptado para oferecer anotações baseadas em caixas delimitadoras, possibilitando sua aplicação tanto em tarefas de detecção quanto de segmentação. A estratégia desenvolvida inclui técnicas de pré-processamento baseadas no recorte sobreposto das imagens, o que ajuda a preservar objetos pequenos durante o treinamento e a inferência dos modelos. Além disso, foram propostas estratégias de pós-processamento, como a adaptação do algoritmo *Non-Maximum Suppression* (NMS) e uma métrica de avaliação baseada na distância entre centros de objetos, buscando superar limitações da métrica IoU tradicional.

Os resultados experimentais demonstram que as técnicas propostas melhoram o desempenho na detecção e segmentação de pequenos objetos, destacando o potencial da abordagem para aplicações em cenários agrícolas e outros domínios com objetos pequenos.

aAcc average Accuracy

ADCSPDarknet53 Advanced Downsampling Cross Stage Partial Darknet-53

AEFNet Attention Enhancement and Fusion Network

AP Average Precision

ARPN Adaptive Region Proposal Network

Cascade R-CNN Cascade Region-based Convolutional Neural Network

CBFF-SSD Context-Based Feature Fusion SSD

CIoU Complete Intersection over Union

COWC Car Overhead with Context

CNN Convolutional Neural Network

DIOR DetectIon in Optical Remote sensing images

DINO DETR with Improved deNoising anchOr boxes

DLR-3K German Aerospace Center 3K Vehicle Dataset

DOTA Dataset for Object deTection in Aerial images

ECA Efficient Channel Attention

EESRGAN Edge-Enhanced Super-Resolution Generative Adversarial Network

EfficientNet Efficient Network

EfficientNetB7 Efficient Network – Model B7

EIoU Efficient Intersection over Union

F F-score

Faster R-CNN Faster Region-based Convolutional Neural Network

FCOS Fully Convolutional One-Stage Object Detection

FEBlock Feature Enhancement Block

FN False Negative

FP False Positive

FPN Feature Pyramid Networks

FPS Frames Per Second

FSD Feature Skyscraper Detector

GAN Generative Adversarial Network

GC-YOLO Ghost Convolution and Centralized Feature Pyramid You Only Look Once

GHOST Guided Hybrid Quantization with One-to-One Self-Teaching

Grad-CAM Gradient-weighted Class Activation Mapping

GUI Graphical User Interface

GWHD Global Wheat Head Detection Dataset

HRRSD High-Resolution Remote Sensing Detection

HRTP-Net High-Resolution Transformer-embedding Parallel detection Network

HSSCenterNet Hierarchical Scale Sensitive CenterNet

HRSC2016 High Resolution Ship Collection 2016

IoU Intersection over Union

LC-YOLO Laplace Bottleneck and Cross-Layer Attention Upsampling You Only Look Once

LIIF Local Implicit Image Function

LPSW Local Perception Swin Transformer

mAcc mean Accuracy

mAP mean Average Precision

Mask R-CNN Mask Region-based Convolutional Neural Network

MASATI Maritime SATellite Imagery

MCFN Multi-Component Fusion Network

MdrlEcf Model with Deep Reinforcement Learning and Efficient Convolution Feature learning

mIoU mean Intersection over Union

MSCCA Multiscale Context and Enhanced Channel Attention

NMS Non-Maximum Suppression

NWPU VHR-10 Northwestern Polytechnical University Very High Resolution 10

OGST Oil and Gas Storage Tank Dataset

OIRDS Overhead Imagery Research Data Set

P Precision

PeleeNet Pelee Network

R Recall

ResNet Residual Neural Network

ResNet101 Residual Neural Network with 101 layers

RetinaNet Retina Network

RSOD Remote Sensing Object Detection

R^2-CNN Remote sensing Region-based Convolutional Neural Network

SCEP Self-Characteristic Expansion Plate

SCFPN Scene-Contextual Feature Pyramid Network

SODCNN Small Object Detection Convolutional Neural Network

SSD Single Shot MultiBox Detector

TickIDNet Tick Identification Network

TP True Positive

UAV Unmanned Aerial Vehicle

UCAS-AOD University of Chinese Academy of Sciences – Aerial Object Detection

VDNET-RSI Vehicle Detection Network based on Remote Sensing Images

VEDAI VEhicle Detection in Aerial Imagery

YOLO You Only Look Once

YOLOv3 You Only Look Once version 3

YOLOv4 You Only Look Once version 4

YOLOv5 You Only Look Once version 5

YOLOv5s You Only Look Once version 5 – small

YOLOv7 You Only Look Once version 7

From Macro to Micro: Approaches to Object Recognition and Segmentation in Images

Anderson Santos[a], José Marcato Junior[b], Wesley Nunes Gonçalves[a,b]

[a] *Faculty of Computer Science, Federal University of Mato Grosso do Sul (UFMS), Cidade Universitária, Av. Costa e Silva, s/nº - Bairro Universitário, Campo Grande, 79070-900, MS, Brazil*

[b] *Faculty of Engineering, Architecture and Urbanism and Geography, Federal University of Mato Grosso do Sul (UFMS), Cidade Universitária, Av. Costa e Silva, s/nº - Bairro Universitário, Campo Grande, 79070-900, MS, Brazil*

Abstract

Detecting small objects in high-resolution images remains a challenging task in computer vision, especially in agricultural scenarios where targets often occupy only a few pixels. This paper proposes an strategy to enhance the detection and segmentation of small objects, focusing on the identification of insects on soybean leaves. A dataset originally annotated with superpixels was adapted to provide bounding box annotations, enabling its use for both detection and segmentation. The methodology introduces preprocessing techniques based on dividing images into overlapping patches, which help preserve small objects during model training and inference. Furthermore, post-processing strategies, including an adaptation of the Non-Maximum Suppression (NMS) algorithm and a distance-based evaluation metric, were developed to overcome the limitations of traditional metrics such as Intersection over Union (IoU) when handling small objects. Experimental results demonstrate that the proposed methods significantly improve the detection and segmentation performance of small insects, as evaluated with models such as DINO and Segformer. The study discusses the strengths and weaknesses of detection and segmentation approaches, providing insights for future applications in different domains.

Keywords: small object detection, high-resolution images, agricultural applications, insect identification, distance-based metric

1. Introduction

Object detection in images has become an increasingly common task in the field of computer vision [1, 2, 3]. Deep neural network architectures have played a central role in this progress. Initially, Convolutional Neural Networks (CNNs) achieved remarkable results in object detection, mainly due to their high capacity for extracting features from images [3, 4]. More recently, Transformer-based models have stood out in different scenarios, introducing attention mechanisms capable of capturing global dependencies and achieving strong performance in several detection tasks [5, 6].

Although CNNs perform well in detecting large and medium-sized objects in conventional datasets [2], identifying small objects remains a challenge [7]. Widely used architectures, such as Faster R-CNN [8], SSD [9], and YOLO [10], for instance, have shown difficulties when applied to the detection of small-sized targets.

Among the factors that increase this challenge, scale variation is particularly noteworthy. Objects of interest are not always well represented in the image, and issues such as illumination and occlusion can also reduce the effectiveness of detection. Furthermore, the features of large objects can be more easily extracted compared to those of small objects, since the latter typically have low resolution and are noisy [11, 12, 13].

Several authors have proposed specific definitions for small objects. The relative definition considers small objects to be those whose width and height correspond to less than 10% of the dimensions of the original image [14, 15]. According to the absolute definition, small objects have a resolution lower than 32 × 32 pixels [14, 16].

To understand how the detection of small objects in images has been addressed, it is essential to review the recent literature. In this context, Section 2 provides an overview of the main related works, highlighting the strategies, techniques, and limitations reported by different authors.

Based on these considerations, the remainder of this paper is organized as follows. Section 2 provides an overview of the related works, with emphasis on the main strategies and limitations reported in the literature. Section 3 discusses the motivation behind this work and defines its main objectives, highlighting the challenges of detecting small objects in high-resolution images. Section 4 details the dataset adaptation, the preprocessing and postprocessing techniques, and the evaluation protocol adopted in this study. Section 5 presents and discusses the experimental results, including a comparative analysis between object detection and segmentation approaches. Finally, Section 6 summarizes the main findings and outlines directions for future research.

2. Related Work

Object detection approaches based on CNNs face recurrent challenges related to scale variation [14, 17, 18], background interference [15, 19, 20], and information loss caused by downsampling operations performed in deeper layers of the network [20, 21]. Such limitations become especially critical when identifying small objects relative to the image size.

With the advancement of satellite technologies, as well as Unmanned Aerial Vehicles (UAVs), object detection has become increasingly important in aerial imagery. Images obtained from these technologies are characterized by large background areas, with only a small portion of the image representing objects of interest [22, 3, 23], which may result in insufficient information to represent the objects due to their small sizes [4, 7].

In addition to remote sensing images [24, 25, 26] and images captured by drones [27, 28], the aforementioned challenges are also evident in domains such as defect and part detection [29, 30, 31], maritime scenes [32, 33, 34], insect and agricultural pest detection [35, 36], embedded systems [37, 38, 39], and traffic sign detection [40, 41, 42]. The diversity of contexts highlights the need for robust and adaptable solutions to mitigate the limitations of CNN in complex scenarios.

In this context, several studies have proposed structural improvements to CNN-based architectures [43], modifications to the region proposal stage [44], as well as advances in upsampling techniques [45], image segmentation [46], knowledge distillation methods [47], splitting the image into overlapping patches [35], and strategies to reduce false positives [26, 48].

Several review studies (surveys) provide a comprehensive overview of small object detection, summarizing advances, challenges, and recurring solutions in this field. For example, one analysis [49] of recent methods discusses aspects such as definitions of small objects, architectural improvements in convolutional and transformer-based networks, feature fusion techniques, data augmentation strategies, and specific adjustments for handling low-resolution objects. Similarly, another study [50] addresses the challenges inherent in large-scale small object detection, highlighting datasets, evaluation metrics, and standardized benchmarks, as well as proposing directions for future research.

In domains such as optical remote sensing, one work [51] describes methods aimed at detecting objects in high-resolution images, with emphasis on applications such as environmental monitoring, infrastructure inspection, and surveillance. This study highlights the importance of adapting architectures and preprocessing techniques to handle scale variations, high target density, and background interference—factors frequently present in agricultural and urban scenarios. These surveys consolidate existing knowledge, mapping gaps and opportunities, and provide guidance for the development of approaches related to small object detection.

This section is organized into subsections that highlight the strategies of related works. Subsection 2.1 provides a brief overview of studies on the detection of small objects in remote sensing images. Subsection 2.2 summarizes works related to drones and aerial images. Subsection 2.3 describes techniques adopted for detecting insects, arachnids, and agricultural pests in plantations.

2.1. Small Objects in Remote Sensing Images

The detection of small objects in remote sensing images is a challenging task, especially due to the high resolution of the images and the small ratio between the object size and the overall scene.

Methods such as You Only Look Once version 3 (YOLOv3) [52], Single Shot MultiBox Detector (SSD) [9], and Faster Region-based Convolutional Neural Network (Faster R-CNN) [8] were compared in a study [53] aimed at identifying small aircraft in images from *Google Earth* and the Dataset for Object deTection in Aerial images (DOTA) dataset [54]. The results showed that, in addition to higher speed, YOLOv3 also achieved better average detection performance compared to the other architectures.

To overcome the limitations of classical models, several approaches have proposed structural modifications. For instance, Context-Based Feature Fusion SSD (CBFF-SSD) [55] integrates feature fusion units and detection maps to improve the identification of small objects. Experiments on the Northwestern Polytechnical University Very High Resolution 10 (NWPU VHR-10) dataset [56] demonstrated significant precision gains compared to the traditional SSD.

Other approaches aim to enhance multiscale feature extraction. One example is the application of Mask Region-based Convolutional Neural Network (Mask R-CNN) [57] with Residual Neural Network with 101 layers (ResNet101) [58] adapted with Feature Pyramid Networks (FPN), which assists in detecting objects at different scales. The proposed method [24] was evaluated on datasets such as DOTA and Remote Sensing Object Detection (RSOD) [59], demonstrating promising results for detecting "airplane" and "ship" classes.

The use of attention mechanisms and context fusion has also proven to be effective. The Multiscale Context and Enhanced Channel Attention (MSCCA) model [25] combines the Pelee Network (PeleeNet) backbone [60] with Efficient Channel Attention (ECA) blocks, achieving 80.4% mean Average Precision (mAP) on DOTA and 94.4% on NWPU VHR-10, balancing detection speed and computational resource efficiency.

In addition to attention mechanisms, maintaining resolution has shown good results in detecting small objects in complex backgrounds. In this context, the High-Resolution Transformer-embedding Parallel detection Network (HRTP-Net) approach [21] proposes modules that preserve the high spatial resolution of small objects and distinguish their pixels from the background by means of attention mechanisms. Evaluated on the Maritime SATellite Imagery (MASATI) [61], VEhicle Detection in Aerial Imagery (VEDAI) [62], and DOTA datasets, the model outperformed traditional methods.

Computational limitations are common in devices such as satellites and drones. In this regard, the Guided Hybrid Quantization with One-to-One Self-Teaching (GHOST) model [47] employs guided distillation to preserve important details and detect small objects, reduces computational costs, and increases accuracy compared to traditional quantization methods. Evaluated on the VEDAI, DOTA, NWPU VHR-10, and DetectIon in Optical Remote sensing images (DIOR) [63] datasets, GHOST outperformed other detectors.

In the context of large-scale images (with dimensions of 20000 × 20000 pixels), Remote sensing Region-based Convolutional Neural Network (R^2-CNN) [26], based on Tiny-Net, stands out for its low memory consumption and for achieving a mAP of 96.04%. This network jointly trains a classifier and a detector, processing overlapping image patches to reduce false positives and increase localization accuracy.

Complex scenarios with overlapping objects and confusing backgrounds require solutions with greater contextual sensitivity. Scene-Contextual Feature Pyramid Network (SCFPN) [64] employs group normalization and improves the detection of small objects at multiple scales. The model was evaluated on the DOTA dataset and demonstrated superior performance over baseline methods at $IoU \geq 0.7$ metrics.

Even more robust approaches include architectures composed of multiple components. The Multi-Component Fusion Network (MCFN) [20] combines three distinct blocks: pyramid fusion, region selection based on relative intersection, and context incorporation. This structure significantly improves detection in complex scenarios, outperforming Faster R-CNN, YOLOv3, and SSD.

Considering low resolution or image noise, Edge-Enhanced Super-Resolution Generative Adversarial Network (EESRGAN) [65] employs a hybrid approach with Generative Adversarial Network (GAN) for edge enhancement and super-resolution. Experiments on the Car Overhead with Context (COWC) [66] and Oil and Gas Storage Tank Dataset (OGST) [67] datasets indicated that preserving structural details is essential for detecting small objects.

Recent approaches explore the potential of hybrid architectures. The Local Perception Swin Transformer (LPSW) [46] architecture incorporates elements from the *Swin Transformer* [68] along with spatial attention techniques to enhance segmentation accuracy. Based on datasets such as DIOR, High-Resolution Remote Sensing Detection (HRRSD) [69], and NWPU VHR-10, the approach demonstrated faster inference and superior segmentation results.

Specific approaches such as Hierarchical Scale Sensitive CenterNet (HSSCenterNet) [70] focus on vessel detection, integrating direction vectors to predict oriented bounding boxes. The Model with Deep Reinforcement Learning and Efficient Convolution Feature learning (MdrlEcf) model [71] incorporates reinforcement learning to improve the localization and classification of small objects, standing out in the detection of maritime and urban images.

Still in the context of oriented objects, some techniques [72, 73] employ rotated region of interest modules and aspect ratio between object width and height to estimate the orientation angle. Experiments conducted on the NWPU VHR-10, DOTA, University of Chinese Academy of Sciences – Aerial Object Detection (UCAS-AOD) [74], High Resolution Ship Collection 2016 (HRSC2016) [75], and German Aerospace Center 3K Vehicle Dataset (DLR-3K) [76] datasets showed that the proposed techniques outperform traditional object representation methods, in addition to being faster and more accurate during inference.

Another noteworthy proposal is Vehicle Detection Network based on Remote Sensing Images (VDNET-RSI) [77], a two-stage network that combines edge preservation using Local Implicit Image Function (LIIF), super-resolution, detection, and attention modules. Evaluated on the DIOR dataset, the approach outperformed models such as You Only Look Once version 5 (YOLOv5) [78], Faster R-CNN, and Fully Convolutional One-Stage Object Detection (FCOS) [79], demonstrating potential for applications in intelligent transportation systems.

These approaches reflect the diversity of strategies employed in the detection of small objects in remote sensing images, combining computational efficiency, accuracy, and robustness.

2.2. Small Objects in Aerial and Drone Images

The detection of small objects in aerial images represents a significant challenge in computer vision, especially in contexts with limited resources and complex visual scenarios. Conventional object detectors are effective for medium or large-sized targets but face difficulties when applied to the identification of small objects. This section summarizes proposed approaches to address these limitations.

The detection of defects in electrical insulators, characterized as small objects in complex backgrounds, motivated the proposal of Ghost Convolution and Centralized Feature Pyramid You Only Look Once (GC-YOLO) [29], an optimization of YOLOv5. While ghost convolutions extract features more efficiently, coordinated attention mechanisms highlight relevant regions of the image. Evaluated on a dataset with 1600 images and 5375 annotations, GC-YOLO outperformed traditional architectures.

An extension [27] of YOLOv5 introduces modules such as Feature Enhancement Block (FEBlock), Self-Characteristic Expansion Plate (SCEP), and additional detection layers to handle small objects in dense scenarios with background noise. Evaluated on the VisDrone2021 [80] dataset, the model significantly improved performance, increasing mAP@0.5 from 42.5% to 54.4% when using a resolution of 1024×1024. The results were promising under conditions such as nighttime streets and lighting variations.

A variation [28] of YOLOv3 incorporates modified residual blocks and a multiscale structure for prediction at different resolutions. The network was trained with a dataset containing 4406 images categorized by distance and background noise. Strategies such as preliminary data classification and retraining resulted in a mAP of 90.88%.

To address the computational limitations of embedded devices, Laplace Bottleneck and Cross-Layer Attention Upsampling You Only Look Once (LC-YOLO) [38] was proposed. The architecture incorporates modules that enhance details in the shallow layers through enhancement filters and fuse shallow and deep features using pixel-level cross-attention. Evaluated on the UCAS-AOD dataset, the model achieved a mAP@0.5 of 94.96%, outperforming more robust versions of You Only Look Once (YOLO).

Aiming at small object detection in UAV missions, a proposal [39] modified the You Only Look Once version 4 (YOLOv4) architecture by introducing the Advanced Downsampling Cross Stage Partial Darknet-53 (ADCSPDarknet53) backbone and a new loss function. The model incorporates data augmentation techniques and a classification method based on distance metrics. Evaluated with aerial images of small objects, the detector achieved mAP@0.5 of 61.00% with 77 Frames Per Second (FPS).

In the same context, Small Object Detection Convolutional Neural Network (SODCNN) [81], a variation of You Only Look Once version 7 (YOLOv7) [82], was proposed with several structural optimizations. Among the improvements are the removal of the large object detection module, an increased number of anchors, and the replacement of the Complete Intersection over Union (CIoU) loss function with Efficient Intersection over Union (EIoU). Evaluated on the VisDrone2019 dataset, the model achieved a mAP@0.5 of 54.03% and outperformed other models in the YOLO and Cascade Region-based Convolutional Neural Network (Cascade R-CNN) categories.

Deconvolution modules, super-resolution, and shallow layer fusion were combined to detect small objects. The model [83] was evaluated on datasets including cattle and pedestrian images captured by drones, achieving a mAP of 79.12% and Recall of 94.10%, outperforming traditional detectors. The balance between performance and accuracy proved suitable for surveillance and precision agriculture applications.

An alternative approach [84] explored the use of two convolutional networks to improve the detection of vehicles with multiple orientations and scales. The first network generates oriented region proposals based on hierarchical feature maps, while the second performs object classification. Evaluated on the VEDAI and Overhead Imagery Research Data Set (OIRDS) [85] datasets, the model outperformed traditional architectures.

Compact models have also been explored for small object detection when there are hardware constraints. One proposal [37] uses pretrained layers, concatenates multiscale features, and applies unsupervised training to extract representations. Prediction is performed by lightweight classifiers and an optimized regression model, balancing accuracy, performance, and low computational cost.

Another relevant approach is Attention Enhancement and Fusion Network (AEFNet) [32], proposed for small object detection in maritime scenes. The architecture combines the Swin-T backbone [68] with self-attention modules, highlighting features in complex backgrounds and fusing information across different scales to preserve small target details. Evaluated on the TinyPerson [86] dataset, AEFNet showed good performance in contexts with small objects and background noise.

A proposal [87] integrated the CSWin Transformer with Mask R-CNN, complemented by a hybrid module that incorporates smaller patches of the images. This approach aims to strengthen detection at multiple scales, preserving details such as edges and corners, and to improve the identification of small objects without increasing model complexity. The results showed significant gains, especially for small objects.

These approaches reflect the diversity of strategies proposed for the detection of small objects in aerial images, combining computational efficiency, detail preservation at multiple scales, and attention mechanisms to address the limitations imposed by low-resolution targets, complex backgrounds, and operational constraints.

2.3. Insects, Arachnids, Agricultural Pests, and Plantations

The detection of insects in plantation images presents challenges similar to small object detection, mainly due to the reduced size of the species and the similarity between individuals.

A study [35] divides images into overlapping 800 × 800 pixel patches to be processed by the YOLOv4 detector. By combining this strategy with Efficient Network (EfficientNet) in the classification stage, an accuracy of 89% was achieved. This approach proved effective in distinguishing small and similar species, such as Phyllotreta striolata and Phyllotreta atra.

To detect small pests, Yolo-Pest [36] was developed with modules that extract features in scenarios with few samples and a layer that expands receptive fields and reinforces informative channels. Evaluated on agricultural pest images, the model achieved 91.9% mAP@0.5, outperforming You Only Look Once version 5 – small (YOLOv5s) by almost 8% while reducing the number of parameters.

An approach [88] based on Gradient-weighted Class Activation Mapping (Grad-CAM) was applied to YOLOv5 for detecting wheat ears. The final architecture removes the large-scale layer, adds a micro-scale layer, and enhances feature extraction at the intermediate scale. Tests on the Global Wheat Head Detection Dataset (GWHD) [89, 90] dataset showed an increase in Average Precision (AP) to 93.5% at high resolution with a reduction in parameters.

The recognition of impurities in corn grains also requires attention to small objects. An architecture [91] extracts multiscale semantic features and generates bounding boxes using an Adaptive Region Proposal Network (ARPN), integrating Faster R-CNN with Efficient Network

– Model B7 (EfficientNetB7). The model outperforms alternatives such as ResNet101 and EfficientNetB7, standing out in the detection of small objects.

The automated identification of tick species in images was enabled by Tick Identification Network (TickIDNet) [92]. The model was trained on a dataset of images with variations in quality and object size. Although good accuracy was achieved, the model was affected by the relative size of the tick and characteristics such as life stage and feeding status.

To distinguish between normal and defective regions in navel oranges, Feature Skyscraper Detector (FSD) [93] was proposed. The architecture employs dense connectivity and optimizes the extraction of small object features, such as black spots, as well as precisely distinguishing the ends of the stem and the flower. Evaluated on a specific dataset, the model outperformed detectors such as YOLOv3 and SSD.

The studies reviewed summarize the detection of insects, pests, and defects in agricultural scenarios, highlighting the challenges associated with the identification of small objects, high visual similarity, and low data representativeness.

3. Motivation and Objectives

As observed in related work, object detection in images has evolved significantly in recent years, with CNNs architectures and, more recently, Transformer-based models achieving high performance mainly in contexts where objects are large relative to the image size, which favors the features extraction by the models.

Despite these advances in larger-scale scenarios, small object detection remains one of the main challenges in computer vision, especially in contexts with limited resolution, complex backgrounds, and high object density. Although significant progress has been made through the use of specialized architectures, attention mechanisms, and multiscale techniques, many studies focus on urban, maritime, and traffic scenarios, with emphasis on vehicles and vessels.

Insect detection, in turn, remains a less explored field, even though it shares several challenges with the aforementioned scenarios, such as the reduced size of objects and visual similarity between classes. Only a small number of approaches have been proposed for this domain, and some of them employ variations of the YOLO family. Moreover, a lack of strategies that leverage image pre and post-processing techniques and result refinement has been observed.

These challenges become even more evident in situations where images are resized to be processed by CNNs, causing information about small objects to disappear [94]. Even when they remain visible, as the images undergo successive convolutions and their feature maps become smaller, the information associated with these objects tends to be further lost [22, 1, 95].

In addition to the challenges mentioned above, the well-known datasets used for training CNNs, such as ImageNet[1], MS COCO[2], and PASCAL VOC[3], consist of images acquired from a frontal viewpoint and with a certain proximity to the objects. In this way, CNNs architectures developed for object detection are more suitable for the characteristics of these datasets [1].

Another issue that arises in small object detection is the lack of data, as most datasets contain annotations for large or medium-scale objects. For these specific problems, object detection algorithms such as CNNs may not be able to provide good results [2].

Related work has shown that the strategy of cropping images helps with the detection of small objects, but there is no standardization regarding its application. Some studies adopt

[1] Available at: `https://www.image-net.org/`

[2] Available at: `https://cocodataset.org/`

[3] Available at: `http://host.robots.ox.ac.uk/pascal/VOC/`

overlap between the cropped regions, while others do not use this feature. Another aspect that is rarely addressed is the procedure for reuniting objects that end up being split during the image cropping process. This highlights that current solutions are generally developed for very specific problems, which hinders their generalization to other scenarios.

In many scenarios, small objects are the main targets, and discarding regions may lead to the loss of important information. Given this challenge, efforts were made to detect insects in images of soybean leaves. When viewed from above, insects appear with reduced dimensions relative to the total image size.

To address this difficulty, the strategy of cropping images into smaller overlapping regions was adopted, ensuring that all areas of the image were analyzed and that no object of interest was discarded. In addition, this work compares approaches based on detectors and segmenters, evaluating the performance of both types of models in identifying the targets.

The general objective of this work was to propose techniques for detecting small objects in high-resolution images, exploring detection and segmentation approaches, especially in underexplored domains such as insect identification. To achieve this general objective, the following specific objectives were defined and accomplished:

- Annotate a dataset with insect images, which was originally developed for superpixel classification, thus contributing to research on the detection and segmentation of small objects;

- Develop and standardize preprocessing techniques, including methods for cropping images with overlap and converting bounding boxes into segmentation masks;

- Propose and implement postprocessing strategies, including the merging of detections that were split during preprocessing and alternatives to address the limitations of the IoU metric;

- Evaluate and compare detection and segmentation approaches for identifying small objects, analyzing the advantages and limitations of the original methods and the proposed pre and postprocessing techniques in this work.

4. Material and Methods

Small objects in high-resolution images become imperceptible when the images are resized as they pass through the layers of machine learning models. Considering that the objects are insects and the images represent soybean leaves, the targets are difficult to visualize even without resizing the image. To address this challenge, several techniques were implemented and are described in detail in this section.

The first technique adapts a dataset in which insects were annotated using superpixels. This adaptation was necessary because the proposed technique employs bounding boxes. Therefore, the superpixels were converted into bounding boxes that delimit the objects of interest. The adaptation of the dataset for the proposed problem and approach is described in Subsection 4.1.

The second technique consists of creating crops from the original image. These crops are then individually used for object detection or segmentation, reducing data loss when resizing the image to a smaller size. After detection or segmentation, the crops are merged to reconstruct the original image. The details of this proposal are presented in Subsection 4.2.

The third technique is a proposed metric that reduces the impact of pixel errors in small objects. This error is present in traditional metrics that use bounding boxes. Subsection 4.3 details the proposed metric and describes the experiments.

Table 1: Number of superpixels for each class [96].

Species	**Samples**
Diabrotica speciosa	358
Euschistus adult	3052
Euschistus mating	132
Euschistus nymph	342
Gastropoda	357
Spodoptera	89
Background	5670
Total	**10000**

4.1. Dataset Adaptation

Originally developed for a classification approach, the dataset [96] consists only of superpixels, which do not contain information about the location of the objects of interest in the images. This limitation restricts the use of the data in techniques that rely on detection and segmentation.

The objects of interest are insects segmented using the SLIC Superpixels technique [97] and classified by an expert. The segmentation process used only superpixels, and the object coordinates were discarded. Subsection 4.1.1 provides more details about the dataset.

The approach proposed in this study is based on bounding boxes, which require the coordinates of the objects of interest with respect to the image, making it necessary to adapt the original dataset because this information is not available. This adaptation is described in Subsection 4.1.2.

After adapting the original dataset to support object detection and segmentation, the final version provides comprehensive annotations for thousands of small insect instances in high-resolution images. The details regarding the structure, class distribution, and partitioning protocol of the final dataset are presented in Subsection 4.1.3.

4.1.1. Original Dataset

The dataset [96] contains 1,000 images collected in a soybean field over several days and under different weather conditions, between 06:00 and 19:30. The images were captured at a distance of approximately one meter from the soybean leaves and at an angle of about 45 degrees. All images have a size of 2268 × 4032 pixels.

In total, seven classes were considered for the superpixels, one of which represents the background of the image, that is, a superpixel that does not contain an insect. The other classes represent insect species, namely, Diabrotica speciosa, Euschistus adult, Euschistus mating, Euschistus nymph, Gastropoda, and Spodoptera.

Table 1 shows the number of samples for each species and the background class, totaling 10,000 superpixels.

4.1.2. Adapted Dataset

The dataset adaptation process was divided into three steps. In the first step, the superpixels were searched in the images. In the second step, matches of superpixels found in the images were selected. Finally, in the third and last step of the adaptation process, the bounding boxes of the superpixels were adjusted.

Table 2: Score for a superpixel from the Spodoptera class.

File name	xmin	ymin	xmax	ymax	score
626.jpg	3520	600	3760	840	17.37
142.jpg	3790	390	4030	630	28.20
185.jpg	3780	1050	4020	1290	29.22
653.jpg	3560	410	3800	650	29.50
994.jpg	110	810	350	1050	29.88
199.jpg	1000	990	1240	1230	29.98
252.jpg	3170	810	3410	1050	30.06
355.jpg	2890	1800	3130	2040	30.45
564.jpg	2410	780	2650	1020	30.47
77.jpg	1230	1960	1470	2200	30.59

The first step of the dataset adaptation process was to search for each superpixel in the original image and obtain the coordinates of its bounding box. For this purpose, the Template Matching algorithm[4], available in the OpenCV[5] Computer Vision library, was used.

A script was developed to search for each superpixel sample, except for the “Background” class, in all 1,000 images of the dataset. This script creates a text file for each superpixel, and each line in the file contains information about the image in which the superpixel was searched, the coordinates of the match found, and its score.

The first 10 lines of the file generated by searching for a superpixel from the Spodoptera class are shown in Table 2. The lines are ordered by increasing score, indicating that the searched superpixel is most likely part of the image “626.jpg”, within a rectangle whose top-left corner is at position $(3520, 600)$ and bottom-right corner is at position $(3760, 840)$.

A Graphical User Interface (GUI) application was developed to assist in selecting the best match. In this application, the user can view the superpixel and the 10 best matches found among the 1,000 images, and then click on the pattern most similar to the superpixel.

When a match is found, a JSON file is used to store information about the image to which the superpixel belongs, as well as the coordinates of the bounding box that delimits the superpixel region. Figure 1 shows the application’s main screen.

In Figure 1, the searched pattern is shown on the left, and the 10 best matches found are displayed side by side and numbered from zero to nine. In this example, the best match found for the superpixel from the Spodoptera class is number 0. By clicking on it, the information about the image in which the pattern was found, the coordinates of the matched pattern, and the respective class of the object are stored.

Most superpixels were found using this technique. When a superpixel was not found among the first 10 matches, the next 90 matches were checked. If, even then, the corresponding superpixel was not found, it was discarded and not added to the dataset.

The final step of the adaptation process was to adjust the object coordinates. Since the coordinates obtained corresponded to the superpixels, most of the object region was actually composed of image background. Therefore, it was necessary to adjust the bounding box of each object.

[4] Available at: `https://docs.opencv.org/master/d4/dc6/tutorial_py_template_matching.html`

[5] Available at: `https://opencv.org/`

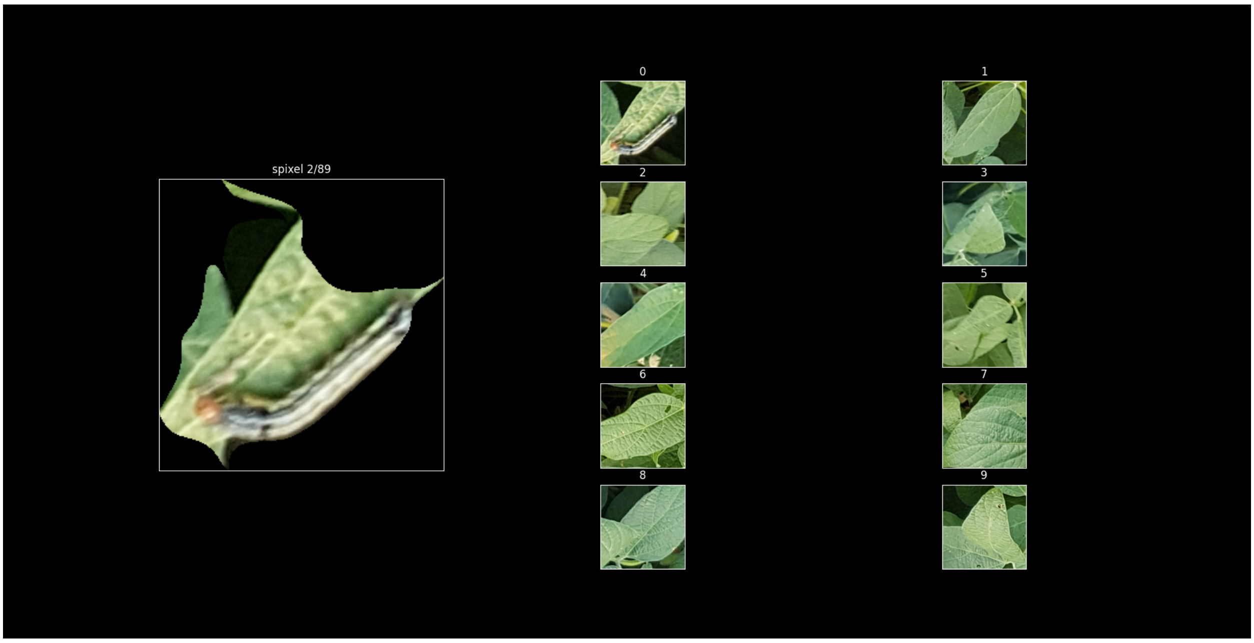


Figure 1: Main screen of the application developed to assist in selecting the best match.

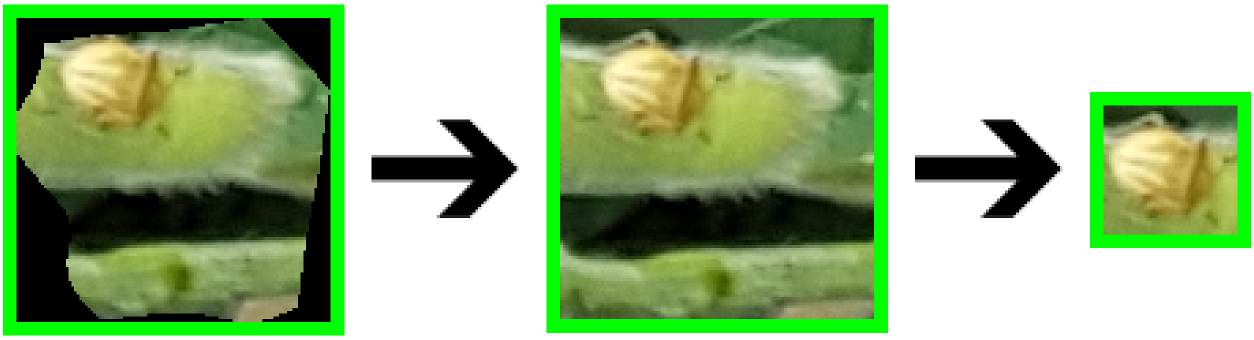


Figure 2: Bounding box coordinate adjustment.

The LabelMe software[6] was used in this step. Figure 2 shows an example of bounding box coordinate adjustment. From left to right, the first image represents a superpixel. The second image depicts the region corresponding to the superpixel in the source image. Finally, the third image corresponds to the bounding box that encompasses only the object of interest. In this case, an insect from the Euschistus class.

In addition to the coordinate adjustment, new insects were annotated in the images, and the insects of the species Euschistus adult, Euschistus mating, and Euschistus nymph were merged into a single class (Euschistus), thus finalizing the dataset adaptation process.

After all adjustments, the dataset was defined with five classes, including one class to represent the image background. Table 3 shows the number of bounding boxes for each class.

4.1.3. Final Dataset

The final dataset comprises 10,537 objects (excluding the “Background” class), annotated in 1,000 images. Figure 3 shows an image from the dataset, with samples enlarged by 10×, highlighting that the objects are small relative to the image size.

The dataset images were randomly distributed to ensure that approximately 70% of the

[6] Available at: `https://github.com/wkentaro/labelme`

Table 3: Number of bounding boxes for each class.

Category Id	Species	Count
1	Diabrotica	1346
2	Euschistus	8126
3	Gastropoda	905
4	Spodoptera	160
5	Background	1000
Object total	—	**11537**
Image total	—	**1000**

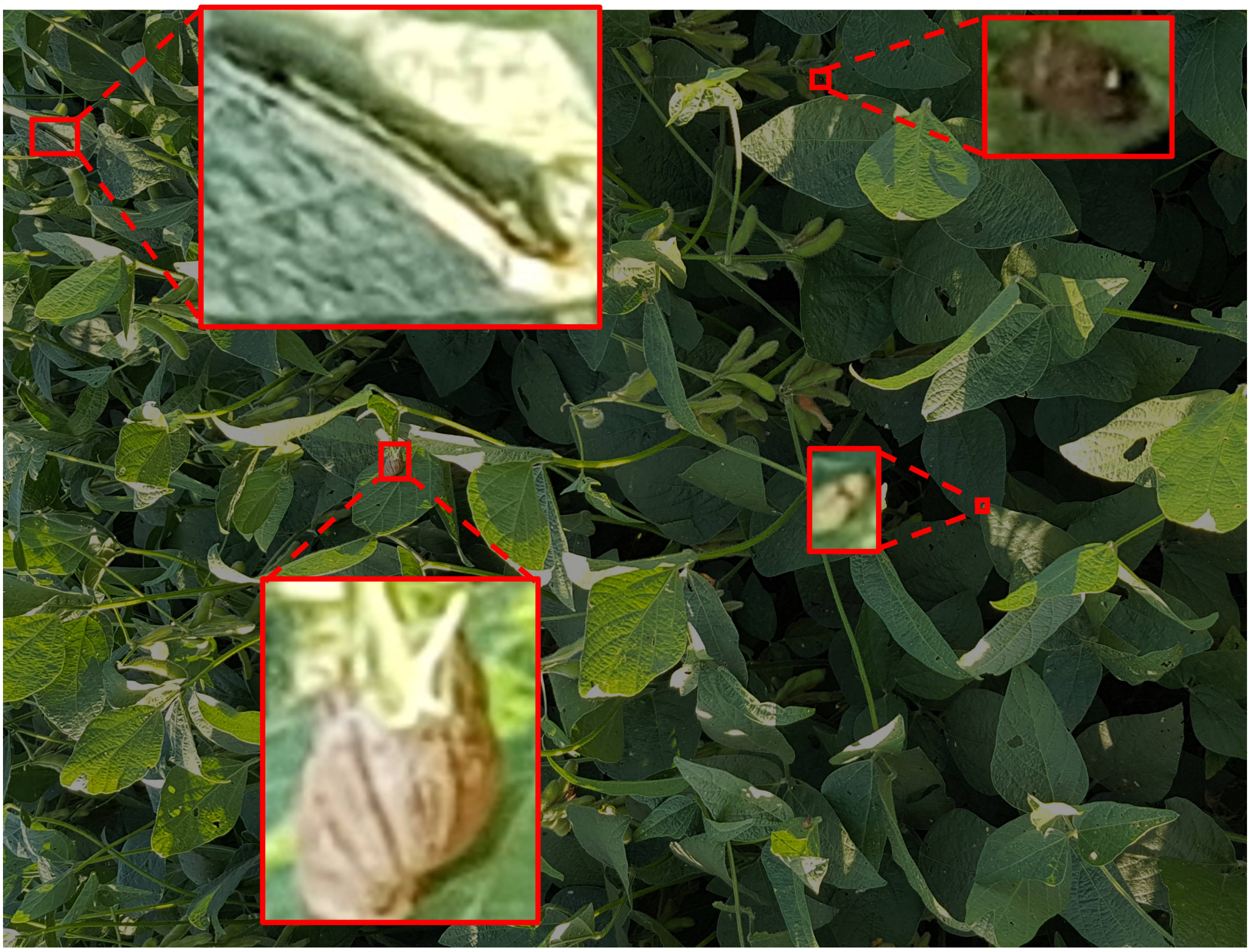


Figure 3: Samples of dataset species enlarged by 10×.

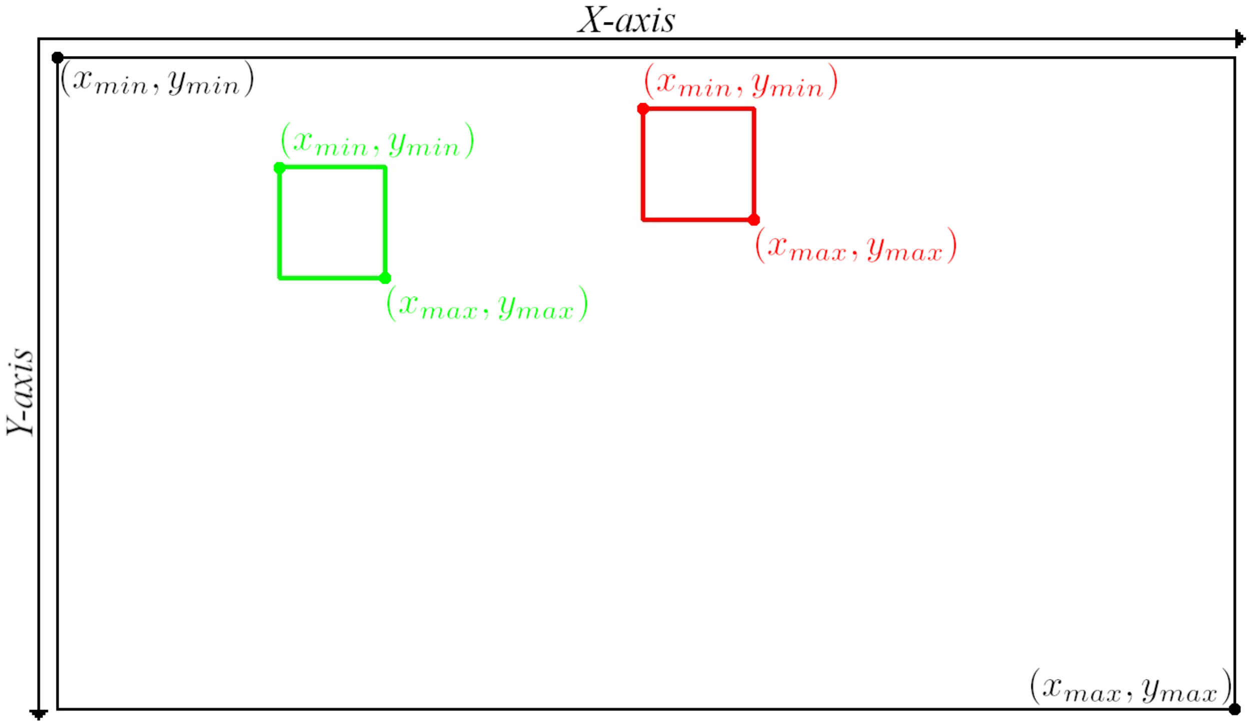


Figure 4: Representation of the plane (black lines with arrows), image or crop (black rectangle), annotation (green rectangle), and detection (red rectangle).

samples of each species were used for training, 15% for validation, and 15% for testing. Table 4 presents the results of this split.

Table 4: Dataset split.

Category Id	Species	Training	Validation	Test	Total
1	Diabrotica	951 (71%)	200 (15%)	195 (14%)	1346
2	Euschistus	5610 (69%)	1264 (16%)	1252 (15%)	8126
3	Gastropoda	633 (70%)	132 (15%)	140 (15%)	905
4	Spodoptera	111 (69%)	24 (15%)	25 (16%)	160
5	Background	700 (70%)	150 (15%)	150 (15%)	1000
Object total	—	**8005 (69%)**	**1770 (15%)**	**1762 (15%)**	**11537**
Image total	—	**700 (70%)**	**150 (15%)**	**150 (15%)**	**1000**

4.2. Proposed Method

An image may lose information during the resizing step. At this stage, objects may become smaller or even disappear, which is a problem observed in the insect dataset. Thus, the proposed method suggests a preprocessing step, referred to as Cropping.

In this section, images, crops, annotations, and detections will be demonstrated using rectangles, which in turn will be represented by two points: the top-left point and the bottom-right point. The coordinates of each point consist of two values, represented by x and y.

The coordinate system is similar to the Cartesian plane, with the Y axis inverted. Figure 4 shows an example of an image, with an annotation and a detection, and their respective points represented in the plane.

4.2.1. Cropping

In the cropping step, each input image is divided into $N \times N$ crops, forming a grid, where N is a parameter to be specified. Each crop approximately preserves the aspect ratio of the original image. An example of an image of size 756×1344 pixels with annotated objects is shown in Subfigure 5a, and an example of a 3×3 grid application is shown in Subfigure 5b.

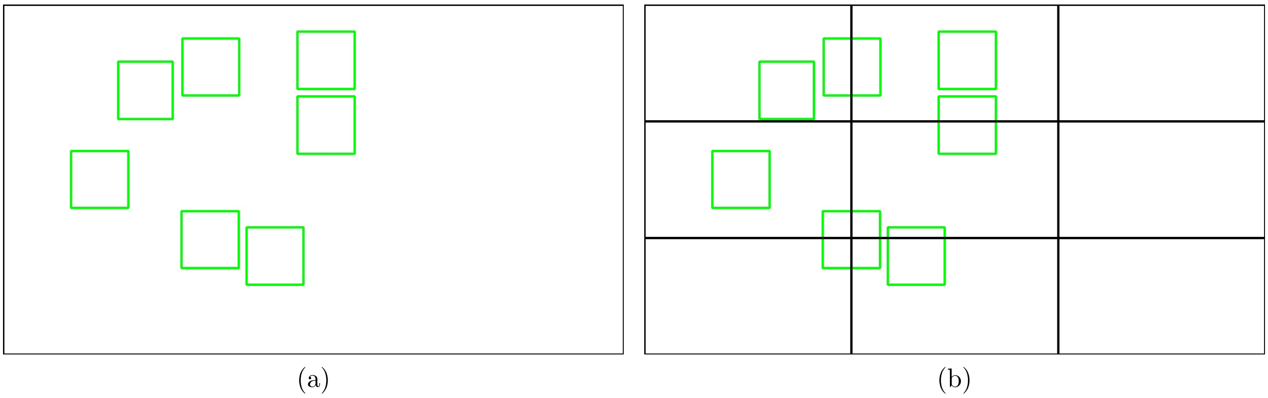


Figure 5: Application of a 3×3 grid on a 756×1344 pixel image. (a) Image with objects annotated in green; (b) Image with grid application dividing the annotated objects.

Each crop in Subfigure 5b is used individually as input for the training and validation steps, including crops that do not contain insects. All crops are annotated with the "Background" class, which encompasses their entire area. These negative samples help the model learn regions of the image without insects, reducing the number of false positives.

If a horizontal or vertical grid line divides an annotated object, the points delimiting this object are adjusted, as are the points of objects that are not divided, to ensure that they do not exceed the limits of their respective crop. The verification of annotated object division is performed by iterating over each crop, calculating its intersection with all ground-truth bounding boxes from the original image. When an intersection is detected, a new annotation is generated with coordinates expressed relative to the crop local reference frame. In cases where a grid line splits an object, the resulting fragments are preserved as separate annotations, unless their height or width falls below predefined thresholds, in which case they are discarded.

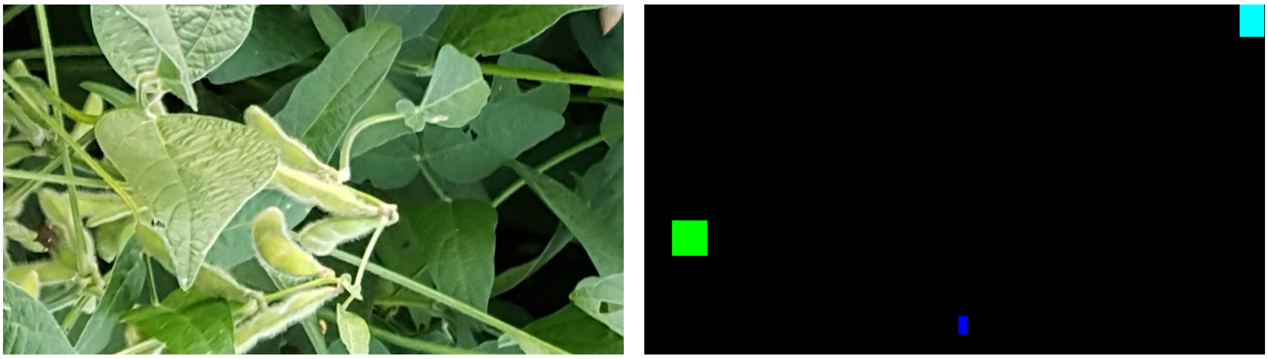
The image splitting process generates new images and, in cases where an object is divided, generates new objects. The behavior of the splitting process is similar to data augmentation methods based on cropping[7], which consist of extracting rectangular crops from the training images to increase the data available for training.

In the example shown in Figure 5, nine new images are created from a single image, and the number of annotated objects increases from seven to 13. The number of images and the number of objects in a dataset may vary depending on the grid size chosen.

4.2.2. Mask Creation for Segmentation

Detection and segmentation are two distinct approaches for analyzing image content. Detection methods define the location of an object through bounding boxes, whose coordinates are obtained by a regression process. Segmentation, in turn, is a method that classifies each pixel in the image with a label corresponding to an object category.

[7] Available at: `https://pytorch.org/vision/stable/transforms.html`



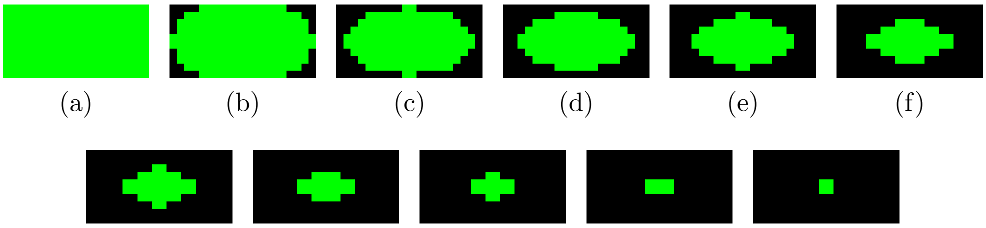
(a) (b)

Figure 6: Pair of crop and its corresponding mask. (a) Crop representation; (b) Corresponding mask.

The proposed approach uses bounding box coordinates to create segmentation masks, which are used to train, validate, and test the model. A mask is an image with the same size as the original image, and each pixel is assigned the value of the corresponding object category.

After the step of splitting the images into smaller patches, each crop is used to create a mask. An example of a crop and its corresponding mask is shown in Figure 6.

The mask for each bounding box is computed such that pixels closer to the center have values equal to or near 1.0, while pixels near the edge have values close to or equal to 0.0. A filter is applied to discard pixels with values below the established threshold. Figure 7 shows a bounding box and the resulting masks with a threshold ranging from 0.0 to 1.0.



(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k)

Figure 7: Masks generated for different threshold values (0.0 to 1.0 in 0.1 increments), corresponding to labels (a)–(k), respectively.

4.2.3. Image Reconstruction

The image is reconstructed from predictions in overlapping crops. The overlaps are performed in such a way that, in at least one crop, an object is not divided. A representation of the overlapping grids is shown in Figure 8.

In Subfigure 8a, a 3 × 3 grid is applied to an image containing four objects in red, green, blue, and light blue. The red object represents an object that was not split at this stage, while all other objects were affected by the application of this grid.

In Subfigure 8b, the blue object—which was split in the previous grid—is now complete, while the red object has a small segment cropped from its top. The green object, which was cropped in both the 3 × 3 and 2 × 3 grids, is not cropped in the 3 × 2 grid shown in Subfigure 8c. Even the light blue object, which was divided into four parts in the first grid, can be seen completely in the 2 × 2 grid.

After prediction in each of the grids, overlapping parts of the same object are eliminated using an adapted non-maximum suppression approach, which keeps objects with larger area

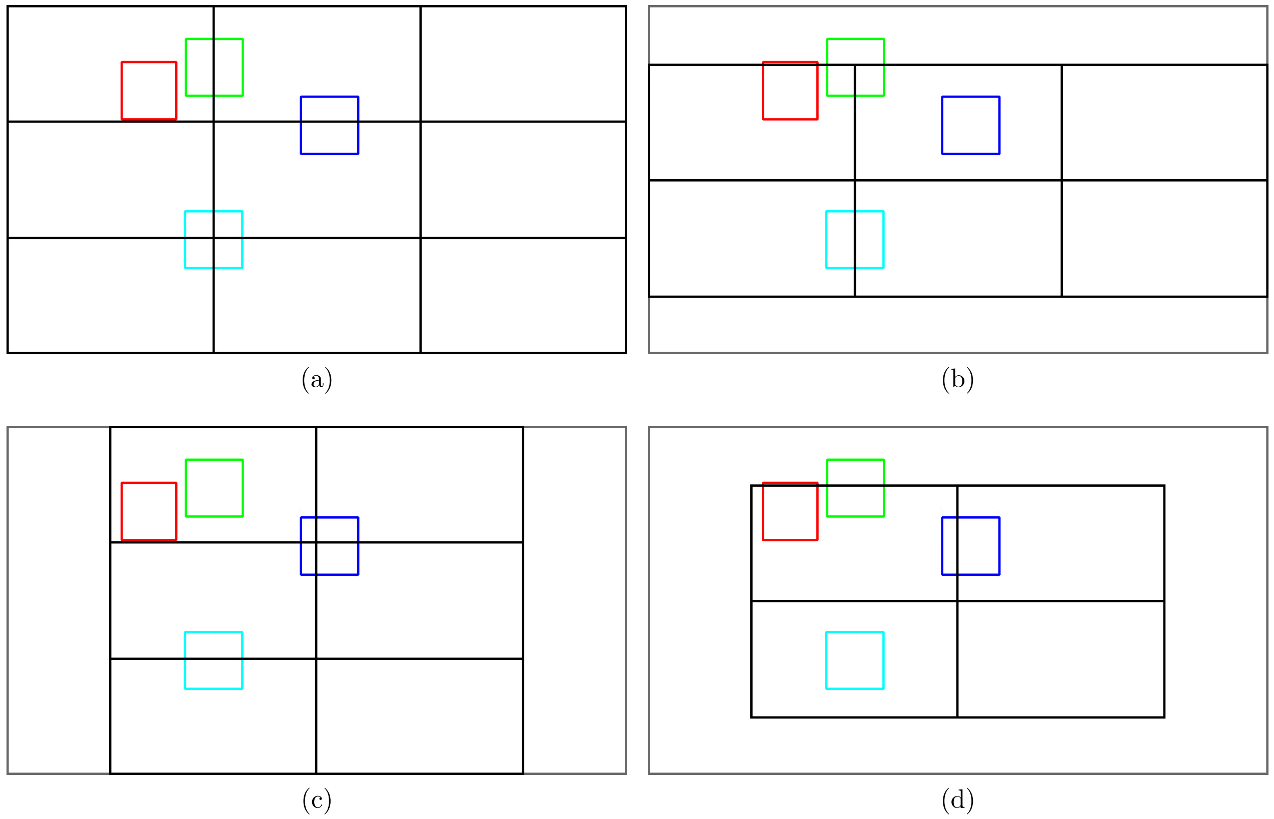


Figure 8: Representation of grids applied to an image. The outer border (in gray) represents the image boundaries. (a) 3 × 3 training grid that covers the entire image; (b) 2 × 3 grid focusing on the vertical center of the image; (c) 3 × 2 grid focusing on the horizontal center of the image; (d) 2 × 2 grid focusing on the center of the image.

instead of those with the highest score. This ensures that objects that were not split are retained rather than their smaller parts.

In addition, objects that have 0.5 or more overlap percentage with another are also discarded. This is necessary because non-maximum suppression does not always eliminate all duplicates.

4.2.4. Proposed Metric for Evaluation

Traditional detection metrics, such as IoU, have some limitations when objects are small [20, 98]. In these cases, small variations in the predicted coordinates decrease the IoU value, even when the prediction is visually close to the annotation. Table 5 demonstrates how pixel displacement affects the IoU value for different bounding box sizes.

In Table 5, it can be observed that the smaller the object, the more sensitive the IoU becomes to prediction displacement, which can increase the number of false negatives. Given this limitation, and based on other distance metrics [39, 99, 100], a metric that combines predictions and annotations is proposed.

The metric consists of measuring the distance between all prediction centers and all annotation centers, for each image and for each class. This process creates a distance matrix, where the rows represent the annotations and the columns represent the predictions.

Once the distance matrix is computed, the smallest distance is compared with a predefined threshold for the minimum distance between centers. The annotation and prediction corresponding to the smallest distance, if the threshold is satisfied, are considered a valid match. Otherwise, they are considered an invalid match.

Table 5: IoU for different bounding box sizes and pixel displacements. In red, values below 0.70.

Unit: pixels	Displacement									
Size	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
10x10	0.82	0.67	0.54	0.43	0.33	0.25	0.18	0.11	0.05	0.00
20x20	0.90	0.82	0.74	0.67	0.60	0.54	0.48	0.43	0.38	0.33
30x30	0.94	0.88	0.82	0.76	0.71	0.67	0.62	0.58	0.54	0.50
40x40	0.95	0.90	0.86	0.82	0.78	0.74	0.70	0.67	0.63	0.60
50x50	0.96	0.92	0.89	0.85	0.82	0.79	0.75	0.72	0.69	0.67
60x60	0.97	0.94	0.90	0.88	0.85	0.82	0.79	0.76	0.74	0.71
70x70	0.97	0.94	0.92	0.89	0.87	0.84	0.82	0.79	0.77	0.75
80x80	0.98	0.95	0.93	0.90	0.88	0.86	0.84	0.82	0.80	0.78
90x90	0.98	0.96	0.94	0.91	0.89	0.88	0.86	0.84	0.82	0.80
100x100	0.98	0.96	0.94	0.92	0.90	0.89	0.87	0.85	0.83	0.82

Valid matches are used to count the true positives. Invalid matches are used to account for annotations without predictions and predictions without annotations, thus allowing the estimation of false negatives and false positives.

After counting the false negatives, false positives, and true positives, it is possible to compute Precision, Recall, and F-measure, the latter being the metric used to define the detection capability of the proposed approach.

4.3. Experiments and Evaluation

The initial experiments were conducted on a computer with 16 GB of RAM, an Intel® Core™ *i*5-9600K CPU @ 3.70 GHz ×6, running Ubuntu 24.04.1 LTS, and an NVIDIA GeForce RTX 2060 GPU with 12 GB of memory.

Data augmentation techniques were applied to improve the generalization capability of the CNN [101]. Random flipping with a probability of 50% was used in the training set. In addition, annotations that were divided among different subimages also increased the overall amount of data, simulating the crop technique.

The object detection experiments were conducted using the MMDetection framework[8]. Five object detectors were used as baselines for comparison: Faster R-CNN [8], Retina Network (RetinaNet) [102], DETR with Improved deNoising anchOr boxes (DINO) [103], DAB-DETR [104], and YOLOv3 [52].

The architectures Segformer [105], DeepLabV3 [106], and DeepLabV3+ [107] were used as baselines for evaluating the proposed method. All segmentation experiments were conducted using the MMSegmentation framework[9]. For Segformer, the standard variants $b0$, $b1$, $b2$, $b3$, $b4$, and $b5$ were used.

All trainings used the default hyperparameter values provided by the framework. Each training run was performed for 30 epochs, with the batch size adjusted according to the architecture and the available GPU memory. Evaluation metrics were computed based on predictions over the entire image, that is, by merging the predictions from all crops.

[8] Available at: `https://github.com/open-mmlab/mmdetection`

[9] Available at: `https://github.com/open-mmlab/mmsegmentation`

5. Results and Discussion

This section presents the results obtained from experiments conducted on the test set. To enable a direct comparison between detection and segmentation tasks, two representative models were selected for this analysis: DINO [103] and Segformer B3 [105].

Subsection 5.1 compares training results using full images and crops, highlighting their impact on the training process. Subsection 5.2 presents the inference and image reconstruction procedures, including bounding box filtering, and describes the evaluation protocol based on center-based matching. This section also compares the results obtained using image reconstruction with those from the approach in which each crop is evaluated independently. Finally, Subsection 5.3 compares object detection with segmentation-based approaches.

5.1. Effect of Image Cropping on Model Training

The effects of applying the image cropping technique become noticeable in the very early stages of model training. The main impact lies in the increased processing time, which grows proportionally to the number of crops generated from the original images. The increase in the number of image crops significantly impacts training time for both detection and segmentation models. Table 6 presents a comparison of training times for DINO and Segformer B3, using full images and crops generated with a 5×5 grid. The reported times correspond to 30 training epochs for each configuration.

Table 6: Comparison of training times for DINO and Segformer B3 using full images and crops generated with a 5×5 grid. Times are presented in the format days:hours:minutes (dd:hh:mm).

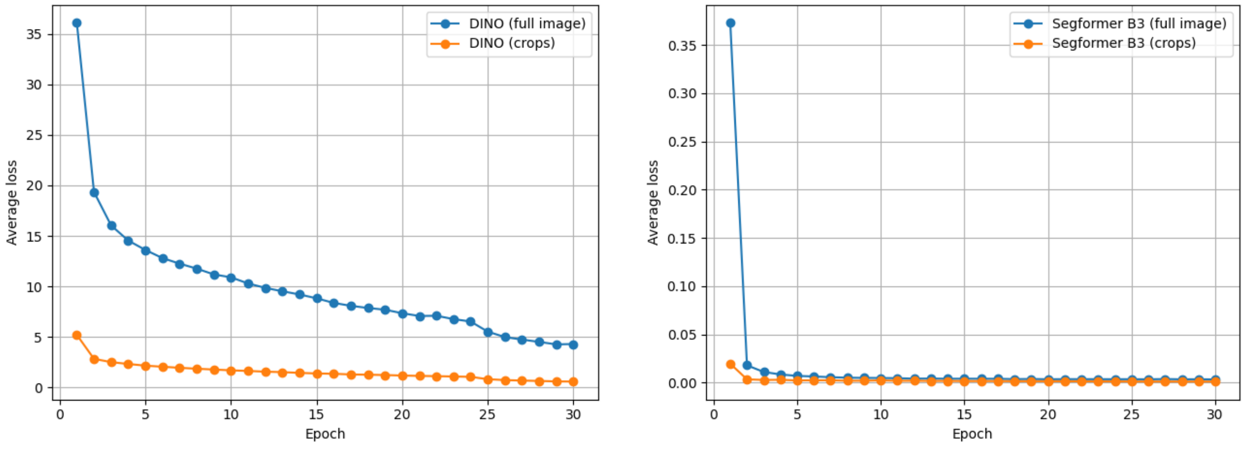
Model	**Full images**	**Crops**
DINO	00:03:11	03:09:22
Segformer B3	00:00:56	00:20:52

In these experiments, a total of 700 full images were used for training and 150 for evaluation. When using crops, each image was divided into 25 patches, resulting in $17,500$ training samples and $3,750$ evaluation samples for each epoch. For the DINO model, a batch size of 2 was used during training and 1 during validation, while for Segformer B3, the batch sizes were 8 for training and 1 for validation. It is important to note that the training times presented in Table 6 include both the training and evaluation steps for all epochs.

Despite the higher computational cost, the increase in the number of samples brings significant benefits to the learning process. Analysis of the loss function shows that training with crops results in lower error values compared to training with full images. An example of this behavior is presented in Figure 9.

The training loss curves shown in Figure 9 reinforce the positive impact of using crops during model training. For both DINO (a) and Segformer B3 (b), the models trained with crops exhibit consistently lower loss values throughout the epochs when compared to those trained with full images. This behavior is especially pronounced for the detection model (DINO), where the use of crops leads to a substantially faster and more stable reduction in loss.

In the case of the Segformer B3, although the training loss quickly converges to very low values, this result should be interpreted with caution. Since background pixels represent the vast majority of the image, the model can achieve low loss values by simply predicting the background class, which may give a false impression of high accuracy. Therefore, it is essential to complement loss analysis with more robust evaluation metrics.



(a)

(b)

Figure 9: Comparison of training loss curves for detection and segmentation models. (a) full images and crops (5×5 grid) for DINO; (b) full images and crops (5×5 grid) for Segformer B3.

Building on these observations, we now present a quantitative comparison of model performance for both detection and segmentation tasks. Table 7 summarizes the results obtained for DINO and Segformer B3, comparing training and evaluation using full images, and using crops generated by dividing each image into a 5×5 grid. For the crop-based approach, predictions were performed on overlapping crops using multiple grid configurations (4×4, 4×5, 5×4, and 5×5), and the results were aggregated.

Table 7: Evaluation metrics for DINO and Segformer B3 on the test set using full images and crops.

Training Method	**Metric**	**Full images**	**Crops**
DINO	**FN**	802	360
	FP	276	393
	TP	810	1252
	P	0.746	0.761
	R	0.502	0.777
	F	0.600	0.769
	mAP*	0.300	0.472
Segformer B3	**FN**	859	334
	FP	265	360
	TP	753	1278
	P	0.740	0.780
	R	0.467	0.793
	F	0.573	0.786
	aAcc*	0.999	0.999
	mIoU*	0.279	0.625
	mAcc*	0.296	0.681

Table 7 includes both the main metrics (False Negative (FN), False Positive (FP), True Positive (TP), Precision (P), Recall (R), and F-score (F)), which were computed by aggregating the predictions across all crops for each image, and the conventional detection and segmentation metrics (indicated with an asterisk). The latter—such as mean mAP for detection and IoU for segmentation—were calculated based on the predictions from individual crops, without merging them, in order to provide comparability with standard evaluation protocols. This distinction is essential for correctly interpreting the values, as the aggregated metrics better reflect the performance of the complete pipeline, while the asterisked metrics align with traditional single-image evaluation approaches.

The results in Table 7 clearly demonstrate the advantages of the cropping strategy for both detection and segmentation models. For DINO, the number of false negatives drops from 802 to 360 with crops, while true positives increase from 810 to 1252. This leads to a substantial improvement in recall and F-score, with only a modest rise in false positives. Similarly, for Segformer B3, applying crops nearly halves the number of false negatives and boosts both recall and F-score, again with a small increase in false positives. Precision remains relatively stable in both cases, indicating that the gains in recall are not solely at the cost of more incorrect predictions.

Conventional metrics (mAP, average Accuracy (aAcc), mean Intersection over Union (mIoU), and mean Accuracy (mAcc)), also show improvements when using crops. However, the absolute values of these metrics remain low compared to the overall performance of the models, and may underestimate their true effectiveness. In contrast, the other metrics reflect the gains achieved through cropping. This highlights the importance of employing evaluation strategies that capture the advances provided by the proposed approach.

Although the metrics in Table 7 summarize the quantitative differences between the evaluated strategies, they do not fully capture the qualitative effects of cropping. Figure 10 provides visual examples for both detection and segmentation models, comparing predictions with and without the use of crops.

The results illustrated in Figure 10 provide further insights into the practical impact of applying the cropping and filtering strategy. To properly interpret the results, it is important to note that the colored boxes—green for true positives, red for false negatives, and orange for false positives—are shown only in the prediction images.

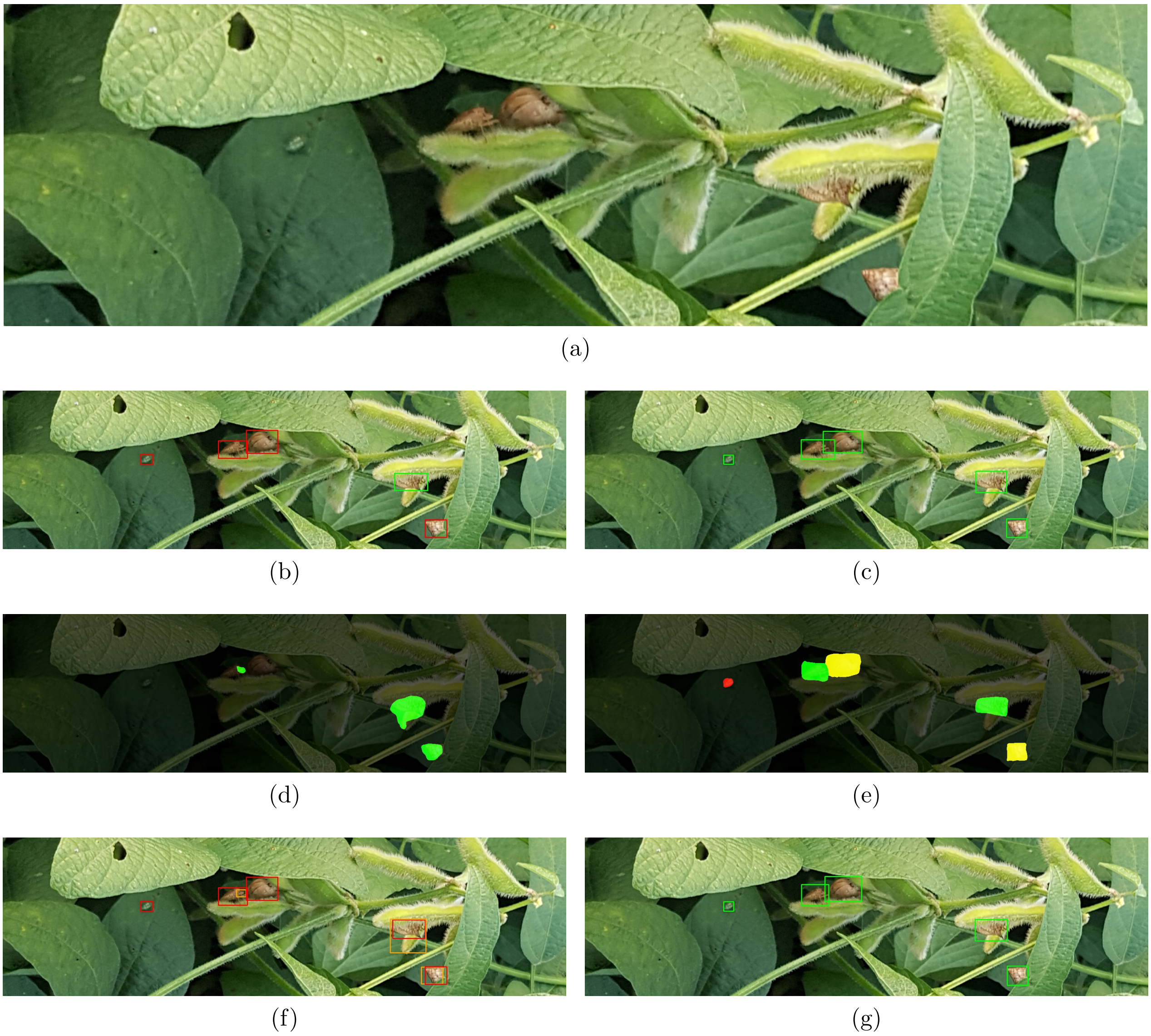
Subfigure 10a displays a crop of the original image, which contains some insects but does not include any annotations or predictions. When predictions are performed using DINO on the full image (Subfigure 10b), only one object is correctly detected. In contrast, when using crops (Subfigure 10c), all objects are identified, demonstrating a clear improvement in detection performance with the proposed approach.

When analyzing the segmentation results in Subfigures 10d and 10e, each color represents a different object category, while the darkest color corresponds to the background. When the segmentation is performed on the full image, not all objects are correctly segmented, and some that are segmented are assigned the wrong class. With cropping, however, all objects are both accurately segmented and correctly classified.

When converting the segmentation predictions into bounding boxes (Subfigures 10f and 10g), it is evident that the predictions from the full image contain only false positives and false negatives. In contrast, the cropping-based technique achieves correct localization and classification for all insect instances, further reinforcing the quantitative findings that cropping and filtering contribute to more effective detection and segmentation of small objects.

5.2. Inference and Image Reconstruction

The inference process consists of loading the trained model and generating predictions for each individual crop. Each crop contains metadata indicating its coordinates within the original image, which allows the mapping of predicted bounding boxes to their correct locations in the


(a)

(b)

(c)

(d)

(e)

(f)

(g)

Figure 10: Comparison of detection and segmentation results before and after applying the cropping strategy. (a) Crop of original image; (b) DINO predictions on the full image (no cropping); (c) DINO predictions after applying cropping and post-processing; (d) Segformer B3 predictions on the full image (no cropping); (e) Segformer B3 predictions after applying cropping; (f) Bounding boxes from Segformer B3 predictions (no cropping); (g) Bounding boxes from Segformer B3 predicions after applying cropping and post-processing.

full image. This subsection details the procedures adopted to reconstruct the complete image from crop-based predictions and to refine the detection results.

Initially, bounding boxes with a confidence score below a predefined threshold, as well as boxes whose width or height are smaller than a minimum value, are filtered out. Next, NMS is applied to remove redundant detections, keeping only the boxes with the largest area. Finally, the remaining boxes are further filtered so that, if the overlap between two boxes exceeds a certain threshold, the one with the smaller area is ignored. This procedures are described in detail in Subsubsection 5.2.1.

After this post-processing, the evaluation metrics are computed. For each image, the distances between the centers of annotated objects and the centers of predicted boxes are calculated. These distances are used to establish correspondences between predictions and ground-truth annotations, allowing the identification of true positives, false positives, and false negatives, and the calculation of performance metrics. The evaluation protocol is detailed in

Subsubsection 5.2.2.

Once the metrics for the reconstructed images are obtained, they are compared with those calculated when evaluating each crop individually. This comparison enables a comprehensive analysis of the benefits and limitations of the image reconstruction approach. Results of this comparison are presented in Subsubsection 5.2.3.

5.2.1. Bounding Box Filtering

Given that the cropping technique proved effective for the detection of small objects, as evidenced in the previous section, the present subsection examines the impact of different filters on the predictions. To this end, two models—DINO and Segformer B3—were selected to analyze the performance of detection and segmentation in distinct post-processing scenarios.

Initially, the results obtained on the test set for both models are presented without the application of filters, always using crops generated by a 5 × 5 grid. The metrics calculated from all predictions are shown in Table 8, serving as a reference for the subsequent post-processing steps.

Table 8: Metrics obtained without the application of filters for both models using crops generated by a 5 × 5 grid.

Filter	**DINO**						**Segformer B3**					
	FN	**FP**	**TP**	**P**	**R**	**F**	**FN**	**FP**	**TP**	**P**	**R**	**F**
No filters	13	2873405	1599	0.001	0.992	0.001	278	4240	1334	0.239	0.828	0.371

In the case of DINO, a significant number of false positives is observed, indicating that the model produces multiple predictions for a single annotation or incorrectly identifies background regions as objects. This characteristic becomes even more evident when analyzing an image containing all generated bounding boxes, as illustrated in Figure 11.

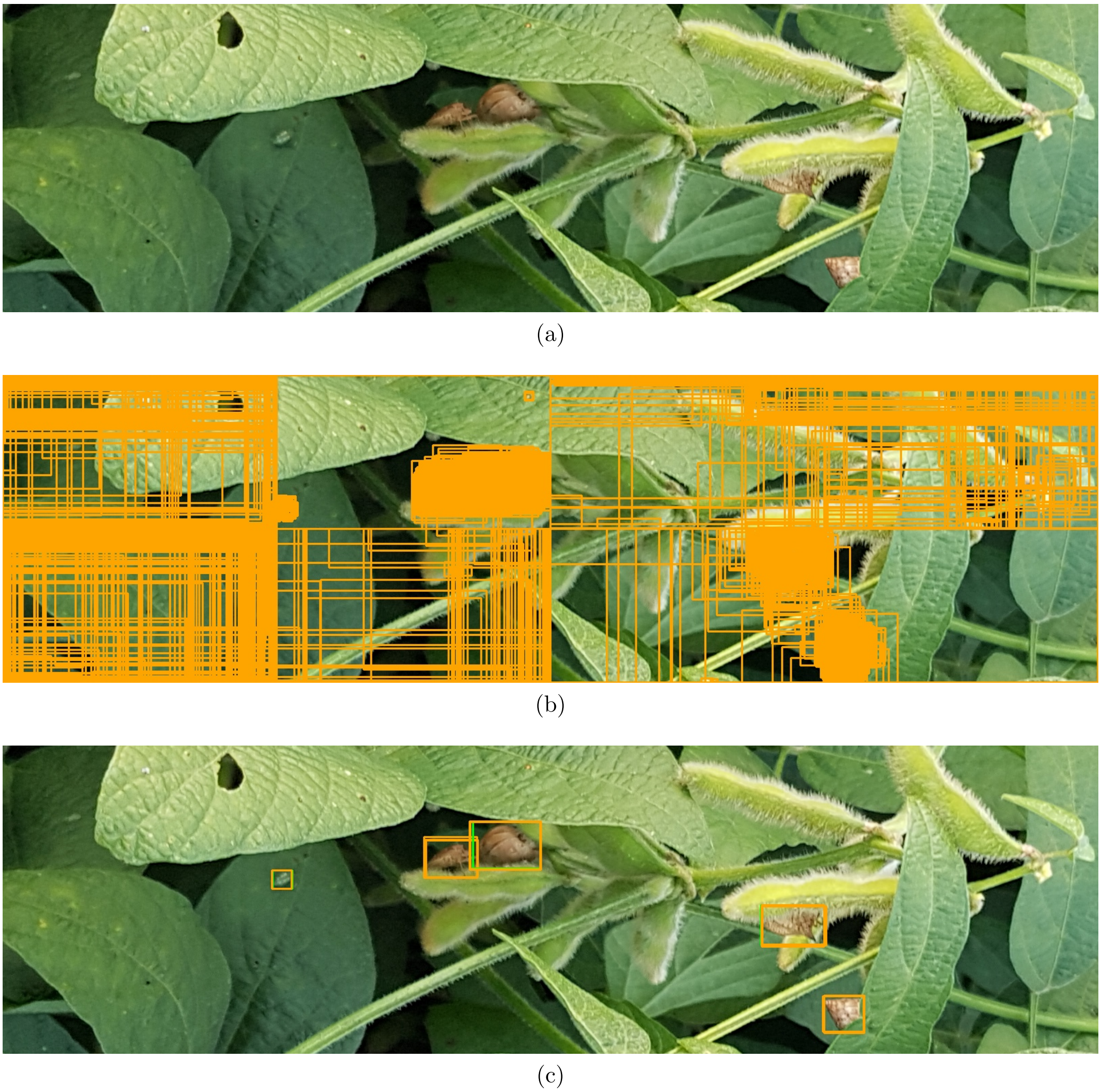
As observed in Subfigure 11b, the large number of false positives produced by DINO results in an excessive number of bounding boxes, making it difficult even to visualize the objects of interest. In contrast, Segformer B3 exhibits a different behavior, with the generation of bounding boxes limited only to relevant regions, that is, those distinct from the background. These initial results serve as a baseline to be surpassed, and the impact of the filters will be analyzed according to the order in which they are applied.

The first filter eliminates bounding boxes based on a minimum confidence threshold. Next, the second filter discards predictions whose area is below predefined values for each category. The third filter applies variations of NMS to remove redundant predictions. Finally, the fourth filter excludes predictions that exhibit an overlap greater than a specified percentage between their areas.

The initial stage of post-processing consists of eliminating bounding boxes whose confidence score is below a defined threshold. To analyze the impact of this filter, different confidence thresholds were tested, ranging from 0.0 to 1.0, in order to evaluate both the scenario without filtering (zero threshold) and the case in which only predictions with maximum confidence are considered. The comparative results of these configurations are presented in Table 9.

As the confidence threshold increases, a significant reduction in the number of false positives is observed, resulting in higher model precision. However, this gain comes at the expense of an increase in false negatives, which negatively affects recall.

Thus, it is essential to seek a balance between precision and recall through the appropriate selection of the confidence threshold. For DINO, in this dataset, a threshold of 0.9 yielded the highest F-score value. However, thresholds close to 1.0 resulted in no predictions with



(a)

(b)

(c)

Figure 11: Detection and segmentation results prior to the application of filtering methods. (a) Crop of the original image; (b) All DINO predictions; (c) All Segformer B3 predictions.

maximum confidence, which led to an F-score value of zero. Therefore, the selection of the ideal threshold depends both on the characteristics of the dataset and on the model and the behavior of the evaluated metrics.

For Segformer B3, high confidence thresholds also reduce the number of true positives. Considering that the prediction score is calculated based on the ratio between the number of object pixels and the area of the bounding box, better performance is observed for thresholds below 0.9.

To illustrate the effect of the confidence score filter on the predictions, Figure 12 presents a visual comparison of the results before and after applying the filter. The threshold values used were 0.9 for DINO and 0.8 for Segformer B3.

As shown in Figure 12, although the application of the confidence filter reduces the number of false positives, this strategy alone does not guarantee the proper detection of the objects

Table 9: Variation of the confidence score threshold for predictions by the DINO and Segformer B3 models.

Thr.	**Dino**						**Segformer B3**					
	FN	**FP**	**TP**	**P**	**R**	**F**	**FN**	**FP**	**TP**	**P**	**R**	**F**
0.0	13	2873405	1599	0.001	0.992	0.001	278	4240	1334	0.239	0.828	0.371
0.1	245	4883	1367	0.219	0.848	0.348	278	4240	1334	0.239	0.828	0.371
0.2	285	4003	1327	0.249	0.823	0.382	278	4240	1334	0.239	0.828	0.371
0.3	304	3766	1308	0.258	0.811	0.391	278	4240	1334	0.239	0.828	0.371
0.4	320	3645	1292	0.262	0.801	0.395	278	4237	1334	0.239	0.828	0.371
0.5	332	3538	1280	0.266	0.794	0.398	279	4221	1333	0.240	0.827	0.372
0.6	347	3455	1265	0.268	0.785	0.400	284	4146	1328	0.243	0.824	0.375
0.7	368	3363	1244	0.270	0.772	0.400	300	3938	1312	0.250	0.814	0.382
0.8	393	3206	1219	0.275	0.756	0.404	416	3050	1196	0.282	0.742	0.408
0.9	474	2763	1138	0.292	0.706	0.413	1265	323	347	0.518	0.215	0.304
1.0	1612	0	0	0.000	0.000	0.000	1586	43	26	0.377	0.016	0.031

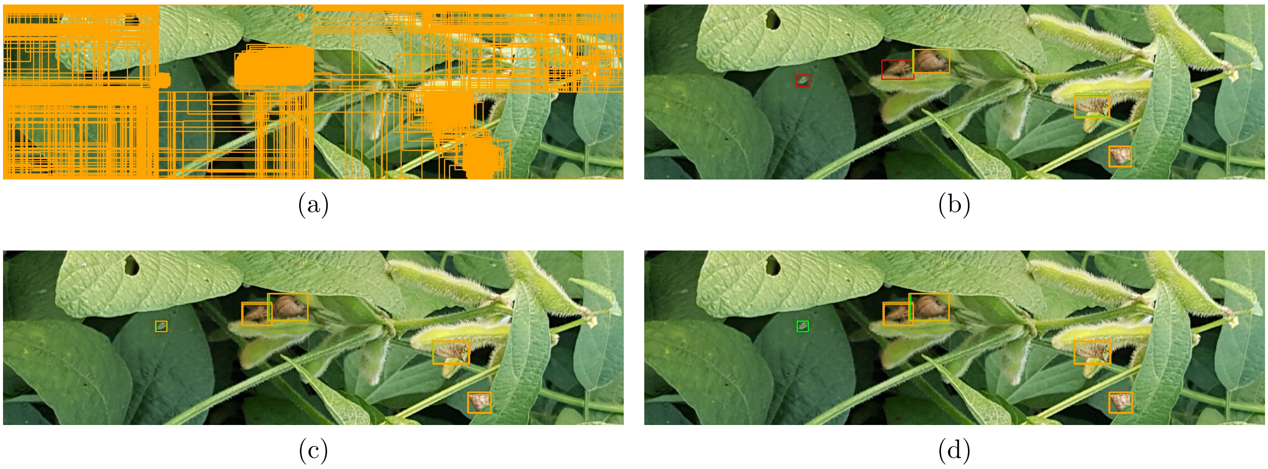


Figure 12: Detection and segmentation results after applying the confidence score filter. (a) All DINO predictions; (b) DINO bounding boxes after applying the confidence score filter; (c) All Segformer B3 predictions; (d) Segformer B3 bounding boxes after applying the confidence score filter;

of interest. In the case of DINO, for example, higher threshold values reduce false positives but increase false negatives, making object identification more challenging. Thus, for the following experiments, a confidence score threshold of 0.5 will be adopted for both detection and segmentation.

In addition to the confidence score filter, a criterion is also applied that discards bounding boxes whose dimensions are below the minimum values established for each category. To define these reference values, the dataset annotations were analyzed to identify the smallest recorded width and height for each object class. Table 10 presents a comparison of metric performance before and after applying the minimum dimension filter.

The results presented in Table 10 indicate that, for DINO, no changes were observed in the evaluated metrics. In contrast, for Segformer B3, the application of the minimum dimension filter reduced the number of false positives and increased precision, although it caused a decrease in recall, since some of the smaller objects were discarded.

Therefore, the adoption of this filter should be carefully considered according to the objectives of the study. In certain contexts, it may be more relevant to detect as many objects

Table 10: Performance metrics for DINO and Segformer B3 before and after applying the minimum dimension filter.

Filter	Dino						Segformer B3					
	FN	FP	TP	P	R	F	FN	FP	TP	P	R	F
No filter	332	3538	1280	0.266	0.794	0.398	279	4221	1333	0.240	0.827	0.372
With filter	332	3534	1280	0.266	0.794	0.398	309	3932	1303	0.249	0.808	0.381

as possible, even with lower precision. In others, the priority may be to obtain more reliable detections, even if not all objects are recognized. There are also scenarios in which balancing precision and recall is the most appropriate approach, aiming to optimize the F-score value. Thus, the definition of filtering parameters should be aligned with the specific needs of each application.

After applying the minimum bounding box dimension filter, the next post-processing step consists of using the NMS technique. This technique can be implemented in different ways, depending on the objectives of the experiment. One possibility is to apply NMS separately for each class, preserving only the most relevant detections in each category. Alternatively, it is possible to perform NMS globally, considering all classes at once, which eliminates redundant predictions regardless of the category.

Another important aspect concerns the criterion adopted for selecting which boxes will be retained during NMS. It is possible to prioritize predictions with higher confidence scores or choose to keep those with the largest bounding box area. The impact of these NMS application strategies can be observed in Table 11, which presents the performance metrics obtained for each analyzed combination.

Table 11: Performance metrics for different NMS application methods.

Method	DINO						Segformer B3					
	FN	FP	TP	P	R	F	FN	FP	TP	P	R	F
By score and class	343	487	1269	0.723	0.787	0.754	323	521	1289	0.712	0.800	0.753
By area and class	344	479	1268	0.726	0.787	0.755	320	510	1292	0.717	0.801	0.757
By score, ignoring class	353	464	1259	0.731	0.781	0.755	326	502	1286	0.719	0.798	0.756
By area, ignoring class	357	459	1255	0.732	0.779	0.755	324	493	1288	0.723	0.799	0.759

As shown in Table 11, the application of NMS contributes to balancing the F-score value, mainly by significantly reducing the number of false positives compared to Table 10. For DINO, no significant differences are observed between the evaluated NMS strategies. In contrast, for Segformer B3, a slight improvement in performance is noted when using the strategy based on the largest object area, regardless of class.

The final filter applied in post-processing aims to eliminate redundant bounding boxes that may still remain after the application of NMS. This removal is performed based on a predefined overlap percentage. When the overlap percentage between two bounding boxes is equal to or greater than this threshold, the box with the smaller area is discarded. The effects of this filter on the performance metrics are presented in Table 12.

The results presented in Table 12 highlight how varying the overlap threshold affects the final step of discarding bounding boxes. Lower thresholds eliminate a greater number of overlapping boxes, since any intersection is sufficient to discard one of them. On the other hand, higher thresholds remove only boxes with high overlap percentages, so that when using the maximum

Table 12: Variation of the overlap percentage between bounding boxes.

Method	Thr.	Dino						Segformer B3					
		FN	FP	TP	P	R	F	FN	FP	TP	P	R	F
By class	0.0	363	390	1249	0.762	0.775	0.768	328	366	1284	0.778	0.797	0.787
	0.1	362	390	1250	0.762	0.775	0.769	328	366	1284	0.778	0.797	0.787
	0.2	361	390	1251	0.762	0.776	0.769	328	366	1284	0.778	0.797	0.787
	0.3	360	390	1252	0.762	0.777	0.770	328	367	1284	0.778	0.797	0.787
	0.4	359	391	1253	0.762	0.777	0.770	328	368	1284	0.777	0.797	0.787
	0.5	359	396	1253	0.760	0.777	0.768	328	369	1284	0.777	0.797	0.787
	0.6	359	397	1253	0.759	0.777	0.768	328	372	1284	0.775	0.797	0.786
	0.7	359	400	1253	0.758	0.777	0.768	328	373	1284	0.775	0.797	0.786
	0.8	359	410	1253	0.753	0.777	0.765	328	378	1284	0.773	0.797	0.784
	0.9	358	420	1254	0.749	0.778	0.763	328	384	1284	0.770	0.797	0.783
	1.0	357	459	1255	0.732	0.779	0.755	324	493	1288	0.723	0.799	0.759
Ignoring class	0.0	366	387	1246	0.763	0.773	0.768	335	357	1277	0.782	0.792	0.787
	0.1	365	387	1247	0.763	0.774	0.768	335	357	1277	0.782	0.792	0.787
	0.2	364	387	1248	0.763	0.774	0.769	334	357	1278	0.782	0.793	0.787
	0.3	362	387	1250	0.764	0.775	0.769	334	358	1278	0.781	0.793	0.787
	0.4	361	388	1251	0.763	0.776	0.770	334	359	1278	0.781	0.781	0.787
	0.5	360	393	1252	0.761	0.777	0.769	334	360	1278	0.780	0.793	0.786
	0.6	360	395	1252	0.760	0.777	0.768	333	363	1279	0.779	0.793	0.786
	0.7	360	398	1252	0.759	0.777	0.768	333	364	1279	0.778	0.793	0.786
	0.8	360	408	1252	0.754	0.777	0.765	333	369	1279	0.776	0.793	0.785
	0.9	358	418	1254	0.750	0.778	0.764	331	377	1281	0.773	0.795	0.783
	1.0	357	459	1255	0.732	0.779	0.755	324	493	1288	0.723	0.799	0.759

value, only boxes that are completely contained within others are discarded.

Comparing the evaluated models, the impact of this filtering is most noticeable in the number of false positives. As the threshold increases, more boxes are retained, raising this index, especially for Segformer B3, regardless of class consideration. To illustrate the behavior of this filter, Figure 13 presents examples of bounding boxes discarded at different overlap ranges.

Figure 13 illustrates predictions with different overlap percentages, ranging from less than 10% to 100%. The green bounding boxes represent those that are retained after filtering, as they have larger areas, while the red boxes are discarded for having smaller areas. It is noteworthy that for very low overlap thresholds (Subfigures 13a-13d), true detections may be eliminated, increasing the number of false negatives. Additionally, when ignoring the object class, there is also a risk of discarding correct detections, as illustrated in Subfigure 13b, where an Euschistus class insect was removed due to its proximity to another object. On the other hand, higher thresholds for the overlap percentage can be useful for eliminating object fragments, which are often generated during the image cropping process.

In general, the choice of filters and their respective thresholds directly impacts the set of predictions used in the subsequent step, which is dedicated to calculating the metrics based on the center of the bounding boxes. Thus, the appropriate selection of thresholds should consider the specific objective of the application, whether it is to prioritize the detection of as many objects as possible, maximize detection precision, or balance these aspects. Moreover, visual comparisons between retained and discarded predictions can assist in defining the most suitable parameters for each scenario.

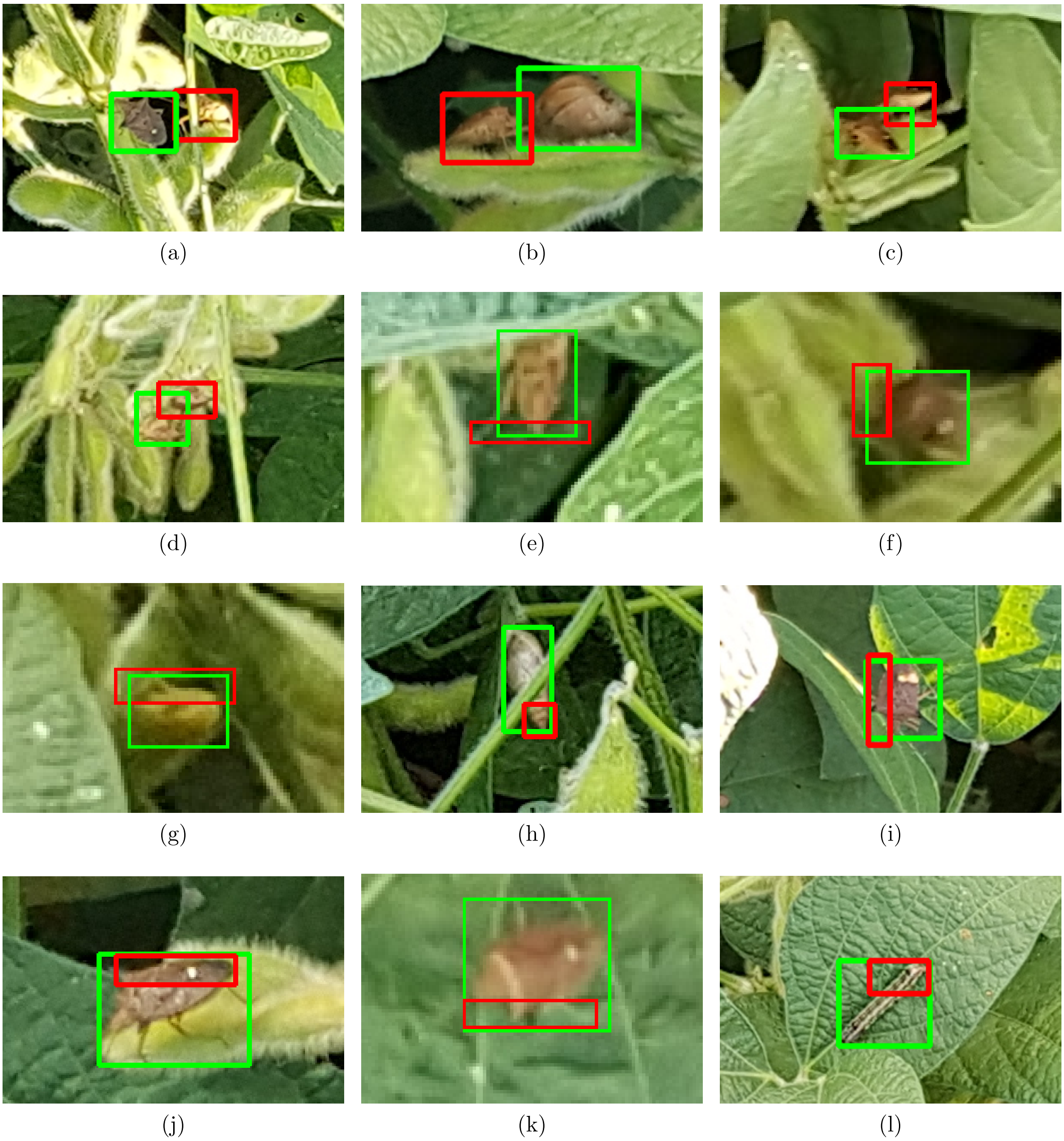


Figure 13: Bounding boxes with different overlap percentages. (a) Overlap less than 10%; (b) Overlap between 10 and 20%; (c) Overlap between 20 and 30%; (d) Overlap between 30 and 40%; (e) Overlap between 40 and 50%; (f) Overlap between 50 and 60%; (g) Overlap between 60 and 70%; (h) Overlap between 70 and 80%; (i) Overlap between 80 and 90%; (j) - (l) Complete overlap (100%).

5.2.2. Center-Based Matching Metric

Previous experiments were conducted using a threshold of 15 pixels between the centers of the annotated and predicted bounding boxes. However, this value can be adjusted depending on the nature of the problem or the expected behavior of the model.

This subsection presents the results obtained using different distance thresholds for the center-based matching technique, in order to evaluate how this variation influences model performance in small object detection tasks. The performance metrics considered include precision, recall, and F-score.

In addition, the results obtained with the proposed technique are compared with those of the traditional IoU-based approach, highlighting its limitations in scenarios involving small objects. Table 13 summarizes the main results for different distance thresholds, as well as for IoU with a minimum threshold of 0.5, enabling a direct comparison between the approaches.

Table 13: Results obtained with the center-based distance metric and IoU.

Method	Thr.	Dino						Segformer B3					
		FN	FP	TP	P	R	F	FN	FP	TP	P	R	F
Distance	0	1586	1619	26	0.016	0.016	0.016	1599	1625	13	0.008	0.008	0.008
	5	476	509	1136	0.691	0.705	0.698	454	480	1158	0.707	0.718	0.713
	10	381	414	1231	0.748	0.764	0.756	360	386	1252	0.764	0.777	0.770
	15	360	393	1252	0.761	0.777	0.769	334	360	1278	0.780	0.793	0.786
	20	352	385	1260	0.766	0.782	0.774	325	351	1287	0.786	0.798	0.792
	25	348	381	1264	0.768	0.784	0.776	321	347	1291	0.788	0.801	0.794
IoU	0.5	395	428	1217	0.740	0.755	0.747	416	442	1196	0.730	0.742	0.736
	0.6	481	514	1131	0.688	0.702	0.695	544	570	1068	0.652	0.663	0.657
	0.7	692	725	920	0.559	0.571	0.565	800	826	812	0.496	0.504	0.500
	0.8	1090	1123	522	0.317	0.324	0.321	1214	1240	398	0.243	0.247	0.245
	0.9	1489	1522	123	0.075	0.076	0.076	1553	1579	59	0.036	0.037	0.036
	1.0	1609	1642	3	0.002	0.002	0.002	1612	1638	0	0.000	0.000	0.000

The results presented in Table 13 show that the center-based distance metric is more tolerant to small spatial variations. When the distance threshold is extremely strict, as in the case of zero, the centers of the predicted and annotated bounding boxes must be exactly aligned. This rigid condition leads to a high incidence of false negatives and false positives for both the DINO and Segformer B3 models, resulting in low values for precision, recall, and consequently, the F-score. As the distance threshold becomes more flexible, an increase in true positives and a reduction in false negatives and false positives is observed, contributing to a better balance among the performance metrics.

On the other hand, when using IoU as the evaluation criterion, the best performance is observed at the most commonly adopted threshold of 0.5, which is widely used in benchmarks such as COCO and Pascal VOC. However, this metric proves to be highly sensitive to small variations in the threshold. A modest increase from 0.5 to 0.6 already causes a significant drop in the F-score, decreasing from 0.747 to 0.695 for the DINO model, and from 0.736 to 0.657 for Segformer B3. This degradation trend becomes more pronounced at higher thresholds, highlighting the limitation of IoU in handling small spatial inaccuracies, particularly in the detection of small objects.

Figure 14 illustrates the differences between the evaluation criteria based on center distance and IoU. Blue bounding boxes represent the manual annotations from the test set. Green boxes indicate predictions considered true positives according to the evaluation criterion, while red boxes correspond to predictions classified as false.

In Figure 14, from left to right, the first column shows different insect samples from the test set. The second column displays the predictions considered true positives by the center-based distance metric with a threshold of 15 pixels, in which the centers of the annotated and predicted bounding boxes are highlighted. The third and final column presents the same predictions, but classified as false positives when using the IoU metric with a threshold of 0.5.

It is observed that several visually correct detections are accepted by the center-based distance metric but rejected by IoU, even when they are positioned close to the annotated

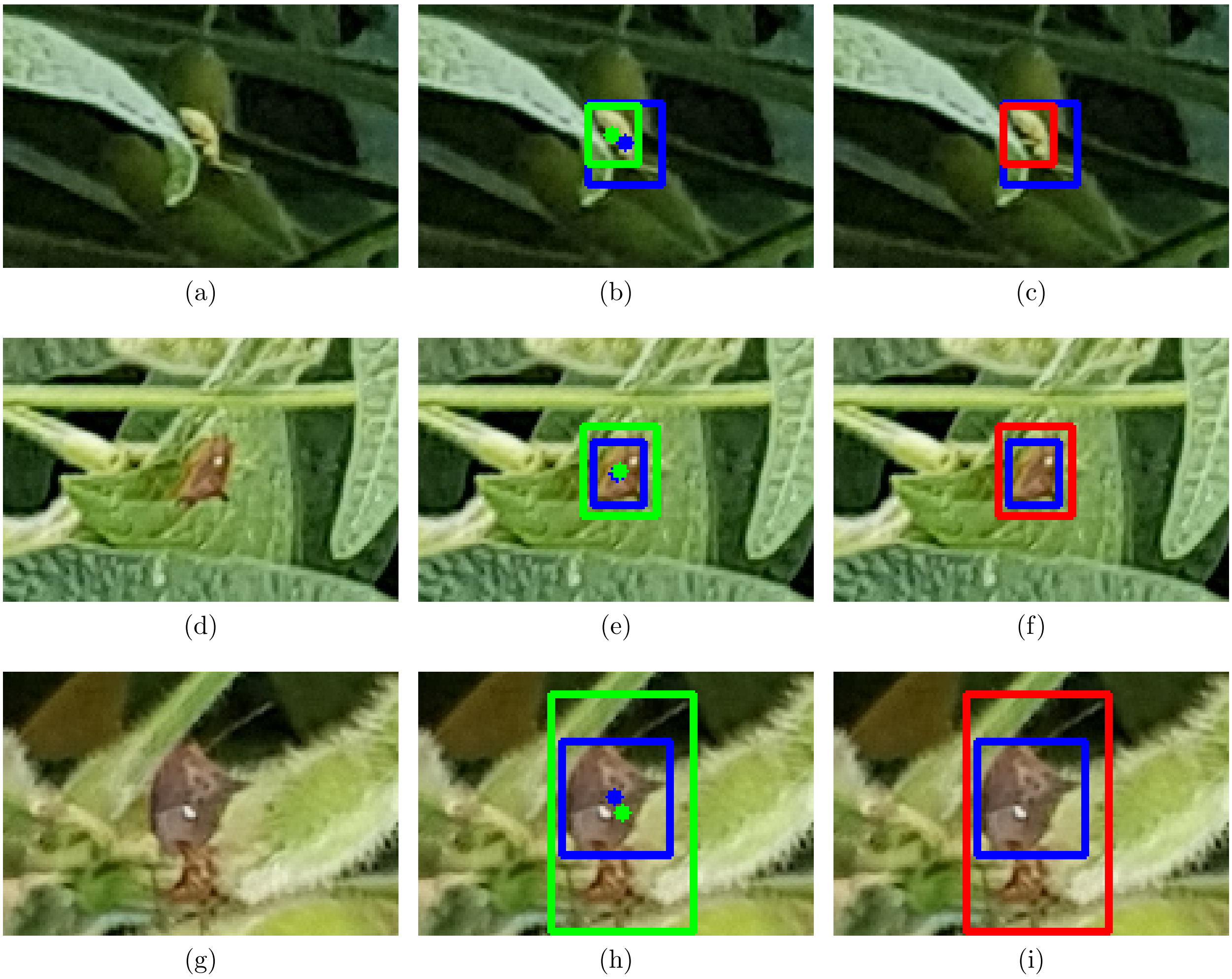


Figure 14: Matches between annotations and predictions using center-based distance and IoU. ((a), (d), (g)) Insect samples from the test set; ((b), (e), (h)) Predictions considered as positives using the center-based distance metric; ((c), (f), (i)) Predictions considered as negatives using IoU.

object. This behavior highlights a significant limitation of IoU in the evaluation of small objects, where minor spatial inaccuracies can lead to the rejection of relevant predictions.

5.2.3. Crops vs. Reconstructed Image

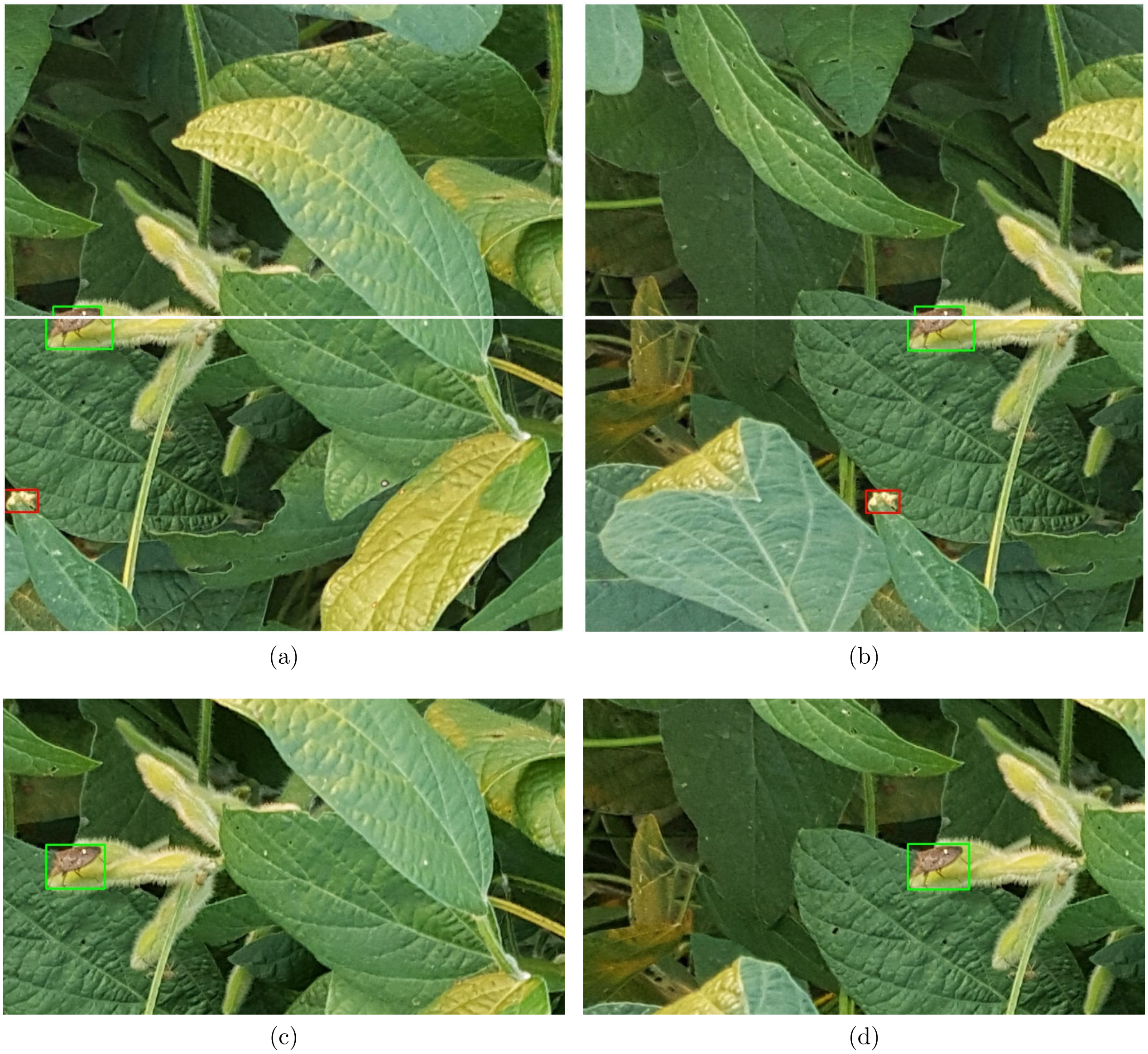
The results discussed in the previous section were obtained by calculating the metrics after the image reconstruction process, taking into account the coordinates of the bounding boxes in relation to the entire image. In this section, we provide an overview of the performance of the methods by considering only the individual crops. Each crop is treated as an independent image, with no connection to other crops that originated from the same original image. As a result, the metrics are calculated directly on the detected objects within each crop, without accounting for possible overlaps or duplications that might be resolved through the reconstruction process. Table 14 presents the results obtained from different grid configurations applied to the images, making it possible to compare the methods evaluated in this scenario.

It can be observed that, compared to the results obtained after image reconstruction, the number of true positives increased considerably when the metrics are calculated directly on the crops. This increase is a consequence of the fact that, from each test image, 81 distinct crops were generated. Since these crops were not subsequently merged, each crop, along with its corresponding objects, is treated as an independent image, which raises the number of

Table 14: Metrics calculated on crops obtained from 5 × 5, 4 × 5, 5 × 4, and 4 × 4 grids.

DINO						Segformer B3					
FN	**FP**	**TP**	**P**	**R**	**F**	**FN**	**FP**	**TP**	**P**	**R**	**F**
1712	762	4044	0.841	0.703	0.766	1421	879	4335	0.831	0.753	0.790

annotations and predictions considered. On the other hand, this approach also increases the likelihood that objects will be divided into multiple parts, which leads to a higher number of false negatives, especially for the DINO model. These aspects can be seen more clearly in Figure 15, which presents examples of the models behavior when operating solely on individual crops.


(a)

(b)

(c)

(d)

Figure 15: Detections for different grid configurations. (a) Detections with a 4 × 4 grid, with objects split between crops; (b) Detections with a 4 × 5 grid, with objects split between crops; (c) Detection with a 5 × 4 grid, with the object kept whole; (d) Detection with a 5 × 5 grid, with the object kept whole.

Figure 15 illustrates the impact of using crops in the evaluation of detection. For example, it can be seen that a single insect from the Euschistus class resulted in six distinct detections due to the overlap of crops and the division of the object along the grid lines, which are indicated in white in Subfigures 15a and 15b. In these configurations, the object was correctly identified in both parts, resulting in four true positives. On the other hand, it is also apparent that another object present in the scene was not detected, which led to two false negatives. Considering only the metrics calculated over the crops, in this example there would be two false negatives, no false positives, and six true positives, resulting in a precision of 1.0, recall of 0.75, and F-score of 0.857. In contrast, when evaluating the same scene using the entire image, there would be a single false negative, no false positives, and only one true positive, yielding a precision of 1.0, recall of 0.5, and F-score of 0.667. These results demonstrate that an evaluation based solely on crops can overestimate the number of true positives and distort performance metrics, reinforcing the need to reconstruct the image in order to obtain an analysis that more accurately reflects real-world applications.

Overall, the results show that although the use of crops leads to an increase in the number of false negatives, it also provides a significant gain in the number of true positives. This indicates that the model is capable of identifying multiple occurrences of the object of interest, even when the same insect is detected in different parts of the image due to fragmentation caused by the grid divisions. Therefore, crop-based evaluation reveals the model ability to recognize isolated parts of the objects, which can be relevant in scenarios with high fragmentation or overlap. In this context, the choice of evaluation strategy should take into account the specific context and objectives of the application.

5.3. Comparison Between Object Detection and Segmentation Approaches

This section presents the results obtained by object detection and segmentation models, covering both conventional architectures and Transformer-based approaches. The comparison between methods was conducted using the previously established metrics, including false negatives, false positives, true positives, precision, recall, and F-score. For all models, intermediate threshold values were adopted for the filters, using 0.5 as the threshold for both the confidence score and the overlap percentage (regardless of class), in addition to applying area-based NMS, also ignoring class, and discarding predictions whose bounding boxes diverged from the annotation patterns. The results obtained for each model on the test set are presented in Table 15.

Table 15 provides an overview of the results obtained by the different models trained on the insect dataset. Among the detection methods, conventional models such as Faster R-CNN, RetinaNet, and YOLOv3 achieved F-score values greater than 0.75, indicating consistent performance for the proposed task. Among the Transformer-based models, DINO stands out, not only for achieving better results compared to DAB DETR, but also for presenting a suitable balance between precision and recall. In contrast, DAB DETR faced challenges during training, resulting in inferior performance and suggesting the need for further experiments and hyperparameter adjustments to improve its effectiveness.

In the context of segmentation, it can be observed that the Segformer architectures considerably outperformed the DeepLab models, whose results revealed limitations, especially in the identification of minority classes and small objects. Among the Segformer variants, the larger architectures achieved higher F-score values. However, the B3 architecture stood out by presenting the best balance between precision and recall.

To provide a more detailed illustration of the benefits and limitations of detection and segmentation approaches, as well as the performance of the different models evaluated, Figure 16 presents an example of a test set image crop along with its corresponding annotation mask. In this mask, each color represents a different insect species, allowing for a clear visualization of

Table 15: Metrics obtained for different object detection and segmentation models.

Method	Model	Metrics					
		FN	FP	TP	P	R	F
Detection	Faster R-CNN	301	511	1311	0.720	0.813	0.764
	Retinanet	332	485	1280	0.725	0.794	0.758
	Yolo V3	438	340	1174	0.775	0.728	0.751
	DAB DETR	743	357	869	0.709	0.539	0.612
	DINO	360	393	1252	0.761	0.777	0.769
Segmentation	DeepLab V3	670	511	942	0.648	0.584	0.615
	DeepLab V3+	639	714	973	0.577	0.604	0.590
	Segformer B0	363	559	1249	0.691	0.775	0.730
	Segformer B1	348	425	1264	0.748	0.784	0.766
	Segformer B2	311	402	1301	0.764	0.807	0.785
	Segformer B3	334	360	1278	0.780	0.793	0.786
	Segformer B4	327	405	1285	0.760	0.797	0.778
	Segformer B5	339	372	1273	0.774	0.790	0.782

how the various classes are distributed within the crop. This example will be used as the basis for the analysis of the results for both approaches.

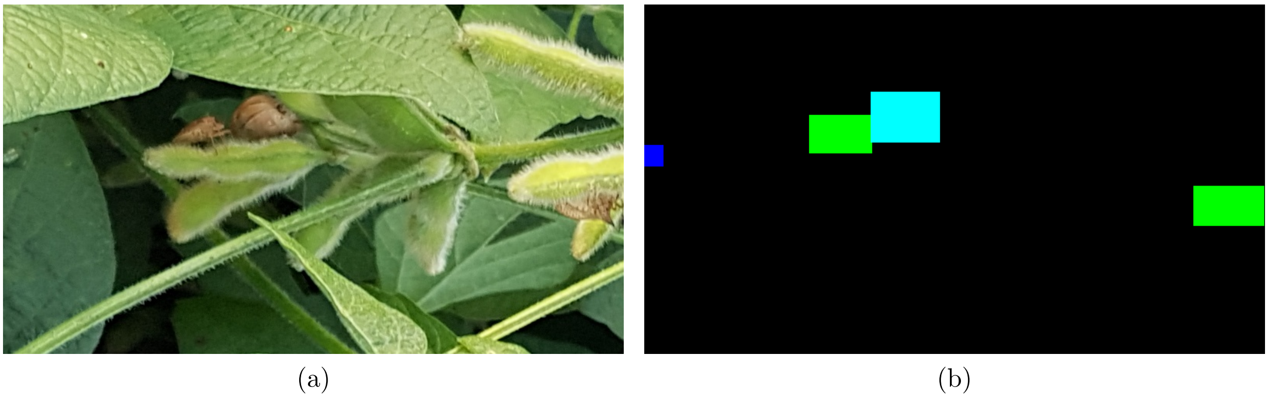
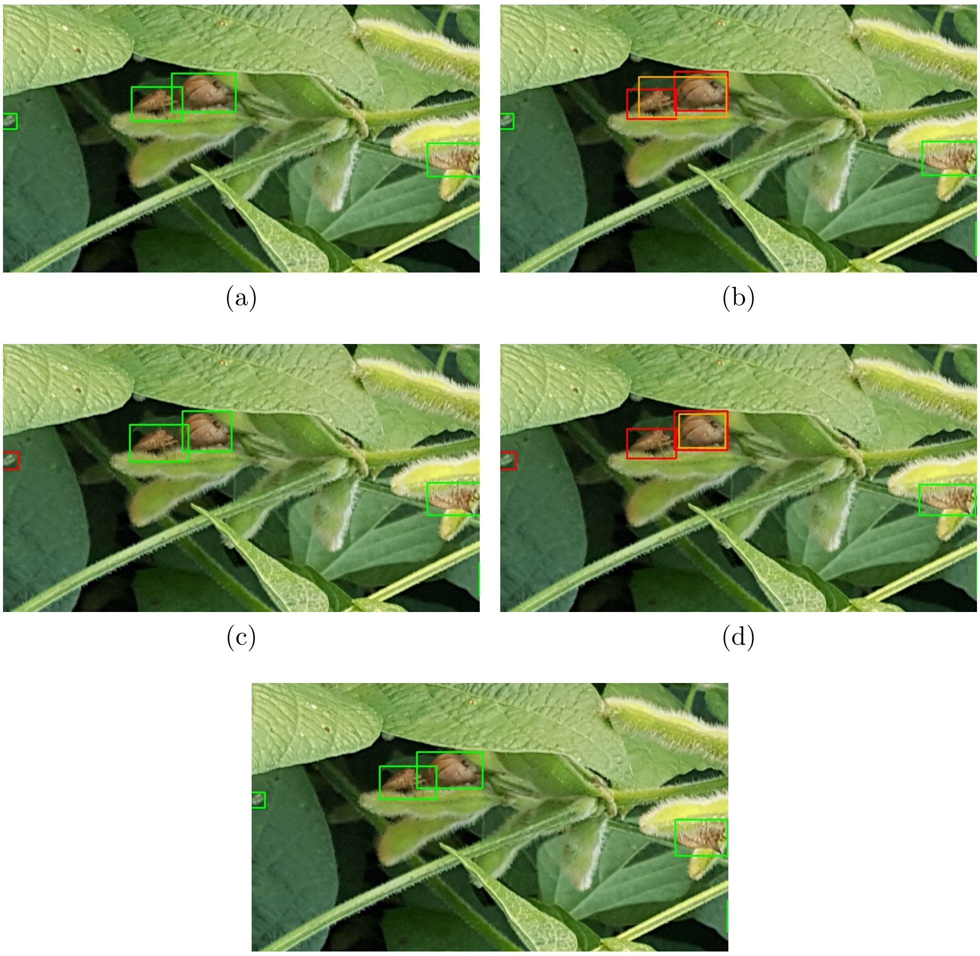


Figure 16: Crop and its corresponding mask. (a) Crop representation; (b) Corresponding mask, where blue represents the Diabrotica class, light blue represents Gastropoda, and green represents Euschistus.

Figure 16 shows a crop from the test set containing four insects from different classes, one from the Diabrotica class, one from the Gastropoda class, and two from the Euschistus class, each identified by its respective mask in the annotation image. Based on this crop, we aim to analyze and compare the predictions generated by the different detection models, placing their outputs side by side to facilitate qualitative evaluation. This specific case is particularly relevant as it brings together a diversity of classes within a single crop area and presents two objects from different classes positioned very close to each other, with their masks adjacent. Figure 17 presents the predictions of the main detection models evaluated for the same crop shown in Subfigure 16a. Different colors indicate the outcome of each prediction, where green bounding boxes represent true positives, orange boxes indicate false positives, and red boxes denote false negatives.

In Figure 17, it can be observed that the Faster R-CNN (Subfigure 17a) and DINO (Sub-



(a)

(b)

(c)

(d)

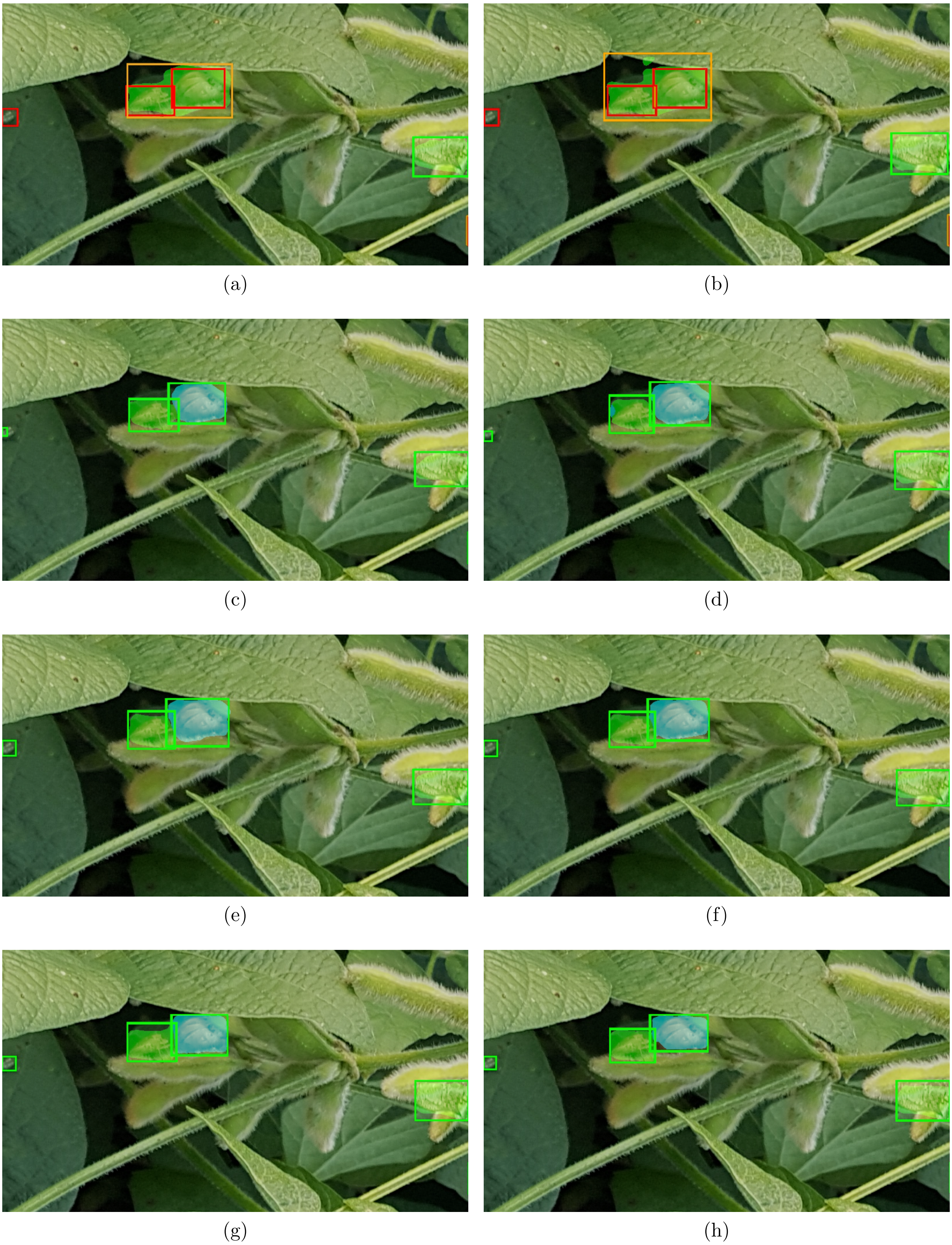
(e)

Figure 17: Detection model predictions. (a) Predictions from Faster R-CNN; (b) Predictions from RetinaNet; (c) Predictions from YOLOv3; (d) Predictions from DAB DETR; (e) Predictions from DINO.

figure 17e) models showed exemplary performance in this crop, correctly identifying all insects present, both in terms of localization and classification of the different categories. In contrast, the RetinaNet (Subfigure 17b) and DAB DETR (Subfigure 17d) models exhibited limitations, resulting in false positives and false negatives to Euschistus and Gastropoda classes. In the case of RetinaNet, the predicted area encompassed two insects of different classes, whereas DAB DETR was able to correctly detect only one of the objects present. The YOLOv3 model (Subfigure 17c) also showed satisfactory performance in most cases, but failed to detect the smallest insect in the set, which belongs to the Diabrotica class.

In general, the bounding boxes generated by the detection methods enabled the individual identification of nearby objects. However, these boxes often also include regions of the image background, which can lead to an overestimation of the area occupied by the object, especially when the object is oriented diagonally. On the other hand, segmentation-based approaches offer the advantage of classifying only the pixels that actually belong to the object of interest, thus providing a more precise localization in the image. In this context, Figure 18 illustrates the results obtained by the segmentation models, presenting both the generated masks and the respective bounding boxes resulting from post-processing.

In Figure 18, the predictions generated by the different segmentation models evaluated in this study are presented It can be observed that the DeepLab family models, shown in Subfig-



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 18: Segmentation model predictions with masks and bounding boxes. (a) Predictions from DeepLabV3; (b) Predictions from DeepLabV3 Plus; (c) Predictions from Segformer B0; (d) Predictions from Segformer B1; (e) Predictions from Segformer B2; (f) Predictions from Segformer B3; (g) Predictions from Segformer B4; (h) Predictions from Segformer B5.

ures 18a and 18b, demonstrated limitations when segmenting the Diabrotica class insect, which is the smallest object in the dataset. Additionally, these models were unable to correctly distinguish between the pixels of the Gastropoda and Euschistus class insects (located in the center of the image). As a result, the region corresponding to two distinct insects was erroneously classified as a single instance of the Euschistus class, leading to one false positive and two false negatives in the evaluation.

In contrast, the Segformer family models (Subfigures 18c–18h) demonstrated superior performance. Even the simplest variants of this architecture were able to correctly segment the Diabrotica class insect, as well as accurately distinguish the pixels belonging to the Gastropoda and Euschistus classes. This enabled all detections in these scenarios to be correctly classified as true positives, including cases with small objects and also with different classes of objects located close to each other.

Despite their superior performance, segmentation approaches still present an important limitation related to the merging of masks when two objects are close together or in direct contact, especially when they belong to the same category. This issue can result in incorrect counting, since multiple instances may be interpreted as a single one, particularly in scenarios with higher object density. Furthermore, the comparison between detection and segmentation approaches required the development of additional procedures, such as converting bounding boxes into masks and, subsequently, reconverting the predictions back into bounding boxes to standardize the evaluated metrics. These steps increase processing time and may introduce minor inaccuracies. Nevertheless, these adaptations were essential to ensure a fair and direct comparison between the different methods.

Given the results presented, it is clear that the choice between detection and segmentation methods should be guided both by the specific characteristics of the objects to be identified and by the requirements of the final application. While detection approaches have shown a greater ability to differentiate nearby objects, segmentation approaches have provided higher spatial accuracy in delineating the insects. However, segmentation still faces challenges in scenarios with high object density or close proximity between instances, which may affect individual counting. Therefore, the decision on the most suitable approach should consider the balance among the evaluated metrics as well as the limitations observed in each context.

6. Conclusions

This work presented and evaluated strategies for the detection and segmentation of small objects in high-resolution images, focusing on agricultural and entomological contexts. First, a dataset originally segmented by superpixels was adapted, allowing its use in detection and segmentation tasks by transforming the annotations into adjusted bounding boxes. Additionally, preprocessing techniques based on image patches were proposed, enabling better preservation of small objects during the training and inference stages of the models. Post-processing strategies were also implemented, including an adaptation of the NMS algorithm and an alternative metric to IoU, based on the distance between centers, aiming to reduce the limitations faced by traditional methods when dealing with small objects.

The results showed that the proposed approaches contribute to improving the detection and segmentation of insects on soybean leaves, highlighting the potential application of these techniques to similar problems involving small objects. However, some limitations were observed, such as increased processing time due to image splitting and reconstruction, the need for further adaptations to support rotated bounding boxes, and the restriction of analyses to the agricultural context, specific detection and segmentation models, and only one adapted dataset.

As future perspectives, we suggest optimizing the performance of the techniques, focusing on reducing processing time, extending the strategies to support rotated bounding boxes, and evaluating them in different domains and datasets, such as urban, medical, or maritime images. In this way, it is expected that the contributions of this study will serve as a basis for the development of increasingly robust and efficient solutions for small object detection in a variety of application contexts.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Finance Code 001

References

[1] Z. Lin, Q. Wu, S. Fu, S. Wang, Z. Zhang, Y. Kong, Dual-nms: A method for autonomously removing false detection boxes from aerial image object detection results, Sensors 19 (21) (2019) 4691. doi:10.3390/s19214691.

[2] B. Wang, Y. Gu, An improved fbpn-based detection network for vehicles in aerial images, Sensors 20 (17) (2020) 4709. doi:10.3390/s20174709.

[3] Z. Wang, D. Liu, Y. Lei, X. Niu, S. Wang, L. Shi, Small target detection based on bird's visual information processing mechanism, Multimedia Tools and Applications 79 (2020) 22083–22105. doi:10.1007/s11042-020-08807-8.

[4] X. Ye, F. Xiong, J. Lu, J. Zhou, Y. Qian, F3-net: Feature fusion and filtration network for object detection in optical remote sensing images, Remote Sensing 12 (24) (2020). doi:10.3390/rs12244027.
URL https://www.mdpi.com/2072-4292/12/24/4027

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, arXiv preprint arXiv:2010.11929 (2020).

[6] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: European conference on computer vision, Springer, 2020, pp. 213–229.

[7] Y. Zhang, Y. Bai, M. Ding, B. Ghanem, Multi-task generative adversarial network for detecting small objects in the wild, International Journal of Computer Vision 128 (6) (2020) 1810–1828. doi:10.1007/s11263-020-01301-6.

[8] S. Ren, K. He, R. B. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, CoRR abs/1506.01497 (2015). arXiv:1506.01497.
URL http://arxiv.org/abs/1506.01497

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, Ssd: Single shot multibox detector, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, Springer, 2016, pp. 21–37.

[10] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

[11] D. Jiang, B. Sun, S. Su, Z. Zuo, P. Wu, X. Tan, Fassd: A feature fusion and spatial attention-based single shot detector for small object detection.

[12] X. Jiang, A. Hadid, Y. Pang, E. Granger, X. Feng, Deep Learning in object detection and recognition, Springer, 2019.

[13] Z. Liu, D. Li, S. S. Ge, F. Tian, Small traffic sign detection from large image, Applied Intelligence 50 (1) (2019) 1–13. doi:10.1007/s10489-019-01511-7.

[14] Y. Zhang, T. Shen, Small object detection with multiple receptive fields, in: IOP Conference Series: Earth and Environmental Science, Vol. 440, IOP Publishing, 2020, p. 032093.

[15] T. Gao, M. Wushouer, G. Tuerhong, Dms-yolov5: A decoupled multi-scale yolov5 method for small object detection, Applied Sciences 13 (10) (2023) 6124.

[16] N. Jia, Z. Wei, B. Li, Attention-enhanced lightweight one-stage detection algorithm for small objects, Electronics 12 (7) (2023) 1607.

[17] H. Cui, Z. Wei, Multi-scale receptive field detection network, IEEE Access 7 (2019) 138825–138832.

[18] Z. Wu, Detect small object based on fcos and adaptive feature fusion, in: Journal of Physics: Conference Series, Vol. 2580, IOP Publishing, 2023, p. 012005.

[19] Y. Zhang, Y. Bai, M. Ding, B. Ghanem, Multi-task generative adversarial network for detecting small objects in the wild, International Journal of Computer Vision 128 (6) (2020) 1810–1828.

[20] J. Liu, S. Yang, L. Tian, W. Guo, B. Zhou, J. Jia, H. Ling, Multi-component fusion network for small object detection in remote sensing images, IEEE Access 7 (2019) 128339–128352.

[21] X. Zhang, Q. Liu, H. Chang, H. Sun, High-resolution network with transformer embedding parallel detection for small object detection in optical remote sensing images, Remote Sensing 15 (18) (2023) 4497.

[22] Y. Feng, L. Wang, M. Zhang, A multi-scale target detection method for optical remote sensing images, Multimedia Tools and Applications 78 (7) (2019) 8751–8766.

[23] J. Yan, H. Wang, M. Yan, W. Diao, X. Sun, H. Li, Iou-adaptive deformable r-cnn: Make full use of iou for multi-class object detection in remote sensing imagery, Remote Sensing 11 (3) (2019). doi:10.3390/rs11030286.
URL https://www.mdpi.com/2072-4292/11/3/286

[24] Y. Gan, S. You, Z. Luo, K. Liu, T. Zhang, L. Du, Object detection in remote sensing images with mask r-cnn, in: Journal of Physics: Conference Series, Vol. 1673, IOP Publishing, 2020, p. 012040.

[25] Q. Ran, Q. Wang, B. Zhao, Y. Wu, S. Pu, Z. Li, Lightweight oriented object detection using multiscale context and enhanced channel attention in remote sensing images, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 14 (2021) 5786–5795.

[26] J. Pang, C. Li, J. Shi, Z. Xu, H. Feng, R2-cnn: Fast tiny object detection in large-scale remote sensing images. arxiv 2019, arXiv preprint arXiv:1902.06042 3 (2019).

[27] Z. Liu, X. Gao, Y. Wan, J. Wang, H. Lyu, An improved yolov5 method for small object detection in uav capture scenes, IEEE Access 11 (2023) 14365–14374.

[28] M. Liu, X. Wang, A. Zhou, X. Fu, Y. Ma, C. Piao, Uav-yolo: Small object detection on unmanned aerial vehicle perspective, Sensors 20 (8) (2020) 2238.

[29] L. Ding, Z. Q. Rao, B. Ding, S. J. Li, Research on defect detection method of railway transmission line insulators based on gc-yolo, IEEE Access (2023).

[30] S. Xu, Q. Feng, J. Fei, G. Zhao, X. Liu, H. Li, C. Lu, Q. Yang, A locating approach for small-sized components of railway catenary based on improved yolo with asymmetrically effective decoupled head, IEEE Access 11 (2023) 34870–34879.

[31] F. Saeed, M. J. Ahmed, M. J. Gul, K. J. Hong, A. Paul, M. S. Kavitha, A robust approach for industrial small-object detection using an improved faster regional convolutional neural network, Scientific reports 11 (1) (2021) 23390.

[32] S. Gao, C. Liu, H. Zhang, Z. Zhou, J. Qiu, Multiscale attention-based detection of tiny targets in aerial beach images, Frontiers in Marine Science 9 (2022) 1073615.

[33] F. Zhang, X. Hou, Multi-site and multi-scale unbalanced ship detection based on centernet, Electronics 11 (11) (2022) 1713.

[34] Z. Jiang, Y. Wang, X. Zhou, L. Chen, Y. Chang, D. Song, H. Shi, Small-scale ship detection for sar remote sensing images based on coordinate-aware mixed attention and spatial semantic joint context, Smart Cities 6 (3) (2023) 1612–1629.

[35] H. Takimoto, Y. Sato, A. J. Nagano, K. K. Shimizu, A. Kanagawa, Using a two-stage convolutional neural network to rapidly identify tiny herbivorous beetles in the field, Ecological Informatics 66 (2021) 101466.

[36] Q. Xiang, X. Huang, Z. Huang, X. Chen, J. Cheng, X. Tang, Yolo-pest: an insect pest object detection algorithm via cac3 module, Sensors 23 (6) (2023) 3221.

[37] V. Moskalenko, A. Moskalenko, A. Korobov, M. Zaretsky, A model and training algorithm of small-sized object detection system for a compact aerial drone, , , (2019) 110–121.

[38] M. Cui, G. Gong, G. Chen, H. Wang, M. Jin, W. Mao, H. Lu, Lc-yolo: A lightweight model with efficient utilization of limited detail features for small object detection, Applied Sciences 13 (5) (2023) 3174.

[39] H. Zhou, A. Ma, Y. Niu, Z. Ma, Small-object detection for uav-based images using a distance metric method, Drones 6 (10) (2022) 308.

[40] H. Zhang, L. Qin, J. Li, Y. Guo, Y. Zhou, J. Zhang, Z. Xu, Real-time detection method for small traffic signs based on yolov3, Ieee Access 8 (2020) 64145–64156.

[41] Y. Rehman, H. Amanullah, M. A. Shirazi, M. Y. Kim, Small traffic sign detection in big images: Searching needle in a hay, IEEE Access 10 (2022) 18667–18680.

[42] H. Lai, L. Chen, W. Liu, Z. Yan, S. Ye, Stc-yolo: small object detection network for traffic signs in complex environments, Sensors 23 (11) (2023) 5307.

[43] G. Ma, et al., A small target detection method based on the improved fcn model, Advances in Multimedia 2022 (2022).

[44] C. Cao, B. Wang, W. Zhang, X. Zeng, X. Yan, Z. Feng, Y. Liu, Z. Wu, An improved faster r-cnn for small object detection, Ieee Access 7 (2019) 106838–106846.

[45] J. Mu, S. Li, Z. Liu, Y. Zhou, Integration of gradient guidance and edge enhancement into super-resolution for small object detection in aerial images, IET Image Processing 15 (13) (2021) 3037–3052.

[46] X. Xu, Z. Feng, C. Cao, M. Li, J. Wu, Z. Wu, Y. Shang, S. Ye, An improved swin transformer-based model for remote sensing object detection and instance segmentation, Remote Sensing 13 (23) (2021) 4779.

[47] J. Zhang, J. Lei, W. Xie, Y. Li, G. Yang, X. Jia, Guided hybrid quantization for object detection in remote sensing imagery via one-to-one self-teaching, IEEE Transactions on Geoscience and Remote Sensing (2023).

[48] N. Bodla, B. Singh, R. Chellappa, L. S. Davis, Soft-nms–improving object detection with one line of code, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 5561–5569.

[49] M. Nikouei, B. Baroutian, S. Nabavi, F. Taraghi, A. Aghaei, A. Sajedi, M. E. Moghaddam, Small object detection: A comprehensive survey on challenges, techniques and real-world applications, arXiv preprint arXiv:2503.20516 (2025).

[50] G. Cheng, X. Yuan, X. Yao, K. Yan, Q. Zeng, X. Xie, J. Han, Towards large-scale small object detection: Survey and benchmarks, IEEE Transactions on Pattern Analysis and Machine Intelligence (2023).

[51] K. Li, G. Wan, G. Cheng, L. Meng, J. Han, Object detection in optical remote sensing images: A survey and a new benchmark, ISPRS journal of photogrammetry and remote sensing 159 (2020) 296–307.

[52] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, arXiv preprint arXiv:1804.02767 (2018).

[53] K. Zhao, X. Ren, Small aircraft detection in remote sensing images based on yolov3, in: IOP Conference Series: Materials Science and Engineering, Vol. 533, IOP Publishing, 2019, p. 012056.

[54] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, L. Zhang, Dota: A large-scale dataset for object detection in aerial images, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 3974–3983.

[55] L. Li, S. Zhang, J. Wu, Efficient object detection framework and hardware architecture for remote sensing images, Remote Sensing 11 (20) (2019) 2376.

[56] G. Cheng, J. Han, P. Zhou, L. Guo, Multi-class geospatial object detection and geographic image classification based on collection of part detectors, ISPRS Journal of Photogrammetry and Remote Sensing 98 (2014) 119–132.

[57] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.

[58] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, CoRR abs/1512.03385 (2015). `arXiv:1512.03385`.
URL `http://arxiv.org/abs/1512.03385`

[59] Z. Xiao, Q. Liu, G. Tang, X. Zhai, Elliptic fourier transformation-based histograms of oriented gradients for rotationally invariant object detection in remote-sensing images, International Journal of Remote Sensing 36 (2) (2015) 618–644.

[60] R. J. Wang, X. Li, C. X. Ling, Pelee: A real-time object detection system on mobile devices, Advances in neural information processing systems 31 (2018).

[61] A.-J. Gallego, A. Pertusa, P. Gil, Automatic ship classification from optical aerial images with convolutional neural networks, Remote Sensing 10 (4) (2018) 511.

[62] S. Razakarivony, F. Jurie, Vehicle detection in aerial imagery: A small target detection benchmark, Journal of Visual Communication and Image Representation 34 (2016) 187–203.

[63] K. Li, G. Wan, G. Cheng, L. Meng, J. Han, Object detection in optical remote sensing images: A survey and a new benchmark, ISPRS journal of photogrammetry and remote sensing 159 (2020) 296–307.

[64] C. Chen, W. Gong, Y. Chen, W. Li, Object detection in remote sensing images based on a scene-contextual feature pyramid network, Remote Sensing 11 (3) (2019) 339.

[65] J. Rabbi, N. Ray, M. Schubert, S. Chowdhury, D. Chao, Small-object detection in remote sensing images with end-to-end edge-enhanced gan and object detector network, Remote Sensing 12 (9) (2020) 1432.

[66] T. N. Mundhenk, G. Konjevod, W. A. Sakla, K. Boakye, A large contextual dataset for classification, detection and counting of cars with deep learning, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14, Springer, 2016, pp. 785–800.

[67] J. Rabbi, S. Chowdhury, D. Chao, Oil and gas tank dataset, Mendeley Data, V3, available online: `https://data.mendeley.com/datasets/bkxj8z84m9/3` (accessed on 30 April 2020) (2020). `doi:10.17632/bkxj8z84m9.3`.

[68] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 10012–10022.

[69] Y. Zhang, Y. Yuan, Y. Feng, X. Lu, Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection, IEEE Transactions on Geoscience and Remote Sensing 57 (8) (2019) 5535–5548.

[70] Z. He, L. Huang, W. Zeng, X. Zhang, Y. Jiang, Q. Zou, Elongated small object detection from remote sensing images using hierarchical scale-sensitive networks, Remote Sensing 13 (16) (2021) 3182.

[71] S. Liu, J. Tang, Modified deep reinforcement learning with efficient convolution feature for small target detection in vhr remote sensing imagery, ISPRS International Journal of Geo-Information 10 (3) (2021) 170.

[72] J. Wang, F. Shao, X. He, G. Lu, A novel method of small object detection in uav remote sensing images based on feature alignment of candidate regions, Drones 6 (10) (2022) 292.

[73] M. Wang, Q. Li, Y. Gu, J. Pan, Highly efficient anchor-free oriented small object detection for remote sensing images via periodic pseudo-domain, Remote Sensing 15 (15) (2023) 3854.

[74] H. Zhu, X. Chen, W. Dai, K. Fu, Q. Ye, J. Jiao, Orientation robust object detection in aerial images using deep convolutional neural network, in: 2015 IEEE international conference on image processing (ICIP), IEEE, 2015, pp. 3735–3739.

[75] Z. Liu, L. Yuan, L. Weng, Y. Yang, A high resolution optical satellite image dataset for ship recognition and some new baselines, in: International conference on pattern recognition applications and methods, Vol. 2, SciTePress, 2017, pp. 324–331.

[76] K. Liu, G. Mattyus, Fast multiclass vehicle detection on aerial images, IEEE Geoscience and Remote Sensing Letters 12 (9) (2015) 1938–1942.

[77] H. Zhu, Y. Lv, J. Meng, Y. Liu, L. Hu, J. Yao, X. Lu, Vehicle detection in multisource remote sensing images based on edge-preserving super-resolution reconstruction, Remote Sensing 15 (17) (2023) 4281.

[78] G. Jocher, K. Nishimura, T. Minerva, R. Vilariño, Yolov5 https://github. com/ultralytics/yolov5, Accessed March 7 (2020) 2021.

[79] Z. Tian, C. Shen, H. Chen, T. He, Fcos: Fully convolutional one-stage object detection, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 9627–9636.

[80] P. Zhu, L. Wen, X. Bian, H. Ling, Q. Hu, Vision meets drones: A challenge, arXiv preprint arXiv:1804.07437 (2018).

[81] L. Meng, L. Zhou, Y. Liu, Sodcnn: A convolutional neural network model for small object detection in drone-captured images, Drones 7 (10) (2023) 615.

[82] C.-Y. Wang, A. Bochkovskiy, H.-Y. M. Liao, Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2023, pp. 7464–7475.

[83] M. Maktab Dar Oghaz, M. Razaak, P. Remagnino, Enhanced single shot small object detector for aerial imagery using super-resolution, feature fusion and deconvolution, Sensors 22 (12) (2022) 4339.

[84] C. Chen, J. Zhong, Y. Tan, Multiple-oriented and small object detection with convolutional neural networks for aerial image, Remote Sensing 11 (18) (2019) 2176.

[85] F. Tanner, B. Colder, C. Pullen, D. Heagy, M. Eppolito, V. Carlan, C. Oertel, P. Sallee, Overhead imagery research data set—an annotated data library & tools to aid in the development of computer vision algorithms, in: 2009 IEEE Applied Imagery Pattern Recognition Workshop (AIPR 2009), IEEE, 2009, pp. 1–8.

[86] X. Yu, Y. Gong, N. Jiang, Q. Ye, Z. Han, Scale match for tiny person detection, in: Proceedings of the IEEE/CVF winter conference on applications of computer vision, 2020, pp. 1257–1265.

[87] W. Lu, C. Lan, C. Niu, W. Liu, L. Lyu, Q. Shi, S. Wang, A cnn-transformer hybrid model based on cswin transformer for uav image object detection, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 16 (2023) 1211–1231.

[88] J. Yan, J. Zhao, Y. Cai, S. Wang, X. Qiu, X. Yao, Y. Tian, Y. Zhu, W. Cao, X. Zhang, Improving multi-scale detection layers in the deep learning network for wheat spike detection based on interpretive analysis, Plant Methods 19 (1) (2023) 46.

[89] E. David, S. Madec, P. Sadeghi-Tehran, H. Aasen, B. Zheng, S. Liu, N. Kirchgessner, G. Ishikawa, K. Nagasawa, M. A. Badhon, et al., Global wheat head detection (gwhd) dataset: A large and diverse dataset of high-resolution rgb-labelled images to develop and benchmark wheat head detection methods, Plant Phenomics (2020).

[90] E. David, M. Serouart, D. Smith, S. Madec, K. Velumani, S. Liu, X. Wang, F. Pinto, S. Shafiee, I. S. Tahir, et al., Global wheat head detection 2021: An improved dataset for benchmarking wheat head detection methods, Plant Phenomics (2021).

[91] H. Yu, Z. Li, W. Li, W. Guo, D. Li, L. Wang, M. Wu, Y. Wang, A tiny object detection approach for maize cleaning operations, Foods 12 (15) (2023) 2885.

[92] L. Justen, D. Carlsmith, S. M. Paskewitz, L. C. Bartholomay, G. M. Bron, Identification of public submitted tick images: A neural network approach, Plos one 16 (12) (2021) e0260622.

[93] X. Sun, G. Li, S. Xu, Fsd: feature skyscraper detector for stem end and blossom end of navel orange, Machine Vision and Applications 32 (1) (2021) 11.

[94] W. Xie, H. Qin, Y. Li, Z. Wang, J. Lei, A novel effectively optimized one-stage network for object detection in remote sensing imagery, Remote Sensing 11 (11) (2019). doi:10.3390/rs11111376.
URL https://www.mdpi.com/2072-4292/11/11/1376

[95] H. Song, H. Liang, H. Li, Z. Dai, X. Yun, Vision-based vehicle detection and counting system using deep learning in highway scenes, European Transport Research Review 11 (1) (2019) 1–16. doi:10.1186/s12544-019-0390-4.

[96] E. C. Tetila, B. B. Machado, G. V. Menezes, N. A. de Souza Belete, G. Astolfi, H. Pistori, A deep-learning approach for automatic counting of soybean insect pests, IEEE Geoscience and Remote Sensing Letters (2019).

[97] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk, Slic superpixels compared to state-of-the-art superpixel methods, IEEE transactions on pattern analysis and machine intelligence 34 (11) (2012) 2274–2282.

[98] S.-H. Kang, J.-S. Park, Aligned matching: improving small object detection in ssd, Sensors 23 (5) (2023) 2589.

[99] Y. Ge, D. Jiang, L. Sun, Wood veneer defect detection based on multiscale detr with position encoder net, Sensors 23 (10) (2023) 4837.

[100] B. Huo, C. Li, J. Zhang, Y. Xue, Z. Lin, Saff-ssd: Self-attention combined feature fusion-based ssd for small object detection in remote sensing, Remote Sensing 15 (12) (2023) 3027.

[101] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, http://www.deeplearningbook.org.

[102] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017.

[103] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, H.-Y. Shum, Dino: Detr with improved denoising anchor boxes for end-to-end object detection (2022). arXiv:2203.03605.

[104] S. Liu, F. Li, H. Zhang, X. Yang, X. Qi, H. Su, J. Zhu, L. Zhang, DAB-DETR: Dynamic anchor boxes are better queries for DETR, in: International Conference on Learning Representations, 2022.
URL https://openreview.net/forum?id=oMI9PjOb9Jl

[105] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, P. Luo, Segformer: Simple and efficient design for semantic segmentation with transformers, arXiv preprint arXiv:2105.15203 (2021).

[106] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, arXiv preprint arXiv:1706.05587 (2017).

[107] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: ECCV, 2018.

Capítulo 5

Conclusões

Neste capítulo, são apresentadas as principais conclusões do trabalho, destacando as contribuições alcançadas, bem como as limitações observadas durante a condução dos experimentos e possíveis caminhos para pesquisas futuras. Os resultados obtidos demonstram o potencial das abordagens propostas para a detecção de objetos pequenos em imagens de alta resolução, usando como base o contexto agrícola e entomológico, além de evidenciarem desafios ainda presentes neste cenário.

Além disso, os trabalhos apresentados nos Capítulos 2 e 3 contribuíram para o amadurecimento metodológico da pesquisa, mostrando limitações nos cenários em que os objetos estão distantes da câmera e suas dimensões são menores em relação ao tamanho da imagem. A seguir, são detalhadas as contribuições científicas e técnicas proporcionadas por este estudo (Seção 5.1), suas limitações e restrições (Seção 5.2), bem como sugestões de melhorias e oportunidades para investigações futuras (Seção 5.3).

5.1 Contribuições

As contribuições deste trabalho abrangem desde a preparação de dados até o desenvolvimento e avaliação de técnicas para detecção e segmentação de objetos pequenos em imagens de alta resolução, usando como base o domínio agrícola e entomológico. As principais contribuições podem ser destacadas a seguir:

- **Adaptação de conjunto de dados:** uma adaptação foi realizada em um conjunto de dados originalmente desenvolvido para classificação de *su-*

perpixels, convertendo cada *superpixel* com inseto em caixa delimitadora e ajustando suas coordenadas para representar apenas a área do inseto. Além disso, novos insetos foram anotados e diferentes classes de *Euschistus* foram unificadas, resultando em 10537 insetos de 4 classes diferentes anotados em 1000 imagens. A disponibilização destas anotações contribui para pesquisas futuras no contexto de insetos em ambientes agrícolas;

- **Proposta de técnicas para pré-processamento:** o uso de cada recorte individualmente, ao invés da imagem inteira, como entrada para modelos de detecção e segmentação, contorna o problema do desaparecimento de objetos ao longo das camadas dos modelos, pois a relação entre o tamanho do objeto e o tamanho do recorte é maior, quando comparada ao tamanho da imagem original. Adicionalmente, a conversão de caixas delimitadoras em máscaras permite que modelos de segmentação sejam utilizados com a técnica de recorte proposta;

- **Proposta de técnicas para pós-processamento:** estratégias de pós-processamento foram implementadas para unir detecções recortadas no pré-processamento. Entre elas, uma adaptação do algoritmo de supressão de não-máximos, para considerar a área dos objetos ao invés de sua pontuação (*score*), e uma métrica alternativa à IoU, baseada na distância entre centros de caixas delimitadoras. Essas soluções buscam reduzir o impacto que métodos tradicionais causam quando aplicados a objetos pequenos;

- **Comparação de abordagens:** abordagens de detecção e segmentação foram avaliadas e comparadas quanto à identificação de objetos pequenos, considerando tanto modelos originais quanto as técnicas de pré e pós-processamento propostas neste trabalho, destacando vantagens, limitações e desafios específicos relacionados à detecção de insetos em imagens de alta resolução.

5.2 Limitações

Apesar dos resultados apresentados, este trabalho apresenta algumas limitações:

- **Aumento do tempo de processamento:** as estratégias de divisão em recortes e reconstrução das imagens, embora preserve objetos pequenos, aumentam o tempo de treinamento e inferência dos modelos, limitando a aplicação das técnicas em cenários que exigem respostas em tempo real;

- **Ausência de testes com caixas delimitadoras inclinadas:** as técnicas propostas foram avaliadas apenas com caixas delimitadoras convencionais. Portanto, adaptações podem ser necessárias para objetos que requerem a utilização de caixas delimitadoras inclinadas;

- **Generalização para outros domínios:** as abordagens foram desenvolvidas e avaliadas no contexto de insetos em folhas de soja, sendo necessárias maiores investigações para outros domínios, como imagens urbanas, médicas ou marítimas;

- **Avaliação restrita a determinadas arquiteturas:** as comparações e análises foram realizadas com um conjunto limitado de modelos de detecção e segmentação, e com um único conjunto de dados adaptado. Novas avaliações, com arquiteturas ou bases de dados distintas podem apresentar comportamentos diferentes;

- **Possíveis impactos nos resultados com objetos fragmentados:** embora as estratégias de pós-processamento busquem mitigar problemas de fragmentação de objetos em recortes sobrepostos, casos difícieis podem ocorrer, como objetos com classes diferentes muito próximos ou parcialmente ocultos, além de outras dificuldades conhecidas ao lidar com objetos pequenos.

5.3 Melhorias futuras

Diversas direções podem ser exploradas para aprimorar e expandir as contribuições deste trabalho:

- **Otimização de desempenho:** investigar estratégias para reduzir o tempo de processamento, como classificar recortes que possuem objetos e usar apenas estes para treinamento e validação;

- **Suporte a caixas delimitadoras inclinadas:** adaptar as técnicas propostas para lidar com caixas delimitadoras inclinadas, avaliando o desempenho em cenários em que a orientação dos objetos não seja fixa;

- **Avaliação em novos domínios e conjuntos de dados:** aplicar as estratégias em diferentes contextos, como imagens urbanas, médicas ou marítimas, com o objetivo de identificar adaptações necessárias;

- **Exploração de formas alternativas de visualização dos resultados:** usar representações gráficas para facilitar a interpretação e comparação do desempenho dos modelos, tais como curvas *Precision-Recall* (PR) e

Area Under the Curve (AUC), assim como matrizes de confusão, que permitem avaliar de forma mais clara os acertos e erros de classificação em cada classe;

- **Investigação de convoluções deformáveis:** explorar convoluções deformáveis como alternativa ou complemento a métodos baseados em *Transformers*, avaliando sua capacidade em tarefas de detecção de pequenos objetos;

- **Possíveis aplicações comerciais e científicas com visão computacional e detecção de objetos:** as técnicas propostas apresentam potencial para diversas aplicações, como monitoramento de árvores e vegetação com imagens obtidas por drones, inspeção e manutenção preventiva de bueiros e poços de visita em áreas urbanas e detecção precoce de doenças e insetos em plantações para suporte a práticas de manejo agrícola.

Referências Bibliográficas

[1] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017. Citado na página 3.

[2] C. Cao, B. Wang, W. Zhang, X. Zeng, X. Yan, Z. Feng, Y. Liu, and Z. Wu. An improved faster r-cnn for small object detection. *Ieee Access*, 7:106838–106846, 2019. Citado na página 2.

[3] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. Citado na página 1.

[4] C. Chen, W. Gong, Y. Chen, and W. Li. Object detection in remote sensing images based on a scene-contextual feature pyramid network. *Remote Sensing*, 11(3):339, 2019. Citado na página 5.

[5] C. Chen, J. Zhong, and Y. Tan. Multiple-oriented and small object detection with convolutional neural networks for aerial image. *Remote Sensing*, 11(18):2176, 2019. Citado na página 7.

[6] G. Cheng, J. Han, P. Zhou, and L. Guo. Multi-class geospatial object detection and geographic image classification based on collection of part detectors. *ISPRS Journal of Photogrammetry and Remote Sensing*, 98:119–132, 2014. Citado na página 4.

[7] G. Cheng, X. Yuan, X. Yao, K. Yan, Q. Zeng, X. Xie, and J. Han. Towards large-scale small object detection: Survey and benchmarks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. Citado na página 3.

[8] H. Cui and Z. Wei. Multi-scale receptive field detection network. *IEEE Access*, 7:138825–138832, 2019. Citado na página 2.

[9] M. Cui, G. Gong, G. Chen, H. Wang, M. Jin, W. Mao, and H. Lu. Lc-yolo: A lightweight model with efficient utilization of limited detail features for small object detection. *Applied Sciences*, 13(5):3174, 2023. Citado nas páginas 2 e 6.

[10] E. David, S. Madec, P. Sadeghi-Tehran, H. Aasen, B. Zheng, S. Liu, N. Kirchgessner, G. Ishikawa, K. Nagasawa, M. A. Badhon, et al. Global wheat head detection (gwhd) dataset: A large and diverse dataset of high-resolution rgb-labelled images to develop and benchmark wheat head detection methods. *Plant Phenomics*, 2020. Citado na página 8.

[11] E. David, M. Serouart, D. Smith, S. Madec, K. Velumani, S. Liu, X. Wang, F. Pinto, S. Shafiee, I. S. Tahir, et al. Global wheat head detection 2021: An improved dataset for benchmarking wheat head detection methods. *Plant Phenomics*, 2021. Citado na página 8.

[12] L. Ding, Z. Q. Rao, B. Ding, and S. J. Li. Research on defect detection method of railway transmission line insulators based on gc-yolo. *IEEE Access*, 2023. Citado nas páginas 2 e 6.

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. Citado na página 1.

[14] Y. Feng, L. Wang, and M. Zhang. A multi-scale target detection method for optical remote sensing images. *Multimedia Tools and Applications*, 78(7):8751–8766, 2019. Citado nas páginas 2 e 10.

[15] A.-J. Gallego, A. Pertusa, and P. Gil. Automatic ship classification from optical aerial images with convolutional neural networks. *Remote Sensing*, 10(4):511, 2018. Citado na página 4.

[16] Y. Gan, S. You, Z. Luo, K. Liu, T. Zhang, and L. Du. Object detection in remote sensing images with mask r-cnn. In *Journal of Physics: Conference Series*, volume 1673, page 012040. IOP Publishing, 2020. Citado nas páginas 2 e 4.

[17] S. Gao, C. Liu, H. Zhang, Z. Zhou, and J. Qiu. Multiscale attention-based detection of tiny targets in aerial beach images. *Frontiers in Marine Science*, 9:1073615, 2022. Citado nas páginas 2 e 8.

[18] T. Gao, M. Wushouer, and G. Tuerhong. Dms-yolov5: A decoupled multi-scale yolov5 method for small object detection. *Applied Sciences*, 13(10):6124, 2023. Citado nas páginas 1 e 2.

[19] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. Citado na página 4.

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. Citado nas páginas 4 e 13.

[21] Z. He, L. Huang, W. Zeng, X. Zhang, Y. Jiang, and Q. Zou. Elongated small object detection from remote sensing images using hierarchical scale-sensitive networks. *Remote Sensing*, 13(16):3182, 2021. Citado na página 5.

[22] N. Jia, Z. Wei, and B. Li. Attention-enhanced lightweight one-stage detection algorithm for small objects. *Electronics*, 12(7):1607, 2023. Citado na página 2.

[23] D. Jiang, B. Sun, S. Su, Z. Zuo, P. Wu, and X. Tan. Fassd: A feature fusion and spatial attention-based single shot detector for small object detection. Citado na página 1.

[24] X. Jiang, A. Hadid, Y. Pang, E. Granger, and X. Feng. *Deep Learning in object detection and recognition*. Springer, 2019. Citado na página 1.

[25] Z. Jiang, Y. Wang, X. Zhou, L. Chen, Y. Chang, D. Song, and H. Shi. Small-scale ship detection for sar remote sensing images based on coordinate-aware mixed attention and spatial semantic joint context. *Smart Cities*, 6(3):1612–1629, 2023. Citado na página 2.

[26] G. Jocher, K. Nishimura, T. Minerva, and R. Vilariño. Yolov5 https://github. com/ultralytics/yolov5. *Accessed March*, 7:2021, 2020. Citado na página 6.

[27] L. Justen, D. Carlsmith, S. M. Paskewitz, L. C. Bartholomay, and G. M. Bron. Identification of public submitted tick images: A neural network approach. *Plos one*, 16(12):e0260622, 2021. Citado na página 9.

[28] H. Lai, L. Chen, W. Liu, Z. Yan, and S. Ye. Stc-yolo: small object detection network for traffic signs in complex environments. *Sensors*, 23(11):5307, 2023. Citado na página 2.

[29] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS journal*

of photogrammetry and remote sensing, 159:296–307, 2020. Citado na página 3.

[30] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS journal of photogrammetry and remote sensing*, 159:296–307, 2020. Citado na página 4.

[31] L. Li, S. Zhang, and J. Wu. Efficient object detection framework and hardware architecture for remote sensing images. *Remote Sensing*, 11(20):2376, 2019. Citado na página 3.

[32] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, 2016. Citado na página 25.

[33] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. Citado na página 13.

[34] Z. Lin, Q. Wu, S. Fu, S. Wang, Z. Zhang, and Y. Kong. Dual-nms: A method for autonomously removing false detection boxes from aerial image object detection results. *Sensors*, 19(21):4691, oct 2019. Citado nas páginas 1 e 10.

[35] J. Liu, S. Yang, L. Tian, W. Guo, B. Zhou, J. Jia, and H. Ling. Multi-component fusion network for small object detection in remote sensing images. *IEEE Access*, 7:128339–128352, 2019. Citado nas páginas 2 e 5.

[36] K. Liu and G. Mattyus. Fast multiclass vehicle detection on aerial images. *IEEE Geoscience and Remote Sensing Letters*, 12(9):1938–1942, 2015. Citado na página 5.

[37] M. Liu, X. Wang, A. Zhou, X. Fu, Y. Ma, and C. Piao. Uav-yolo: Small object detection on unmanned aerial vehicle perspective. *Sensors*, 20(8):2238, 2020. Citado nas páginas 2 e 6.

[38] S. Liu and J. Tang. Modified deep reinforcement learning with efficient convolution feature for small target detection in vhr remote sensing imagery. *ISPRS International Journal of Geo-Information*, 10(3):170, 2021. Citado na página 5.

[39] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016:*

14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, pages 21–37. Springer, 2016. Citado nas páginas 1 e 3.

[40] Z. Liu, X. Gao, Y. Wan, J. Wang, and H. Lyu. An improved yolov5 method for small object detection in uav capture scenes. *IEEE Access*, 11:14365–14374, 2023. Citado nas páginas 2 e 6.

[41] Z. Liu, D. Li, S. S. Ge, and F. Tian. Small traffic sign detection from large image. *Applied Intelligence*, 50(1):1–13, jun 2019. Citado na página 1.

[42] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. Citado nas páginas 5 e 8.

[43] Z. Liu, L. Yuan, L. Weng, and Y. Yang. A high resolution optical satellite image dataset for ship recognition and some new baselines. In *International conference on pattern recognition applications and methods*, volume 2, pages 324–331. SciTePress, 2017. Citado na página 5.

[44] W. Lu, C. Lan, C. Niu, W. Liu, L. Lyu, Q. Shi, and S. Wang. A cnn-transformer hybrid model based on cswin transformer for uav image object detection. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 16:1211–1231, 2023. Citado na página 8.

[45] G. Ma et al. A small target detection method based on the improved fcn model. *Advances in Multimedia*, 2022, 2022. Citado na página 2.

[46] M. Maktab Dar Oghaz, M. Razaak, and P. Remagnino. Enhanced single shot small object detector for aerial imagery using super-resolution, feature fusion and deconvolution. *Sensors*, 22(12):4339, 2022. Citado na página 7.

[47] L. Meng, L. Zhou, and Y. Liu. Sodcnn: A convolutional neural network model for small object detection in drone-captured images. *Drones*, 7(10):615, 2023. Citado na página 7.

[48] V. Moskalenko, A. Moskalenko, A. Korobov, and M. Zaretsky. A model and training algorithm of small-sized object detection system for a compact aerial drone. , , , pages 110–121, 2019. Citado nas páginas 2 e 7.

[49] J. Mu, S. Li, Z. Liu, and Y. Zhou. Integration of gradient guidance and edge enhancement into super-resolution for small object detection in aerial images. *IET Image Processing*, 15(13):3037–3052, 2021. Citado na página 2.

[50] T. N. Mundhenk, G. Konjevod, W. A. Sakla, and K. Boakye. A large contextual dataset for classification, detection and counting of cars with deep learning. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 785–800. Springer, 2016. Citado na página 5.

[51] M. Nikouei, B. Baroutian, S. Nabavi, F. Taraghi, A. Aghaei, A. Sajedi, and M. E. Moghaddam. Small object detection: A comprehensive survey on challenges, techniques and real-world applications. *arXiv preprint arXiv:2503.20516*, 2025. Citado na página 3.

[52] J. Pang, C. Li, J. Shi, Z. Xu, and H. Feng. R2-cnn: Fast tiny object detection in large-scale remote sensing images. arxiv 2019. *arXiv preprint arXiv:1902.06042*, 3, 2019. Citado nas páginas 2, 3, e 4.

[53] J. Rabbi, S. Chowdhury, and D. Chao. Oil and gas tank dataset. Mendeley Data, V3, 2020. Available online: https://data.mendeley.com/datasets/bkxj8z84m9/3 (accessed on 30 April 2020). Citado na página 5.

[54] J. Rabbi, N. Ray, M. Schubert, S. Chowdhury, and D. Chao. Small-object detection in remote sensing images with end-to-end edge-enhanced gan and object detector network. *Remote Sensing*, 12(9):1432, 2020. Citado na página 5.

[55] Q. Ran, Q. Wang, B. Zhao, Y. Wu, S. Pu, and Z. Li. Lightweight oriented object detection using multiscale context and enhanced channel attention in remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:5786–5795, 2021. Citado nas páginas 2 e 4.

[56] S. Razakarivony and F. Jurie. Vehicle detection in aerial imagery: A small target detection benchmark. *Journal of Visual Communication and Image Representation*, 34:187–203, 2016. Citado na página 4.

[57] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. Citado na página 1.

[58] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. Citado nas páginas 3 e 13.

[59] Y. Rehman, H. Amanullah, M. A. Shirazi, and M. Y. Kim. Small traffic sign detection in big images: Searching needle in a hay. *IEEE Access*, 10:18667–18680, 2022. Citado na página 2.

[60] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015. Citado nas páginas 1, 3, e 13.

[61] F. Saeed, M. J. Ahmed, M. J. Gul, K. J. Hong, A. Paul, and M. S. Kavitha. A robust approach for industrial small-object detection using an improved faster regional convolutional neural network. *Scientific reports*, 11(1):23390, 2021. Citado na página 2.

[62] A. Santos, J. Marcato Junior, J. de Andrade Silva, R. Pereira, D. Matos, G. Menezes, L. Higa, A. Eltner, A. P. Ramos, L. Osco, and W. Gonçalves. Storm-drain and manhole detection using the retinanet method. *Sensors*, 20(16), 2020. Citado na página 25.

[63] A. A. d. Santos, J. Marcato Junior, M. S. Araújo, D. R. Di Martini, E. C. Tetila, H. L. Siqueira, C. Aoki, A. Eltner, E. T. Matsubara, H. Pistori, R. Q. Feitosa, V. Liesenberg, and W. N. Gonçalves. Assessment of cnn-based methods for individual tree detection on images captured by rgb cameras attached to uavs. *Sensors*, 19(16), 2019. Citado na página 13.

[64] H. Song, H. Liang, H. Li, Z. Dai, and X. Yun. Vision-based vehicle detection and counting system using deep learning in highway scenes. *European Transport Research Review*, 11(1):1–16, dec 2019. Citado na página 10.

[65] X. Sun, G. Li, and S. Xu. Fsd: feature skyscraper detector for stem end and blossom end of navel orange. *Machine Vision and Applications*, 32(1):11, 2021. Citado na página 9.

[66] H. Takimoto, Y. Sato, A. J. Nagano, K. K. Shimizu, and A. Kanagawa. Using a two-stage convolutional neural network to rapidly identify tiny herbivorous beetles in the field. *Ecological Informatics*, 66:101466, 2021. Citado nas páginas 2 e 8.

[67] F. Tanner, B. Colder, C. Pullen, D. Heagy, M. Eppolito, V. Carlan, C. Oertel, and P. Sallee. Overhead imagery research data set—an annotated data library & tools to aid in the development of computer vision algorithms. In *2009 IEEE Applied Imagery Pattern Recognition Workshop (AIPR 2009)*, pages 1–8. IEEE, 2009. Citado na página 7.

[68] Z. Tian, C. Shen, H. Chen, and T. He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. Citado na página 6.

[69] B. Wang and Y. Gu. An improved fbpn-based detection network for vehicles in aerial images. *Sensors*, 20(17):4709, aug 2020. Citado nas páginas 1 e 10.

[70] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7464–7475, 2023. Citado na página 7.

[71] J. Wang, F. Shao, X. He, and G. Lu. A novel method of small object detection in uav remote sensing images based on feature alignment of candidate regions. *Drones*, 6(10):292, 2022. Citado na página 5.

[72] M. Wang, Q. Li, Y. Gu, and J. Pan. Highly efficient anchor-free oriented small object detection for remote sensing images via periodic pseudo-domain. *Remote Sensing*, 15(15):3854, 2023. Citado na página 5.

[73] R. J. Wang, X. Li, and C. X. Ling. Pelee: A real-time object detection system on mobile devices. *Advances in neural information processing systems*, 31, 2018. Citado na página 4.

[74] Z. Wang, D. Liu, Y. Lei, X. Niu, S. Wang, and L. Shi. Small target detection based on bird's visual information processing mechanism. *Multimedia Tools and Applications*, 79:22083–22105, may 2020. Citado nas páginas 1 e 2.

[75] Z. Wu. Detect small object based on fcos and adaptive feature fusion. In *Journal of Physics: Conference Series*, volume 2580, page 012005. IOP Publishing, 2023. Citado na página 2.

[76] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3974–3983, 2018. Citado na página 3.

[77] Q. Xiang, X. Huang, Z. Huang, X. Chen, J. Cheng, and X. Tang. Yolo-pest: an insect pest object detection algorithm via cac3 module. *Sensors*, 23(6):3221, 2023. Citado nas páginas 2 e 8.

[78] Z. Xiao, Q. Liu, G. Tang, and X. Zhai. Elliptic fourier transformation-based histograms of oriented gradients for rotationally invariant object

detection in remote-sensing images. *International Journal of Remote Sensing*, 36(2):618–644, 2015. Citado na página 4.

[79] W. Xie, H. Qin, Y. Li, Z. Wang, and J. Lei. A novel effectively optimized one-stage network for object detection in remote sensing imagery. *Remote Sensing*, 11(11), 2019. Citado na página 10.

[80] S. Xu, Q. Feng, J. Fei, G. Zhao, X. Liu, H. Li, C. Lu, and Q. Yang. A locating approach for small-sized components of railway catenary based on improved yolo with asymmetrically effective decoupled head. *IEEE Access*, 11:34870–34879, 2023. Citado na página 2.

[81] X. Xu, Z. Feng, C. Cao, M. Li, J. Wu, Z. Wu, Y. Shang, and S. Ye. An improved swin transformer-based model for remote sensing object detection and instance segmentation. *Remote Sensing*, 13(23):4779, 2021. Citado nas páginas 2 e 5.

[82] J. Yan, H. Wang, M. Yan, W. Diao, X. Sun, and H. Li. Iou-adaptive deformable r-cnn: Make full use of iou for multi-class object detection in remote sensing imagery. *Remote Sensing*, 11(3), 2019. Citado na página 2.

[83] J. Yan, J. Zhao, Y. Cai, S. Wang, X. Qiu, X. Yao, Y. Tian, Y. Zhu, W. Cao, and X. Zhang. Improving multi-scale detection layers in the deep learning network for wheat spike detection based on interpretive analysis. *Plant Methods*, 19(1):46, 2023. Citado na página 8.

[84] X. Ye, F. Xiong, J. Lu, J. Zhou, and Y. Qian. F3-net: Feature fusion and filtration network for object detection in optical remote sensing images. *Remote Sensing*, 12(24), 2020. Citado nas páginas 1 e 2.

[85] H. Yu, Z. Li, W. Li, W. Guo, D. Li, L. Wang, M. Wu, and Y. Wang. A tiny object detection approach for maize cleaning operations. *Foods*, 12(15):2885, 2023. Citado na página 9.

[86] X. Yu, Y. Gong, N. Jiang, Q. Ye, and Z. Han. Scale match for tiny person detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1257–1265, 2020. Citado na página 8.

[87] F. Zhang and X. Hou. Multi-site and multi-scale unbalanced ship detection based on centernet. *Electronics*, 11(11):1713, 2022. Citado na página 2.

[88] H. Zhang, L. Qin, J. Li, Y. Guo, Y. Zhou, J. Zhang, and Z. Xu. Real-time detection method for small traffic signs based on yolov3. *Ieee Access*, 8:64145–64156, 2020. Citado na página 2.

[89] J. Zhang, J. Lei, W. Xie, Y. Li, G. Yang, and X. Jia. Guided hybrid quantization for object detection in remote sensing imagery via one-to-one self-teaching. *IEEE Transactions on Geoscience and Remote Sensing*, 2023. Citado nas páginas 2 e 4.

[90] X. Zhang, Q. Liu, H. Chang, and H. Sun. High-resolution network with transformer embedding parallel detection for small object detection in optical remote sensing images. *Remote Sensing*, 15(18):4497, 2023. Citado nas páginas 2 e 4.

[91] Y. Zhang, Y. Bai, M. Ding, and B. Ghanem. Multi-task generative adversarial network for detecting small objects in the wild. *International Journal of Computer Vision*, 128(6):1810–1828, feb 2020. Citado nas páginas 1 e 2.

[92] Y. Zhang, Y. Bai, M. Ding, and B. Ghanem. Multi-task generative adversarial network for detecting small objects in the wild. *International Journal of Computer Vision*, 128(6):1810–1828, 2020. Citado na página 2.

[93] Y. Zhang and T. Shen. Small object detection with multiple receptive fields. In *IOP Conference Series: Earth and Environmental Science*, volume 440, page 032093. IOP Publishing, 2020. Citado nas páginas 1 e 2.

[94] Y. Zhang, Y. Yuan, Y. Feng, and X. Lu. Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 57(8):5535–5548, 2019. Citado na página 5.

[95] K. Zhao and X. Ren. Small aircraft detection in remote sensing images based on yolov3. In *IOP Conference Series: Materials Science and Engineering*, volume 533, page 012056. IOP Publishing, 2019. Citado na página 3.

[96] H. Zhou, A. Ma, Y. Niu, and Z. Ma. Small-object detection for uav-based images using a distance metric method. *Drones*, 6(10):308, 2022. Citado nas páginas 2 e 7.

[97] H. Zhu, X. Chen, W. Dai, K. Fu, Q. Ye, and J. Jiao. Orientation robust object detection in aerial images using deep convolutional neural network. In *2015 IEEE international conference on image processing (ICIP)*, pages 3735–3739. IEEE, 2015. Citado na página 5.

[98] H. Zhu, Y. Lv, J. Meng, Y. Liu, L. Hu, J. Yao, and X. Lu. Vehicle detection in multisource remote sensing images based on edge-preserving super-

resolution reconstruction. *Remote Sensing*, 15(17):4281, 2023. Citado na página 6.

[99] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu. Vision meets drones: A challenge. *arXiv preprint arXiv:1804.07437*, 2018. Citado na página 6.