Detecção e rastreamento de múltiplos objetos utilizando redes profundas no contexto de mapeamento de formigueiros em plantação de Eucaliptos

Gian Lucas da Silva Ramalho

Gian Lucas da Silva Ramalho

Orientador: *Profº Drº Jonathan de Andrade Silva*
Coorientador: *Profº Drº José Marcato Junior*
Coorientador: *Profº Drº Lúcio André de Castro Jorge*

Dissertação apresentada ao Programa de Pós-Graduação em Computação Aplicada da Faculdade de Computação/FACOM pela Universidade Federal do Mato Grosso do Sul/UFMS, como parte dos requisitos para a obtenção do título de Mestre Computação Aplicada.

UFMS - Campo Grande
setembro/2022

Abstract

The forestry sector enables economic and environmental development, offering employment and income to the population and helping to reduce climate change. According to IBGE, in 2019, the area of forests cultivated throughout the national territory reached a total of 9.98 million hectares. Eucalyptus cultivation represents approximately 76%, equivalent to 7.61 million hectares. In the forest plantations present in Brazil, one of the main pests that intensely affect production, are leaf-cutting ants. These insects consume a lot of vegetation, attacking different plant species and causing defoliation to death. of the plant, regardless of its size, from seedlings to trees. To fight ants, chemical products are used, along with plantation monitoring. You can apply detection and tracking of objects in images of the plantations, to assist in the monitoring of the plantation and the anthills. The detection and tracking of objects in this study fit the context of tracking multiple objects, Multiple Object Tracking (MOT). The MOT task refers to locating multiple objects, identifying them, and calculating their trajectories. individual images in a sequence of images. In this study, three object detectors, Faster R-CNN, RetinaNet, and VFNet, along with the Tracktor tracking methods, Byte Tracker Deep Sort in addition to the proposal of a method based on the SORT Method, for tracking anthills. Evaluations of object detection and tracking methods were carried out, and the best tracking results obtained were using the RetinaNet detector which achieved 0.817 of Average Precision (AP), 53,004 of Higher Order Tracking Accuracy (HOTA) with method Byte Tracker, 47,120 HOTA with the Proposed Method and 43,426 HOTA with the Deep Sort. Although the Byte Tracker indicates a superior HOTA result, the Method Proposed excels in counting objects, outperforming other methods tracking.

Resumo

O setor da silvicultura possibilita o desenvolvimento econômico e ambiental, oferecendo emprego e renda para a população e auxiliando com a redução das mudanças climáticas. Segundo IBGE, no ano de 2019, a área de florestas cultivadas em todo o território nacional alcançara um total de 9,98 milhões de hectares. O cultivo de eucalipto representa aproximadamente 76%, equivalente a 7,61 milhões de hectares. Nas plantações florestais presentes no Brasil, uma das principais pragas e que afetam intensamente a produção, são as formigas cortadeiras. Esses insetos consomem muita vegetação, atacando diferentes as espécies de plantas e causando a desfolha até a morte da planta, independendo do tamanho dela, de mudas até árvores. Para o combater às formigas, são utilizados produtos químicos, juntamente com o monitoramento da plantação. É possível aplicar a detecção e o rastreamento de objetos em imagens das plantações, com o intuito de auxiliar no monitoramento da plantação e dos formigueiros. A detecção e o rastreamento dos objetos nesse estudo se encaixam no contexto do rastreamento de múltiplos objetos, *Multiple Object Tracking* (MOT). A tarefa do MOT refere-se na localização de múltiplos objetos, na sua identificação e no cálculo de suas trajetórias individuais, em uma sequência de imagens. Neste estudo foram avaliados três detectores de objetos, *Faster R-CNN*, *RetinaNet* e *VFNet*, juntamente com os métodos de rastreamento *Tracktor*, *Byte Tracker Deep Sort*, além da proposta de um método baseado no Método *SORT*, para rastreamento de formigueiros. As avaliações dos métodos de detecção e rastreamento de objetos foram realizadas, e o melhores resultados de rastreamento obtidos foram utilizando o detector *RetinaNet* que atingiu 0.817 de *Average Precision* (AP), 53.004 de *Higher Order Tracking Accuracy* (HOTA) com o método de rastreamento *Byte Tracker*, 47.120 de HOTA com o Método Proposto e 43.426 de HOTA com o *Deep Sort*. Apesar do *Byte Tracker* indicar resultado HOTA superior, o Método Proposto se destaca na contagem dos objetos, superando os outros métodos de rastreamento.

Sumário

Sumário vi
Lista de Figuras vii
Lista de Tabelas viii
Lista de Abreviaturas viii
Lista de Algoritmos 1

1 Introdução **2**
1.1 Hipóteses 4
1.2 Objetivos 4
1.3 Organização do Texto 4

2 Revisão de Literatura **6**
2.1 Visão Computacional 6
2.1.1 Detecção de Objetos 6
2.1.2 Rastreamento de Objetos 7
2.2 Redes Neurais 8
2.2.1 Redes Neurais Convolucionais 9
2.3 Trabalhos Relacionados 10
2.3.1 Detecção de Objetos 11
2.3.2 Rastreamento de Objetos 14

3 Metodologia **16**
3.1 Aquisição das imagens 16
3.2 Anotação dos objetos 17
3.3 Detecção dos objetos 18
3.3.1 *Faster R-CNN* 19
3.3.2 *RetinaNet* 19
3.3.3 *VFNet* 20
3.4 Rastreamento dos objetos 20
3.4.1 Método *Deep SORT* 21

3.4.2 Método *Tracktor* 22
3.4.3 Método *Byte Track* 22
3.4.4 Método Proposto 23
3.5 Métricas de avaliação 25
3.5.1 *Average Precision* (AP) 25
3.5.2 *Higher Order Tracking Accuracy* (HOTA) 26
3.5.3 Avaliação das curvas de aprendizagem 28

4 Resultados **29**
4.1 Detecção de objetos 29
4.2 Rastreamento de objetos 30

5 Conclusão **39**
5.1 Conclusão 39
5.2 Trabalhos Futuros 39

Referências **44**

Lista de Figuras

2.1 *Pipeline* da detecção de objetos. 9
2.2 Fluxo Rastreamento. 9
2.3 Estrutura RNA. 10
2.4 Estrutura RNC. 11

3.1 Fluxo da metodologia. 16
3.2 *Frame* do vídeo contendo formigueiros. 17
3.3 Anotação formigueiro com IDs. 18
3.4 Cálculo localização. 27

4.1 curva de aprendizagem *Faster R-CNN*. 30
4.2 curva de aprendizagem *RetinaNet*. 30
4.3 Curva de aprendizagem *VFNet*. 31
4.4 GPBK0197 Corte1 Faster R-CNN + Método Proposto X GT. 38
4.5 GPBK0197 Corte1 Faster R-CNN + *Tracktor* X GT. 38

Lista de Tabelas

3.1 Informações das bases de dados 19

4.1 Avaliação dos métodos de detecção 30
4.2 Resultados HOTA: Detector *Faster R-CNN* + *ResNet101* 32
4.3 Resultados DetA: Detector *Faster R-CNN* + *ResNet101* 32
4.4 Resultados AssA: Detector *Faster R-CNN* + *ResNet101* 33
4.5 Resultados LocA: Detector *Faster R-CNN* + *ResNet101* 33
4.6 Contagem de Dets: Detector *Faster R-CNN* + *ResNet101* 33
4.7 Contagem de IDs: Detector *Faster R-CNN* + *ResNet101* 33
4.8 Resultados HOTA: Detector *RetinaNet* + *ResNet101* 34
4.9 Resultados DetA: Detector *RetinaNet* + *ResNet101* 34
4.10 Resultados AssA: Detector *RetinaNet* + *ResNet101* 34
4.11 Resultados LocA: Detector *RetinaNet* + *ResNet101* 35
4.12 Contagem de Dets: Detector *RetinaNet* + *ResNet101* 35
4.13 Contagem de IDs: Detector *RetinaNet* + *ResNet101* 35
4.14 Resultados HOTA: Detector *VFNet* + *ResNet101* 36
4.15 Resultados DetA: Detector *VFNet* + *ResNet101* 36
4.16 Resultados AssA: Detector *VFNet* + *ResNet101* 36
4.17 Resultados LocA: Detector *VFNet* + *ResNet101* 36
4.18 Contagem de Dets: Detector *VFNet* + *ResNet101* 37
4.19 Contagem de IDs: Detector *VFNet* + *ResNet101* 37

Lista de Abreviaturas

IBGE Instituto Brasileiro de Geografia e Estatística

VBP Valor Bruto da Produção Agropecuária

RNAs Redes Neurais Artificiais

RNC Rede Neural Convolucional

CNN Convolutional Neural network

MAPE Erro Absoluto Médio Percentual

IoU Intersecção sobre a união

MOT Rastreamento de Múltiplos Objetos

DBT *Detection-Based Tracking*

DFT *Detection-Free Tracking*

RPN Rede de proposta de região

SGD Descida de gradiente estocástico

VFNet *VarifocalNet*

SORT *Simple online and realtime tracking*

NMS Supressão dos não máximos

G-IoU Intersecção Generalizada sobre União

AP *Average Precision*

TP Verdadeiro-positivo

FP Falso-positivo

FN Falso-negativos

HOTA *Higher Order Tracking Accuracy*

CAPÍTULO

1

Introdução

A agricultura é um dos principais setores da economia brasileira. De acordo com o Instituto Brasileiro de Geografia e Estatística (IBGE), estimou-se que no ano de 2020, o Brasil alcançaria um Valor Bruto da Produção Agropecuária - VBP de aproximadamente, 771 bilhões de Reais [1]. Outro setor que apresenta crescimento é a silvicultura, que representa o setor das florestas plantadas [2].

O setor da silvicultura possibilita o desenvolvimento econômico e ambiental, oferecendo emprego e renda para a população e auxiliando com a redução das mudanças climáticas [2]. Segundo IBGE, no ano de 2020, a área de florestas cultivadas em todo o território nacional alcançou um total de 9,61 milhões de hectares. O cultivo de eucalipto representa aproximadamente 77,28%, equivalente a 7,43 milhões de hectares [3]. A aplicação do material extraído se divide principalmente para a produção de celulose e papel que representam 36% do total extraído, carvão vegetal que equivale em torno de 12% do total extraído, painéis de madeira que corresponde a 6% da produção, o restante é destinado a outros produtos [2].

Nas plantações florestais presentes no Brasil, uma das principais pragas e que afetam intensamente a produção, são as formigas cortadeiras. Esses insetos consomem muita vegetação, atacando diferentes espécies de plantas e causando a desfolha até a morte da planta, de mudas até árvores [4]. A desfolha causada pelas formigas resulta em perdas para as plantações florestais, especialmente para plantações de Eucaliptos, necessitando assim do combate à praga. Para o combate às formigas, são utilizados produtos químicos, juntamente com o monitoramento da plantação [5]. O monitoramento da plantação e formigueiros é realizado por pessoas que localizam e quantificam os ninhos desses insetos [6].

O emprego das técnicas atuais de controle das plantações, apresentam dificuldades quando aplicadas em grandes plantações, visto que é necessário, cobrir um espaço para avaliar o estado das plantas [7], além de necessitar de profissionais para realizar as ações necessárias. Com a aplicação de novas tecnologias, é possível entregar um maior controle ao produtor possibilitando diminuição das perdas da produção, em comparação com técnicas que aplicadas atualmente. Um exemplo de aplicação de novas tecnologias foi a inspeção de qualidade de alimentos, que a princípio era realizada por pessoas que observavam e definiam a qualidade dos produtos alimentícios [8].

Com a captura de imagens é possível utilizar as técnicas de visão computacional para o processamento e análise das imagens de maneira automática. Essas tecnologias quando combinadas com técnicas de aprendizagem de máquina permitem realizar o reconhecimento de padrões nas imagens. Nesse sentido técnicas de *Deep Learning* para o reconhecimento de padrões em imagens foram exploradas neste trabalho para a identificação e o rastreamento de formigueiros em imagens de plantações de Eucaliptos. Especificamente, foram estudadas técnicas para a detecção e rastreamento de objetos em imagens. O rastreamento de objetos acontece através da detecção de um objeto de interesse presente em uma imagem e o acompanhamento da trajetória do objeto nos quadros seguintes. A aplicação dessas técnicas permite auxiliar na etapa de monitoramento dos formigueiros da plantação e assim ajudar no controle e na tomada de decisão de maneira rápida. Possivelmente minimizando as perdas na produção, buscando redução nos custos e alta eficiência. Os resultados obtidos foram avaliados tanto com métricas de avaliação em problemas de detecção de objetos e rastreamento e avaliados 3 detectores de objetos combinados com 4 métodos de rastreamento. Entre os métodos de rastreamento foi proposto uma variante do algoritmo *Sort* que se destacou em relação aos métodos avaliados. Como contribuição dessa dissertação, podemos destacar:

- Preparação de uma base de dados de imagens de formigueiros anotadas registrando a localização dos formigueiros e atribuindo um identificador único em cada formigueiro;

- Proposição de um método de rastreamento para o problema de rastrear formigueiros a partir dos objetos detectados via métodos de detecção de objetos

- Permitir o avanço das pesquisas nesse tipo de cenário ainda não explorados na literatura disponibilizando as bases de dados bem como o método proposto.

1.1 Hipóteses

Considerando o contexto apresentado e a necessidade da aplicação de tecnologias para uma melhor eficácia na produção e controle por parte do produtor, este estudo investiga as seguintes hipóteses:

- A aplicação dos métodos de detecção de objetos ao problema de identificação de formigueiros funciona de forma adequada.
- Considerando as detecções obtidas, os métodos de rastreamento de objetos obtêm bons resultados na contagem de formigueiros em plantações de Eucaliptos.

1.2 Objetivos

O objetivo geral deste trabalho foi avaliar o potencial de métodos de detecção e rastreamento de objetos em aplicações voltadas para a silvicultura de precisão, com o objetivo da contagem de formigueiros presentes em plantações de eucalipto.

Os objetivos específicos são:

- Preparar a base de dados de vídeos de formigueiros e anotar cada *frame* (quadro de imagem em video) dos objetos e seus respectivos número de identificação únicos que foram utilizados para treinar os modelos de detecção bem como a sua avaliação.
- Estudar e avaliar métodos do estado-da-arte em detecção de objetos para a identificação de formigueiros em imagens;
- Estudar e avaliar os métodos do estado-da-arte para o rastreamento dos objetos identificados nas imagens;
- Estudar a ferramenta *MMTracking* para a implementação de algoritmos e métricas de avaliação para a realização de experimentos e a otimização dos parâmetros.

1.3 Organização do Texto

No Capítulo seguinte faremos uma revisão de literatura, contendo: uma introdução sobre a área de Visão Computacional, introdução aos conceitos relacionados as Redes Neurais e uma revisão dos principais métodos de detecção de objetos e rastreamento de objetos. A metodologia proposta nessa

dissertação é apresentada no Capítulo 3. Os resultados dos métodos de detecção de objetos e rastreamento são apresentados no Capítulo 4. Por fim, as conclusões e os trabalhos futuros são apresentados no Capítulo 5.

CAPÍTULO

2

Revisão de Literatura

2.1 Visão Computacional

A visão computacional refere-se à forma como um computador enxerga, com a coleta e análise de dados visuais a partir de imagens e vídeos. Em outras palavras a visão computacional permite que o computador possa extrair um conjunto de características que compõem uma imagem [9]. Dentre as diversas técnicas de análise de dados visuais podemos destacar a Detecção de Objetos.

2.1.1 Detecção de Objetos

A detecção de Objetos refere-se identificação e classificação dos objetos presentes na imagem, o *pipeline* de detecção de objetos está representado na Figura 2.1. Tipicamente esses métodos consistem em uma Rede Neural Convolucional para gerar características das imagens (extração de características) para em seguida realizar a identificação de regiões nas imagens contendo os objetos (regressão para estimação da localização dos objetos) de interesse e a classificação de cada região encontrada.

Existem no momento dois tipos de detectores de objetos que se destacam: detectores de dois estágios, que utilizam uma Rede de Proposta de Região que propõe regiões candidatas de interesse no primeiro estágio e envia essas propostas de região para classificação dos objetos e a regressão de caixa delimitadora do objeto para estimar 4 valores para representar as caixas delimitadoras (coordenada x, y, o comprimento e a altura). Esses modelos obtêm bons resultados na precisão, porém normalmente são mais lentos. Os detectores de estágio único, tratam a detecção de objetos como um problema de regressão

simples, buscam entre os possíveis objetos com suas caixas delimitadoras, na imagem e as classificam. Esses modelos alcançam taxas de precisão mais baixas, mas são muito mais rápidos do que os detectores de objetos de dois estágios [10].

Resumidamente, os detectores de objetos atuais, funcionam representando cada objeto na imagem por meio de um limite em forma de caixa que envolve o objeto. Para cada uma dessas caixas delimitadoras, um classificador (em geral, *softmax*), categoriza o conteúdo da caixa. Os detectores de um estágio, extraem características e permitindo assim a buscam entre os possíveis objetos com suas caixas delimitadoras, chamadas âncoras, na imagem e as classifica sem definir o objeto da caixa. Os detectores de dois estágios funcionam de forma diferente, eles utilizam o conteúdo da caixa definida e recalculam os parâmetros que estão na imagem para cada uma das possíveis caixas e, com isso os classificam.

2.1.2 Rastreamento de Objetos

o rastreamento de Objetos trata-se de uma abordagem que permite o cálculo da trajetória de um ou vários objetos específicos em cena. Com isso, o rastreamento de objetos pode ser usado para a contagem e localização de itens em uma cena, uma vez que cada trajetória é associada a um objeto. Para lidar com o rastreamento de vários objetos existe o Rastreamento de Múltiplos Objetos do inglês *Multiple Object Tracking* (MOT). A tarefa do MOT refere-se na localização de múltiplos objetos, na sua identificação e no cálculo de suas trajetórias individuais, em uma sequência de imagens. Esse método torna possível a aplicação a diversas classes de objetos, por exemplo, pedestres, veículos, ou até mesmo grupos de animais ou insetos.

É possível agrupar os trabalhos MOT em dois grupos, dependendo da maneira em que realizam o rastreamento. O rastreamento baseado em detecção do inglês, *Detection-Based Tracking* (DBT) e o rastreamento livre de detecção em inglês, *Detection-Free Tracking* (DFT). No DBT os objetos são detectados em cada quadro, após isso efetua-se a conexão das possíveis trajetórias aos respectivos objetos. No DFT executa-se a marcação manual dos objetos, no primeiro quadro do vídeo, em seguida, atinge-se o rastreamento da trajetória do objeto. A abordagem DBT é uma das mais populares por lidar da melhor forma com objetos surgindo ou sumindo do campo de visão. Já o DFT por necessitar da inicialização de cada um dos objetos no primeiro quadro do vídeo, não consegue solucionar o surgimento de objetos no campo de visão [11].

O nosso estudo seguirá a linha do DBT. Na Figura 2.2 vemos o fluxo de funcionamento da abordagem DBT, ela inicia-se com a entrada da sequência de imagens, com as imagens aplicar-se o método de detecção de objetos, respon-

sável por detectar os objetos presentes em cada um dos quadros da sequência, vê-se que o detector funciona de forma acoplada ao DBT, permitindo a inserção de diferentes métodos de detecção. Os resultados da detecção passam por uma etapa de extração de características, que permitem futuramente a associação dos objetos detectados em um quadro, com o mesmo objeto detectado em outro quadro, através da comparação das características entre os objetos.

A associação de dados representa a etapa responsável por identificar uma correspondência entre um objeto detectado anteriormente e outro detectado futuramente na sequência de imagens, com base nas características extraídas na etapa anterior. Um exemplo de características, são histograma de cores, baseado em recurso de gradiente, fluxo óptico e padrão binário local.

Após a associação de dados, aplica-se a gestão de trajetórias, ela contém três etapas essenciais: atualização da trajetória, encerramento da trajetória, e a inicialização da trajetória. Após essa etapa obtém-se os objetos e suas respectivas trajetórias na sequência das imagens

- Inicialização da trajetória: quando a detecção não está associada a nenhuma trajetória existente, considera-se que a detecção se trata de um novo objeto, que surgiu no campo de visão da imagem. A partir disso inicia-se uma nova trajetória se iniciando no objeto em questão.
- Atualização da trajetória: Quando uma trajetória possui correspondência com uma detecção, atualiza-se seu estado e o vincula toda a trajetória ao objeto.
- Encerramento da trajetória: Quando uma faixa não se associa com outras detecções, considera-se que o objeto tenha saído do campo da imagem, sendo assim, a trajetória daquele objeto será encerrada.

2.2 Redes Neurais

As Redes Neurais Artificiais (RNAs) em inglês *Neural Networks* ou também *Artificial Neural Networks* são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural existente no cérebro dos animais. As RNAs possuem uma enorme capacidade de aprendizagem, utilizando dados que são processados pelos neurônios artificiais. Na Figura 2.3 é possível visualizar a estrutura de uma RNA e seus neurônios artificiais [12].

Na RNA temos, a Camada de Entrada, responsável por receber as informações. Temos também as Camadas escondidas igualmente chamadas de camadas intermediarias, é nela que se encontram os neurônios artificiais. Nesta camada é realizado a extração de características da entrada informada. Por fim temos, a camada de Saída, por onde exibe-se o resultado final da RNA.

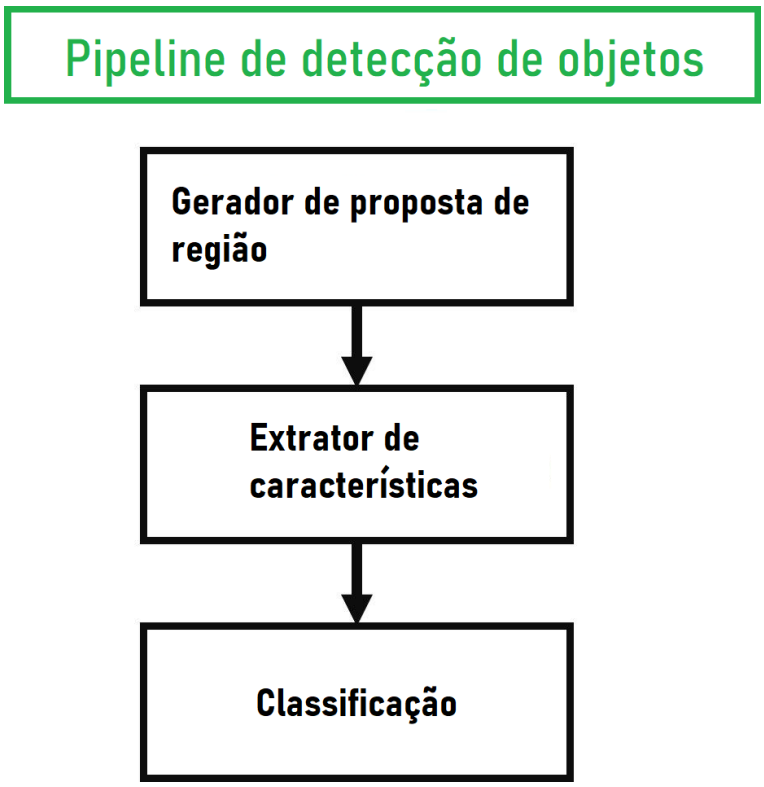

Figura 2.1: *Pipeline* da detecção de objetos.

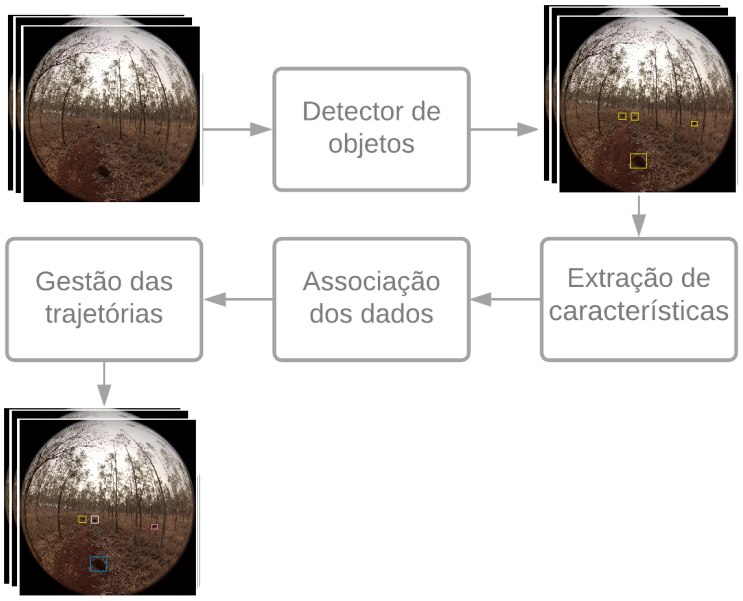

Figura 2.2: Fluxo Rastreamento.

2.2.1 Redes Neurais Convolucionais

As Redes Neurais Convolucionais também chamadas RNC ou CNN são comumente utilizadas para aplicações em imagens, essas redes funcionam de forma similar às redes neurais comuns, diferenciando-se na estrutura. Exemplo de uma RNC na Figura 2.4

Na RNC temos, a camada de convolução que é composta por filtros, eles

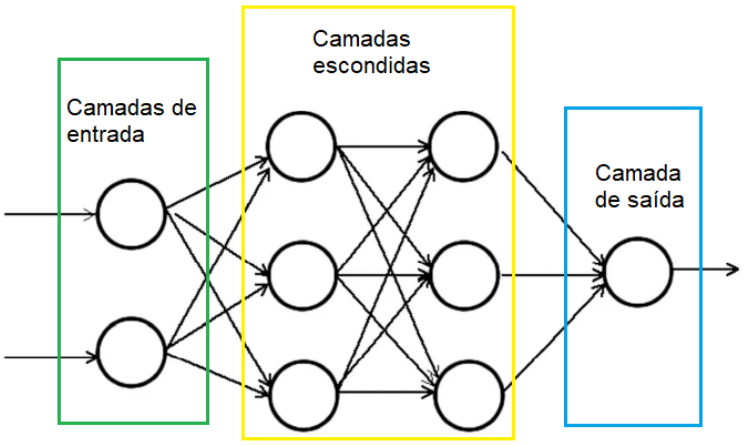

Figura 2.3: Estrutura RNA.

são aplicados a imagem de entrada, obtendo um mapa de característica em retorno. A convolução, resumidamente multiplica os elementos de entrada do filtro por um elemento de entrada, e após isso é feita uma soma desses valores, e assim é gerado o mapa de características como saída. Outra camada importante na estrutura RNC é a camada de *pooling*, o objetivo dessa camada é realizar um tipo de extração de algumas das características da entrada. Obtendo assim, de forma reduzida algumas das características que foram extraídas na camada de convolução.

Por fim temos as Camadas totalmente conectadas, que como o nome diz, cada nó de uma camada está conectado aos outros nós da próxima camada. A entrada para as camadas totalmente conectadas se trata das características obtidas e filtradas pelas camadas de convolução e *pooling* (geralmente a saída dessa camada é transformada em um vetor de valores utilizado como uma representação das características das imagens), com isso as camadas totalmente conectadas farão a interpretação das características recebidas, com isso é obtida a camada de saída que o seu número de neurônios é igual ao número de classes possíveis no problema. Após as operações nas camadas totalmente conectadas tem-se a camada de saída. O tamanho da camada de saída, ou seja, o número de neurônios, é igual ao número de classes no problema [13].

2.3 Trabalhos Relacionados

A Agricultura de precisão tem como objetivo o aumento da eficiência na produção, permitindo coleta e análise de dados de em plantações, possibilitando que com base nas informações coletadas seja possível o controle de pragas, controle dos danos ambientais e até mesmo o controle da irrigação da plantação [8].

A aplicação em conjunto de tecnologias, como visão computacional, apren-

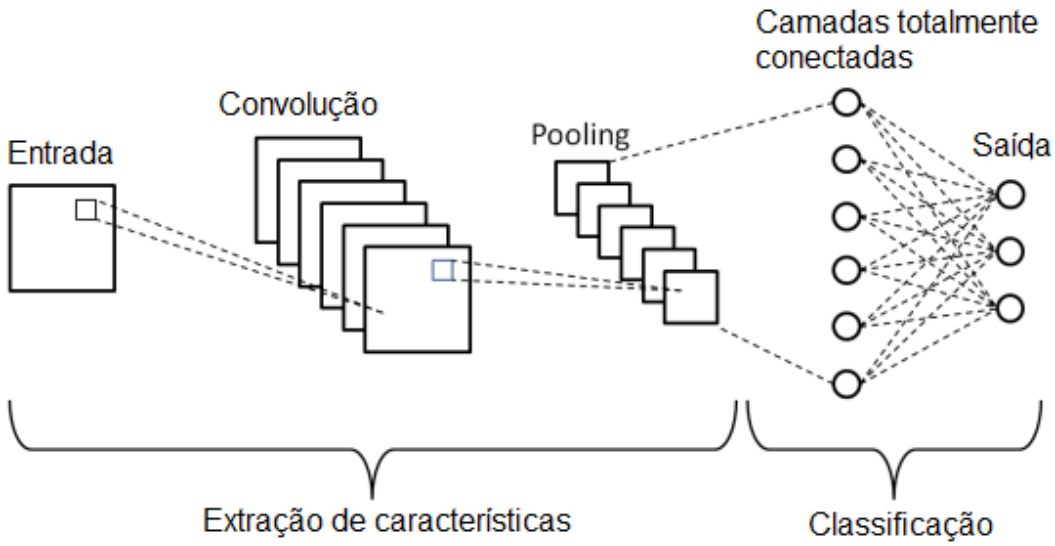

Figura 2.4: Estrutura RNC.

dizado de máquina tem se mostrado promissora na utilização no campo agrícola. A *Deep Learning* é um ramo do aprendizado de máquina que possui um imenso potencial de reconhecimento, destacando-se entre as técnicas de aprendizado de máquina [14]. Essa seção apresenta alguns estudos realizados nos âmbitos de detecção e rastreamento de objetos, aprendizado de máquina e agricultura de precisão. Com o intuito de apresentar um contexto do que vem sendo feito na área de agricultura de precisão e as possibilidades de aplicação ao contexto desse trabalho.

2.3.1 Detecção de Objetos

Utilizando tecnologias como processamento de imagens e aprendizado de máquina e com o intuito de aumento de eficiência e redução do custo no processo de cultivo de mudas de alface hidropônica em [15], foi proposto um método que permite a detecção de mudas problemáticas de alface. Existem duas possibilidades de problemas que possam ser identificados nas mudas, o plantio duplo onde duas plantas crescem em um único local de plantação e a morte da planta, onde nenhuma muda cresce naquele local de plantação. Para a aquisição de imagens foram utilizados uma câmera e alguns aparelhos celulares. Inicialmente foram removidas manualmente, imagens repetidas ou que não possuíssem boa qualidade. Após esse processamento inicial a base de dados resultou em 600 imagens das mudas de alface hidropônicas, desse total 420 imagens foram utilizadas no conjunto treinamento, 60 imagens no conjunto de validação e 120 no conjunto de teste. Com o intuito de melhorar o treinamento do modelo de detecção foi aplicado o aumento de dados no conjunto de treinamento. No aumento de dados, foi aplicada na escala das imagens originais, inversões e rotações, com isso o conjunto de treinamento

alcançou 2940 imagens. Após o pré-processamento as imagens foram rotuladas com as duas classes de mudas problemáticas. Por serem muito grandes o tamanho das imagens fora ajustado para 500 x 375 *pixels*. Como modelo de detecção de objetos foi selecionado o *Faster R-CNN*. Pelas mudas de alface hidropônicas serem pequenas nas imagens, foi utilizado no modelo de detecção o *backbone HRNet* para a extração de recursos das imagens. A métrica de avaliação utilizada foi a precisão média. Com a abordagem utilizada a média dos resultados para o plantio duplo e a morte da planta foram de 86,2% superando outros detectores, como por exemplo o *VFNet* que teve o resultado 78,2%.

Em [16] foi proposto o desenvolvimento e aplicação de uma ferramenta com processamento em nuvem, chamada *Agroview*. Resumidamente a aplicação permitiria o tratamento das imagens obtidas através drones equipados com câmeras RGB e o retorno de informações sobre a plantação, como detecção das plantas, contagem e medição. O primeiro passo do processo é a aquisição das imagens. No estudo de caso apresentado no artigo foram utilizados dois drones equipados com câmeras permitindo o sensoriamento remoto de uma plantação de cítricos. Após a aquisição das imagens, o usuário opta por gerar o ortomosaico em um software diferente, ou utiliza o *Agroview* para realizar o processamento e gerar o mapa ortomosaico, esse mapa seria algo como a junção de diferentes fotos em perspectiva aérea que permite a visualização da plantação. Para gerar o mapa ortomosaico o *Agroview* utiliza um kit de ferramentas de código aberto chamado *Open Drone Map*. Com o mapa disponível o usuário seleciona a área onde deseja realizar as detecções das árvores. Antes de iniciar a detecção as imagens passam por um pré-processamento em que as cores são calibradas. Na etapa de detecção e processamento das imagens, duas redes profundas foram utilizadas, primeiramente foi aplicada a *Faster R-CNN* combinada com o *backbone ResNet101*, ela realiza a detecção e encontra o número de árvores e realiza um processamento para identificar os padrões de orientação da plantação e a espaço entre as linhas de árvores na plantação. Em seguida a outra Rede neural convolucional é aplicada, para um refinamento das detecções obtidas, essa rede é a *YOLO* com o *backbone Darknet19*. E por fim é retornado ao usuário o processamento com as informações obtidas. Como métrica de avaliação do método propostos, foi utilizado o Erro Absoluto Médio Percentual - MAPE, que representa a diferença entre uma medida obtida com o método e o valor padrão, que neste estudo são as previsões do algoritmo e as medições manuais. O MAPE obtido para a detecção e contagem de árvores, foi de 3,18% (aproximadamente 97% de precisão).

Em [17], foi realizado um estudo sobre a detecção de danos em plantações de milho, com base na detecção de objetos em imagens aéreas, obtidas com

uma câmera RGB conectada ao drone. No estudo foi avaliado o quebramento do colmo (comum no cultivo do milho), que ocorre quando o colmo perde a resistência e não consegue sustentar a espiga, onde ocorre a quebra. Com isso, um conjunto de dados foi construído, com um total de 478 imagens. Na detecção de objetos foram utilizadas três redes, são elas: *Faster R-CNN*, *YOLOv2* e *RetinaNet*. Os detectores foram avaliados em razão da capacidade de detectar as partes com o colmo quebrado da plantação. Para avaliar o desempenho de cada modelo criado com diferentes detectores de objetos, foi aplicada a, precisão média e o *Faster R-CNN* obteve 52,81%, *YOLOv2* obteve 89,90% e *RetinaNet* 84,24%.

Os autores, em [18] propõem a construção de um drone com baixo custo, que possibilite a captura de imagens, que permita a detecção de objetos aplicados à contagem de plantas ornamentais que se encontram na natureza. A princípio, são obtidas as imagens através de uma câmera RGB instalada no drone que permite a coleta das imagens em diferentes condições de clima e luz. Juntamente são baixadas imagens da internet com o intuito de aumentar a quantidade de imagens. Em seguida, aplica-se o aumento artificial de dados, que consiste em rotacionar ou alterar cores das imagens. Após isso realiza-se o redimensionamento dessas imagens para 300 x 300 *pixels*. O conjunto de dados contém 2250 imagens no total dividida em três categorias, 300 imagens capturadas, 200 imagens por baixadas da internet e 250 imagens aumentadas. O conjunto de dados divide-se em duas partes treinamento e teste a proporção para cada imagem é de 1/3, que equivale a 500 imagens para o treinamento e 250 imagens para o teste em cada uma das categorias. A partir disso é aplicada a detecção e classificação do objeto, que categoriza a planta, retornando o resultado positivo caso seja uma planta ornamental. O modelo de detecção utilizado *YOLOv3* obteve como melhor resultado a precisão média de 79,85% na detecção das plantas.

O estudo [19] propõe a detecção de objetos em campos de plantação de soja, com o objetivo principal de construir um sistema que permita detectar as vagens e flores da planta. Através de imagens obtidas em campos de soja utilizando uma câmera RGB, capturando as imagens em solo, a quantidade total de imagens capturadas foi de 12.659 imagens. Com a detecção dos objetos presentes nas imagens é analisado se a quantidade de flores influência nos rendimentos da safra. Para efetuar a detecção dos objetos foram utilizados os detectores, *RetinaNet*, *Faster R-CNN* e *Cascade R-CNN*. O desempenho dos detectores é avaliado pela precisão média. Os resultados totais obtidos foram 89,6% para o *Cascade R-CNN*, 83,3% para o *RetinaNet* e 88,7% *Faster R-CNN*. Outros testes foram realizados para avaliar a precisão entre as flores e as vagens, para isso os modelos foram treinados com dois *backbones*, sendo

eles o *ResNet50* e o *ResNet101*. Em todos os casos o *ResNet101* se mostrou superior e o *Cascade R-CNN* superou os outros detectores, usando o *backbone ResNet101*. A média de precisão foi de 86,6% aplicado as flores e 92,5% aplicado as vagens, o *Faster R-CNN* com o *backbone ResNet101* obteve como medidas de precisão 85,8% para as flores e 91,7% para as vagens e por último o *RetinaNet* utilizando o *ResNet101* alcançou médias de precisão de 81,2% para as flores e 85,5% para as vagens.

2.3.2 Rastreamento de Objetos

No estudo [20] é avaliada a abordagem de baseado em detecção (DBT). As imagens utilizadas são do conjunto de dados *on the VisDrone* 2018 dataset obtidas através de drones. Para realizar as detecções necessárias para o rastreamento, utilizou-se um *framework* de detecção que combina *YOLOv3* e *RetinaNet*. As escolhas de arquiteturas de detecção se devem ao fato de os autores avaliarem que a *YOLOv3* é rápida e eficaz para a detecção de objetos de médio porte, porém não trabalha adequadamente para tamanhos pequenos. De outra forma, o *RetinaNet* se destaca em casos onde os objetos são menores.

O *framework* de detecção é o responsável por retornar as detecções de um *frame* de imagem, que consiste nos novos objetos detectados neste quadro. As detecções obtidas de cada um dos quadros foram aplicadas a uma RNC, aplicou-se também o conjunto de dados *person re-identification dataset*. O algoritmo de rastreamento se baseia no *Deep SORT*, nessa etapa aplica-se o filtro Kalman que realiza a estimativa das variáveis de estado, utilizando uma matriz de associação que é gerada pelo algoritmo. Essa matriz relacionada a detecção, juntamente com as características de aparência dos objetos, possibilitando a predição da posição dos alvos nos futuros *frames* da imagem. A abordagem apresentou bons resultados nos experimentos, porém o mais importante aspecto desse estudo foi constatar que a qualidade da detecção dos objetos mantém-se muito importante no rastreamento de objetos.

No estudo [21] apresenta-se uma abordagem para estimativa de rendimento e contagem de frutos em vídeos de maçãs, laranjas e abóboras. Para estimar a quantidade de frutos utilizou-se a detecção e rastreamento de objetos. Na etapa de detecção aplicou-se o modelo *You Only Look Once* (*YOLO*). As detecções obtidas são usadas como entrada para o modelo de rastreamento, *Deep SORT*. Problemas de rastreamento de objetos normalmente aplicam-se em pedestres e para adaptar o modelo ao rastreamento de frutas, implementou-se a rede *ResNet* para a extração de recursos do *Deep SORT*, utilizando-se do fornecimento de pesos pré-treinados no conjunto de dados da Imagenet, esse conjunto de dados possui diferentes classes de objetos, incluindo diversos tipos de frutas. Eliminando a exigência de treinamento do extrator de recursos e

possibilitando a aplicação do modelo *ResNet* em diferentes frutas. Para avaliar os resultados obtidos com o modelo de estimativa de rendimento, calculou-se a precisão comparando o rendimento previsto pelo modelo treinado e a contagem de verdade fundamental que é realizada manualmente por um humano. obteve-se uma precisão de 91% em imagens de maças. Também testou-se a contagem de laranjas e abóboras, e obteve-se uma precisão obtida de 79% para laranjas e 93,9% para abóboras.

CAPÍTULO

3

Metodologia

Neste capítulo é apresentado o método aplicado a esse estudo, a Figura 3.1 está o fluxo de processos da metodologia proposta. É objetivado que a metodologia desenvolvida, seja aplicada à detecção e rastreamento dos formigueiros em vídeos com o intuito de controle de pragas em plantações de eucalipto. A utilização do rastreamento, busca evitar que a contagem fique comprometida com a contagem do mesmo objeto mais de uma vez.

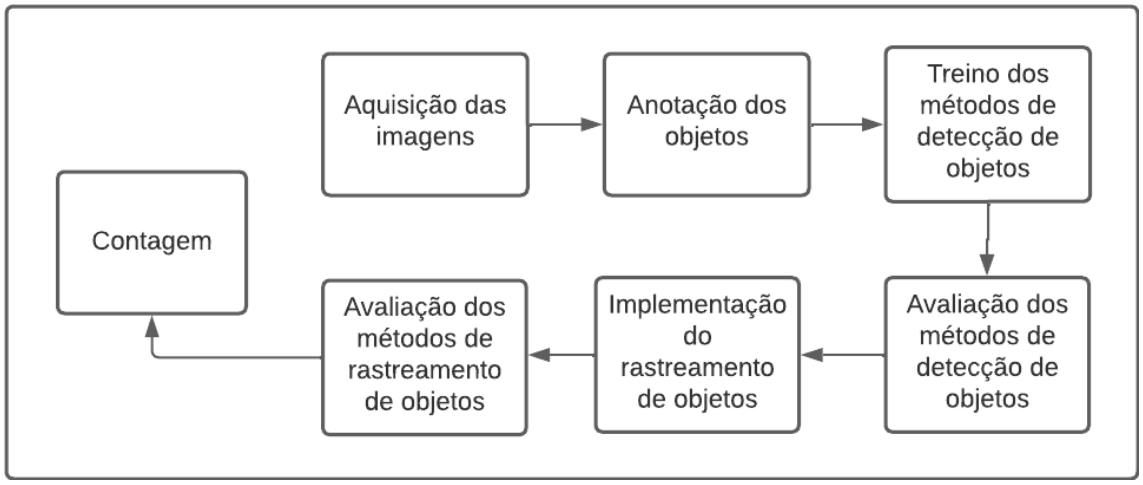

Figura 3.1: Fluxo da metodologia.

3.1 Aquisição das imagens

As imagens foram gravadas em uma plantação de eucalipto usando uma câmera RGB *GoPro MAX* com uma lente *Fisheye*, que proporciona um campo de visão mais amplo e adiciona a imagem uma forte distorção. As imagens

foram capturadas com uma pessoa segurando a câmera com a mão sem o uso de estabilizador e caminhando na plantação de eucaliptos. Na Figura 3.2 é possível visualizar um *frame* do vídeo contendo formigueiros.

Figura 3.2: *Frame* do vídeo contendo formigueiros.

3.2 Anotação dos objetos

Na etapa de anotação dos objetos, utilizou-se a ferramenta *DarkLabel*. A ferramenta possibilita rotular caixas delimitadoras de objetos em vídeos e imagens, com diferentes formatos de anotação, possibilitando realizar anotações para detecção e rastreamento de objetos. Neste trabalho utilizou-se o formato *MOTChallenge*que permite que seja informado o ID de cada objeto anotado, tornando cada objeto presente na imagem único e com isso a avaliação dos método de rastreamento é possível. Ao iniciar a ferramenta é necessário selecionar o arquivo que deseja anotar, podendo ser imagem ou vídeo. Para a anotação da base de dados utilizou-se dois vídeos obtidos na etapa de aquisição de imagens, esses vídeos foram obtidos na mesma plantação, mas em áreas diferentes. Foi necessário dividir os vídeos em outras partes que são necessárias para evitar movimentos bruscos com a câmera, fazendo com que cada parte do vídeo siga em uma direção. A ferramenta de anotação percorre

os *frames* do vídeo e os objetos são rotulados com seus respectivos IDs. É possível visualizar um exemplo da anotação na Figura 3.3. O processo de anotação dos objetos, resultou na criação de duas base de dados diferentes, com cerca de 11891 imagens no total, a primeira base de dados derivou-se de um vídeo com duração de quatro minutos e quarenta segundos e conta com um total de 8071 imagens e foi dividido em 5 partes do vídeo, a segunda base de dados originou-se de um vídeo com duração de dois minutos e dezessete segundos e possui 3820 imagens e foi divido em 5 partes do vídeo. Na Tabela 3.1 vemos outras informações pertinentes da base de dados que ajudam a entender melhor o problema de detecção de formigueiros, como quantidade Objetos, tamanho médio dos objetos em *pixels* e quantidade média de formigueiros por *frames*. Uma das principais diferenças entre as bases de dados GPBK0196 e GPBK0197 é a quantidade de objetos.

Figura 3.3: Anotação formigueiro com IDs.

3.3 Detecção dos objetos

Na etapa de detecção dos objetos, utilizou-se três diferentes métodos de detecção de objetos, são eles: *Faster R-CNN* [23], *RetinaNet* [24] e *VFNet* [25]. A implementação dos métodos de detecção foi realizada através ferramenta de

Corte do vídeo	Quantidade Objetos	Tamanho médio dos objetos em *pixels* (largura x altura)	Quantidade média de formigueiros por *frames*
GPBK0196_Corte1	8	86,87 x 86,12	1,76
GPBK0196_Corte2	8	76,02 x 80,83	1,89
GPBK0196_Corte3	9	104,76 x 108,80	1,61
GPBK0196_Corte4	4	76,51 x 76,57	1,90
GPBK0196_Corte5	1	59,12 x 75,04	1
GPBK0197_Corte1	2	112,01 x 108,71	1
GPBK0197_Corte2	2	86,68 x 80,95	1
GPBK0197_Corte3	1	153,92 x 160,67	1
GPBK0197_Corte4	1	145,08 x 124,82	1
GPBK0197_Corte5	1	57,81 x 93,18	1

Tabela 3.1: Informações das bases de dados

código aberto *MMTracking*, com pesos pré-treinados no *ImageNet*, para cada um dos detectores [22].

3.3.1 Faster R-CNN

O *Faster R-CNN* é um modelo de dois estágios, uma das suas principais características é a utilização de uma rede de proposta de região (RPN). A RPN gera um número pré-definido de propostas. Além disso utiliza-se a camada *ROI Pooling* que permite extrair um vetor de características de cada proposta de região e após isso acontece a classificação das regiões (*bounding boxes* ou caixas retangulares delimitadoras) detectadas [23].

Após diferentes testes, o *backbone ResNet101* destacou-se em relação a sua versão com 50 camadas (*ResNet50*) e foi aplicado juntamente ao modelo de detecção *Faster R-CNN* para a detecção dos formigueiros. O modelo *Faster R-CNN* com o *backbone ResNet101* foram treinados por 9 épocas. Como otimizador empregou-se o SGD (descida de gradiente estocástico) com uma taxa de aprendizado de 0,02. Para aumentar a quantidade de dados aplicou-se as técnicas de aumento de dados (*data augmentation*): girar, inverter, dimensionar, ajuste no brilho e contraste.

3.3.2 RetinaNet

O *RetinaNet* é um modelo de um estágio, uma das suas principais características é a aplicação de uma função de perda focal a fim de solucionar o problema de desequilíbrio de classe com modelos de detecção de objeto de estágio único [24].

O modelo de detecção *RetinaNet* foi aplicado juntamente ao *backbone Res-*

Net101, a escolha do *backbone* se deu por apresentar melhores resultados em comparação com o *RetinaNet + ResNet50*. O modelo *RetinaNet* com o *backbone ResNet101* foram treinados por 9 épocas, como realizado com o algoritmo *Faster R-CNN*. Como otimizador utilizou-se o SGD com uma taxa de aprendizado de 0,0002. No aumento de dados aplicou-se as mesmas técnicas que no *Faster R-CNN*: girar, inverter, dimensionar, ajuste no brilho e contraste.

3.3.3 VFNet

O *VFNet* é um modelo de um estágio, que visa classificar o maior número de detecções candidatas na detecção de objetos, isso influência diretamente no desempenho do modelo. Para isso utilizam uma função de perda, chamada *Varifocal Loss* [25].

O modelo de detecção *VFNet* foi aplicado juntamente ao *backbone ResNet101*, o motivo de escolha do *backbone ResNet101* se deu por obter melhores resultados com o modelo *VFNet* em contrapartida com o *ResNet50*. O modelo *VFNet* com o *backbone ResNet101* foram treinados por 9 épocas. Como otimizador utilizou-se o SGD com uma taxa de aprendizado de 0.0002. E no aumento de dados as mesmas técnicas anteriores foram aplicadas.

3.4 Rastreamento dos objetos

O intuito de utilizar a técnica de rastreamento é evitar com que a detecção e contagem dos objetos seja afetada, por contar o mesmo objeto mais de uma vez. Com o rastreamento, o acompanhamento do objeto em uma sequência de quadros das imagens é realizado desde o momento em que ele entra na imagem e é detectado, até o momento em que sai.

Em comparação com a detecção de objetos, nela um objeto é detectado, delimitado (caixa retangular) e classificado. O processo se limita a um quadro do vídeo. No rastreamento de objeto, em contrapartida, é buscado efetuar o rastreio dos objetos em toda a sequencia de quadros. Se na detecção forem encontrados três objetos em um quadro, no rastreamento deverá ser identificada as três detecções de forma separada (cada um com identificador único) e assim rastreá-los nos quadros subsequentes.

Neste trabalho foram investigados quatro diferentes métodos de rastreamento de objetos, são eles: *Deep SORT*, *Tracktor*, *Byte Track* e o *Método Proposto*.

3.4.1 Método Deep SORT

O método de rastreamento *Deep SORT* é composto por dois itens principais, são eles: Filtro de Kalman e *SORT*.

O Rastreamento simples online e em tempo real (*SORT*) se apresenta como um *framework* de rastreamento muito mais acessível, que permite a detecção e o rastreamento de objetos em tempo real. O *SORT* divide-se em 4 estágios: Detecção, Modelo de estimativa, Associação de dados e Criação e Exclusão de Identidades de Trajetórias [26].

- No estágio de detecção que os objetos são obtidos e permitem a criação das trajetórias.

- No estágio de Modelo de estimativa é utilizado o Filtro de Kalman que define um espaço de oito dimensões, dentre elas existe a posição central da caixa delimitadora a altura, e suas respectivas velocidades nas coordenadas da imagem. Esses dados permitem que o Filtro de Kalman funcione iterativamente, extraindo informações dos dados atuais e depois utilizando esses dados para atualização das predições. Com isso é possível associar uma detecção à uma possível trajetória, corrigindo os valores com o objetivo de melhorar as estimativas futuras. Resumindo, o filtro Kalman, utiliza as detecções realizadas e as previsões anteriores, a fim de alcançar uma melhor estimativa do estado atual, de cada objeto e suas possíveis trajetórias.

- No estágio de Associação de dados calcula-se a distância de interseção sobre união (IOU) entre as detecções obtidas e as trajetórias previstas. Para isso existe uma matriz de custo que permite definir a nova detecção a trajetória com o menor custo possível.

- O *SORT* aplica a gestão de trajetórias, no estágio de Criação e Exclusão de Identidades de Trajetórias. Para o controle da identidade das trajetórias é preciso a criação e exclusão de trajetórias. Para a criação, consideram-se as detecções que obtenham uma sobreposição maior que um valor minimo de IOU. Além disso a nova trajetória passa por um período em que é preciso obter outras detecções com o intuito de evitar criar uma trajetória inexistente. Caso as trajetórias não recebam uma nova detecção em um limite máximo de *frames*, são encerradas, Caso um objeto reaparecer, cria-se uma nova identidade.

Apesar de obter bons resultados, o modelo *SORT* apresenta uma certa dificuldade na criação de trajetórias, criando inúmeras identidades para cada objeto. Com o propósito de melhorar o desempenho do modelo, surge o *Deep*

SORT que implementa uma nova métrica de distância baseada na aparência do objeto. Para isso utiliza-se um modelo de classificação, treinado com base no conjunto de dados, que possibilita a associação entre as novas detecções e as trajetórias existentes além de permitir rastrear os objetos que não apareçam em um maior número de quadros, e também reduzindo o número de trocas de identidade [27].

3.4.2 Método Tracktor

O método de rastreamento *Tracktor* tem como característica principal, o uso da regressão de caixa delimitadora gerada pelo detector para prever a posição do objeto no quadro seguinte. Resumidamente, no primeiro quadro o rastreador inicializa as trajetórias com as primeiras detecções e a partir disso aplica-se a regressão da caixa delimitadora das trajetórias existentes até o quadro atual. Com isso é possível estimar a posição do objeto no quadro seguinte. Além da regressão o *Tracktor* utiliza outras duas estratégias para manter a identidade dos objetos com o passar dos quadro do vídeo, são elas: *Motion model* e um modelo de re-identificação [28].

- O *Motion model* serve para tratar dois problemas comuns, a câmera em muito movimento e vídeos com baixa quantidade de quadros por segundo. Para vídeos com câmera em movimento, aplica-se uma compensação ao movimento da câmera e para vídeos com baixas taxas de quadros aplica-se uma suposição de velocidade constante para os objetos.

- O modelo de re-identificação utiliza-se de uma rede neural, armazenando objetos que foram perdidos e comparando com objetos que acabaram de ser detectados.

O *Tracktor* aplica a gestão de trajetórias, de forma parecida com os outros, a trajetória recém criada passa por um período em que é preciso obter outras detecções, com o intuito de evitar criar uma trajetória inexistente e caso as trajetórias não recebam uma nova detecção em um limite máximo de quadros, são consideradas perdidas, Caso um objeto reaparecer, ele passa pelo modelo de re-identificação.

3.4.3 Método Byte Track

O *Byte Track* é um método de rastreamento, que apresenta um modo de aplicar a detecção e o rastreamento aos objetos que estão oclusos, isso tudo sem a utilização de um modelo de classificação para a re-identificação. O que diferencia o *Byte Track* da maioria dos métodos de rastreamento é que ele não

seleciona apenas detecções acima de um determinado limite de confiança do detector de objeto (*score*) como entrada para associação de dados.

O *Byte Track* funciona, basicamente dividindo-se em dois estágios. No primeiro estágio são selecionados os objetos com as pontuações altas (acima de um limiar definido), claramente visíveis para o modelo. Já no segundo estágio o método propõe-se a selecionar os objetos que possuem as pontuações mais baixas (abaixo de um segundo limiar definido), esses objetos muitas vezes estão oclusos ou o fundo é detectado erroneamente. Cada detecção é guardada e separada conforme seu tipo, as de pontuações altas e as de pontuações baixas. Após isso aplica-se um filtro de Kalman que permite prever os novos objetos a partir do quadro atual. E com isso ocorre a primeira associação, como é chamada a etapa que associa as detecções que possuem pontuação alta com as informações dos objetos já rastreados em quadros anteriores (lista de trajetórias ou *trackers*). A segunda associação, nome dado a etapa que associa as detecções que possuem pontuações baixas, acontece logo após a primeira associação é nessa etapa que ocorre a tentativa de recuperação dos objetos oclusos. Caso as detecções de baixa pontuação obtenham correspondência a uma trajetória, são excluídas. Caso as trajetórias sobrevivam a segunda associação e desaparecem após isso, são mantidas por um certo número de quadros e após isso são encerradas, considerando que o objeto tenha saído do vídeo e dessa forma o *Byte Track* aplica a gestão de trajetórias [29].

3.4.4 Método Proposto

Por fim temos o método proposto, este método baseia-se no *SORT*. Nele busca-se uma melhora, em comparação aos outros métodos de rastreamento de múltiplos objetos avaliados. Diferentemente dos métodos propostos para o cenário de identificação de pessoas, os formigueiros não apresentam muita discrepância entre eles o que permite desconsiderar o componente de Re-Identificação, pois a Re-Identificação se baseia especificamente na extração de características. Nesta variação do modelo *SORT* ao invés do rastreador se basear apenas nas detecções superiores a um limite definido são considerados também os objetos que obtiveram uma pontuação baixa mas que estão muito próximas (de acordo com um limiar, no nosso caso 0.6) as trajetórias já identificadas. A premissa (assim como no *ByteTrack*) é que esses objetos podem estar oclusos e por isso obtiveram uma pontuação baixa do modelo. Como mostrado no Algoritmo 1, o método recebe como entradas os *bounding boxes* (caixas retangulares delimitadores, de maneira abreviada *bboxes*) e o limite mínimo de pontuação e emite como saída as trajetória dos objetos. Os *bounding boxes* são uma lista de vetores de cinco posições contendo as coordenadas das caixas delimitadoras e a pontuação dos objetos detectados. O

limite mínimo de pontuação é definido para filtrar as detecções iniciais (no primeiro quadro).

O método possui duas etapas iniciais, na primeira verifica-se a inexistência de trajetórias ou caixas delimitadoras, caso seja verdadeiro seleciona-se todas as detecções superiores ao limite mínimo de pontuação definido, dessa forma o rastreador pode iniciar as trajetórias com essas detecções selecionadas. A função novasTrajetórias, representa a inicialização das novas identidades de objetos por definir cada coordenada dos objetos como as posições das trajetórias iniciais e também definir para cada objeto um ID que vai de 0 até o número de objetos selecionados e assim as primeiras trajetórias de rastreamento são geradas e atualização dos parâmetros do Filtro de Kalman, porém de forma provisoria até alcançar um limite de quadros (definido pelo usuário, utilizamos 2 quadros) para confirmar a identidade.

Caso já existam trajetórias aplica-se a supressão dos não máximos (NMS) a todos os *bounding boxes*, a fim de filtrar as caixas delimitadoras que estão sobrepostas, com um valor definido. Nesse caso o valor limite para o NMS de 0,1, um valor bem baixo possibilitando a filtragem, caso ocorra qualquer mínima sobreposição e mantendo apenas as caixas delimitadores de maior pontuação. A consequência da escolha de um valor baixo para a supressão dos não máximos é a criação de uma dependência de um cenário que possua uma baixa densidade de formigueiros, ou seja poderia remover as caixas delimitadoras de formigueiros que estão muito próximos entre si, mas no nosso caso os formigueiros estão bem dispersos. Após a filtragem via NMS, representada pela função NMS, utilizam-se todas trajetórias ativas (que já se mantiveram por 2 quadros consecutivos) ou pendentes (recém criadas) a fim de combiná-las com as detecções, utiliza-se a função *bbox_overlaps*, juntamento com a Intersecção Generalizada sobre União (G-IoU) [31] que permite o cálculo da distância entre os *bboxes* detectados e os *bboxes* pertencentes as trajetórias anteriores. Quando os valores de GIoU são maiores que 0 e menores que 1 identificamos a sobreposição entre o par de *bboxes* e nesse caso a fórmula é o próprio IoU, quando os valores são negativos temos que não houve sobreposição e o par de *bboxes* estão distantes. A premissa é que pode acontecer das trajetórias obtidas estarem distantes dos objetos obtidos no quadro corrente por vários motivos (movimentação brusca da câmera ou baixa taxa de captura das imagens) e apenas considerar o IoU pode não ser suficiente nesses casos o que acarretaria na criação de novas trajetórias. Essa etapa é realizada pela função combinarTrajetórias que também atualiza o filtro de Kalman considerando os *bboxes* selecionados. Após esse processo, são selecionados também entre os *bboxes* não combinados na função combinarTrajetórias aqueles que possuem pontuação acima de um limiar (no nosso caso em 0.6), o que pode configurar

potenciais novas trajetórias.

Algoritmo 1: Variação do método de rastreamento de objetos *SORT*

Entrada: *bboxes[frame]* (Detecções para cada *frame* do video),
limite_Pontuação (limite mínimo de pontuação),
numero_frames_video (Número total de *frames* do vídeo)
Saída: *tracks* (Resultados do rastreamento)
início
for $i = 0$ *to numero_frames_video* **do**
if *tracks* == *VAZIO ou bboxes* == *VAZIO* **then**
validos = *bboxes[i]*[Pontuação] >= limite_Pontuação;
bboxes[i] = *bboxes[i]*[validos];
tracks = novasTrajetórias(*bboxes[i]*);
else
validos = nms(*bboxes[i], limiteSupressao = 0,1*);
bboxes_nms = *bboxes[i]*[validos];
tracks = combinarTrajetórias(*bboxes_nms, tracks*);
*bboxes*_validos = *bboxes[i]*[Pontuação] >= limite_Pontuação;
if *bboxes_validos != tracks* **then**
*bboxes*_nao_associados = *bboxes*_validos;
tracks = novasTrajetórias(*bboxes*_nao_associados);
end
end
fim

3.5 Métricas de avaliação

As métricas de avaliação retornam os resultados dos modelos e com isso possibilitam analisar se o modelo de fato está funcionando de maneira correta. Para a avaliação da detecção dos objetos utilizou-se a métrica chamada de *Average Precision* (AP) e a avaliação do rastreamento dos objetos é realizada através de uma métrica chamada HOTA em inglês *Higher Order Tracking Accuracy*.

3.5.1 Average Precision (AP)

A AP se baseia na Intersecção sobre a união (IoU), que é adquirida através da divisão entre a sobreposição entre as caixas delimitadoras e a união das caixas, essas caixas são: a caixa rotulada manualmente que representa a verdade fundamental e, a caixa delimitadora gerada pela previsão do modelo de detecção. Em suma, a AP define acertos como verdadeiro-positivo (TP) quando o valor IoU é maior do que um limite definido para cada experimento

e falso-positivo (FP) caso contrário. De acordo com o número de TP e FP, o AP é calculado como a área sob a curva de *Precision-Recall* e seu valor varia de 0 a 1. *Precision* e *Recall* são calculados de acordo com as Equações 3.1 e 3.2, respectivamente. *Precision* aponta, das classificações marcadas como certas, quantas estão realmente certas e *Recall* mostra, com que frequência o classificador está definindo objetos de uma certa classe. Se o objeto pertence a essa classe, o quão frequente essa classificação é recebida.

$$P = \frac{TP}{TP + FP} \tag{3.1}$$

$$R = \frac{TP}{TP + FN} \tag{3.2}$$

3.5.2 Higher Order Tracking Accuracy (HOTA)

O HOTA (Higher Order Tracking Accuracy) é uma nova métrica que permite avaliar o desempenho dos métodos MOT. Ela foi aplicada a este trabalho, pois está integrada a ferramenta *MMTracking*.

Quanto ao seu funcionamento a HOTA funciona combinando três pontuações de IoU. São elas: detecção, associação e localização. Cada uma delas é calculada usando o IoU e em seguida, as pontuações são combinadas para a pontuação final da HOTA [30].

- Localização: Como é possível ver na Figura 3.4 localização mensura a distância espacial entre a detecção prevista e a detecção real, detecção essa que vem da anotação dos dados. O cálculo do IoU da localização (Loc-IoU) acontece com a razão entre a intersecção das detecções e união das detecções. A Precisão geral da localização (LocA) é calculada, usando a média do Loc-IoU nas detecções existentes, como mostrado na Equação 3.3

- Detecção: A detecção basicamente mede a precisão da detecção entre o conjunto de detecções verdadeiras e previstas. Durante o calculo filtra-se o conjunto de detecções previstas e verdadeiras usando um limite de localização, por exemplo, Loc-IoU > 0,5. Acima de 0,5 significa que as detecções se correspondem. Os resultados das detecções que se correspondem são chamados de Verdadeiros Positivos (TP). As detecções previstas que não correspondem são falsos positivos (FP) e as detecções de verdadeiras que não correspondem são falsos negativos (FN). O cálculo do IoU da detecção (Det-IoU) se dá pela Equação 3.4. A estrutura da Det-IoU é parecida com a Loc-IoU. A diferença é que o Det-IoU mensura a distancia entre o conjunto de todas as detecções previstas e detecções verdadeiras

e o Loc-IoU mensura apenas uma detecção. Para mensurar a precisão geral da detecção (DetA) utiliza-se a Equação 3.4.

- Associação: A associação mede quão bem um rastreador vincula as detecções ao longo dos *frames* nas mesmas identidades (IDs), utilizando o conjunto de trajetórias de identidade verdadeiras. Mensura-se usando uma detecção prevista e uma verdadeira que são correspondentes e com isso mede a distância entre toda a trajetória. Para obter o valor da interseção entre duas trajetórias utiliza-se as correspondências Verdadeiras Positivas (TP), as associações que se correspondem são chamadas de associações verdadeiras Positivas (TPA). As associações Falsas Positivas (FPA) são as detecções que não obtiveram correspondência com uma trajetória ou que obtiveram correspondência com uma trajetória errada. Caso ainda reste detecções na trajetória verdadeira, chamam-se Associações Falsas Negativas (FNA). A Associação IoU (Ass-IoU) segue-se de maneira semelhante as anteriores, vê-se na Equação 3.5. Para calcular a precisão geral da associação (AssA) utiliza-se a Equação 3.6, nela calcula-se a média do Ass-IoU em todos os pares de detecções previstas e verdadeiras que se correspondem dentro do conjunto de dados.

Ao final do calculo de cada uma das três métricas (localização, detecção e associação) é necessário combinar as métricas obtidas, com o intuito de conseguir o resultado do desempenho geral do método de rastreamento avaliado. Como visto nas Equações anteriores, DetA e o AssA utilizaram para cálculo um limite definido em Loc-IoU. Usando isso como base calcula-se esse valor em uma série de diferentes limites definidos em α. Para cada valor do limite, calcula-se a pontuação final como a média geométrica das pontuações de detecção e associação. E ao integrar os diferentes limites de α, inclui-se o valor de precisão da localização na pontuação final [30].

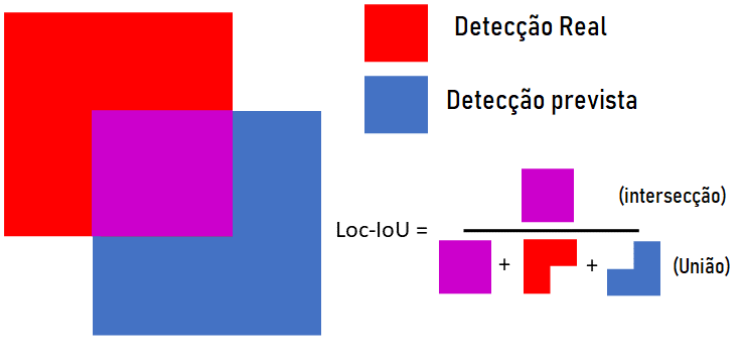

Figura 3.4: Cálculo localização.

$$LocA = \frac{1}{|TP|} \sum_{c \in TP} Loc\text{-}IoU(c) \tag{3.3}$$

$$DetA = Det\text{-}IoU = \frac{|TP|}{|TP| + |FN| + |FP|} \tag{3.4}$$

$$Ass\text{-}IoU = \frac{|TPA|}{|TPA| + |FNA| + |FPA|} \tag{3.5}$$

$$AssA = \frac{1}{|TP|} \sum_{c \in TP} Ass\text{-}IoU \tag{3.6}$$

$$HOTA_{\alpha} = \sqrt{DetA_{\alpha} \cdot AssA_{\alpha}} \tag{3.7}$$

$$HOTA = \int_{0<\alpha\leq 1} HOTA_{\alpha} \tag{3.8}$$

3.5.3 Avaliação das curvas de aprendizagem

Além da avaliação da precisão do modelo é interessante avaliar as curvas de aprendizagem, formadas com o valor da perda durante o treinamento dos modelos de detecções. Esses gráficos permitem diagnosticar o comportamento de um modelo e, possivelmente sugerir uma mudança de configuração necessária para melhorar o aprendizado.

Através das curvas de perda é existem três possíveis diagnósticos:

- *Underfit*;
- *Overfit*;
- *Good Fit*.

O *Underfit* acontece quando um modelo não consegue aprender os padrões do conjunto de dados de maneira adequada. Os gráficos de perda no caso do *Underfit* pode mostrar uma linha reta, ruidosa ou de valores muito altos, o que mostra que modelo não conseguiu aprender o com o conjunto de dados de treinamento.

O *Overfit* acontece quando um modelo aprendeu o conjunto de dados de treinamento muito bem, até mesmo os ruídos dele, isso faz com que quando aplicado a novos dados o modelo não trabalhe com precisão. Os gráficos de perda no caso do *Overfit* podem apresentar na curva do treinamento uma queda continua com mesmo com o passar do tempo. Outra possibilidade é o gráfico da validação diminuir a um ponto e após isso voltar a aumentar.

O *Good Fit* acontece quando o modelo está ajustado corretamente em relação ao conjunto de dados, ele pode ser identificado no gráfico por uma curva do treinamento e da validação que diminuem até um ponto, em que encontrem estabilidade com uma distancia pequena entre eles.

CAPÍTULO

4

Resultados

Neste capítulo encontram-se os resultados obtidos na detecção e rastreamento dos objetos. Possuindo duas base de dados diferentes que foram geradas através de dois vídeos, seguiu-se a abordagem de treinar o método de detecção na primeira base de dados chamada de GPBK0196 e utilizar a segunda base de dados chamada de GPBK0197 apenas para testes. A base de dados GPBK0196 foi dividida em duas partes treino e validação. A avaliação dos métodos de rastreamentos possuem os resultados para cada um dos cortes do vídeo e também contam com o resultado geral.

4.1 Detecção de objetos

Foram aplicados à base de dados, três diferentes métodos de detecção de objetos, *Faster R-CNN* [26], *RetinaNet* [27], *VFNet* [28]. A ideia é permitir avaliar os principais métodos disponíveis e seu desempenho na detecção dos formigueiros. Esses métodos foram selecionados pela possibilidade de acoplamento aos métodos de rastreamento utilizados.

Na Tabela 4.1, vemos os resultados dos modelos de detecção. Os resultados são divididos em dois. AP com o IoU de 50% e 75%. O modelo com o melhor resultado foi o *RetinaNet* apresentando melhores resultados quando testado na base de dados GPBK0197.

É possível visualizar nas Figuras 4.1, 4.2 e 4.3. As curvas de perdas, obtidas durante o treinamento dos modelos de detecção. Cada Figura apresenta um gráfico por epócas, este gráfico foi gerado com a ferramenta de *Log Analysis* do *MMTracking*. Com os gráficos é possível avaliar a qualidade do modelo de detecção, evitando um modelo super ajustado. Ao analisar os gráficos po-

demos notar que as curvas de treino e validação atingiram um ponto de estabilidade e apresentam uma pequena distância pequena entre eles, isso significa que os modelos estão ajustados corretamente para a detecção de formigueiros.

Método	Base de dados	AP IoU 50%	AP IoU 75%
Faster R-CNN	GPBK0197	0.817	0.181
RetinaNet	GPBK0197	0.834	0.207
VFNet	GPBK0197	0.789	0.213

Tabela 4.1: Avaliação dos métodos de detecção

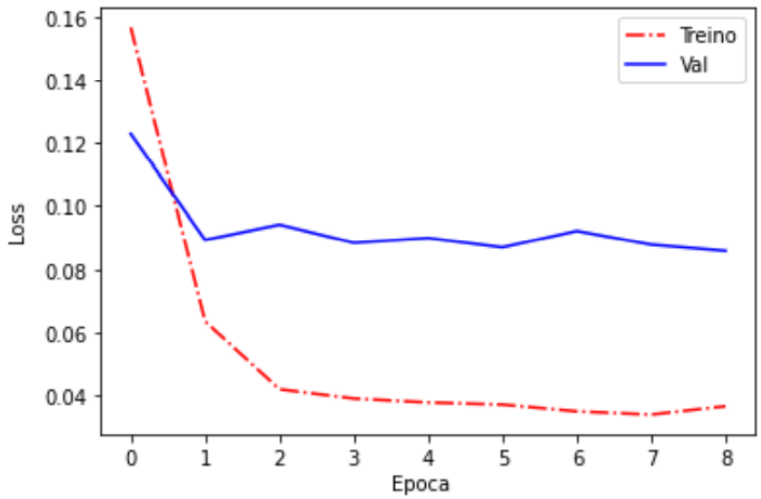

Figura 4.1: curva de aprendizagem *Faster R-CNN*.

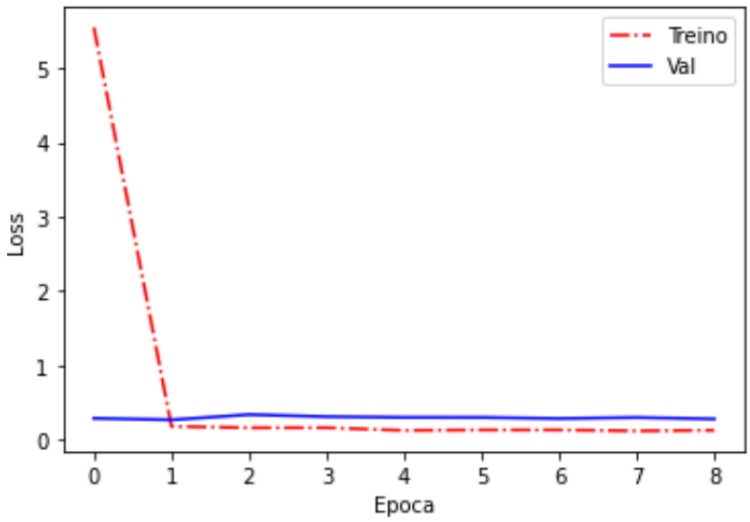

Figura 4.2: curva de aprendizagem *RetinaNet*.

4.2 Rastreamento de objetos

Com os modelos de detecção treinados, foram aplicados os métodos de rastreamento, *Deep SORT*, *Tracktor*, *Byte Track* e o Método Proposto. Inicial-

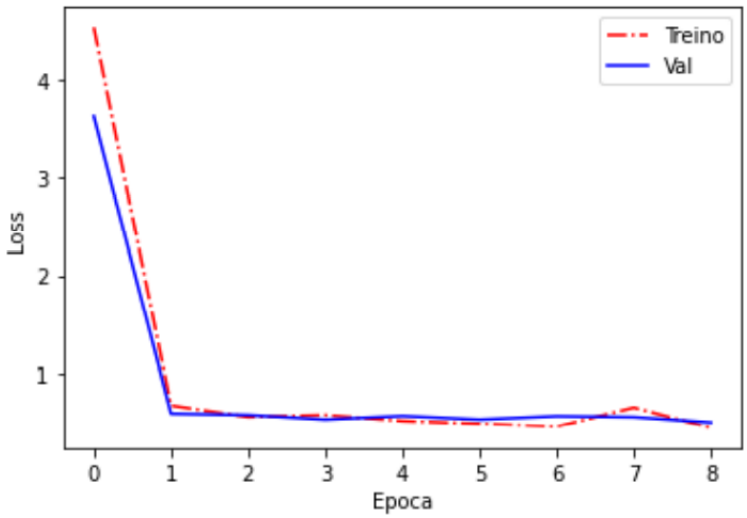

Figura 4.3: Curva de aprendizagem *VFNet*.

mente foram utilizados os parâmetros padrão para o rastreamento, entre os principais parâmetros, estão: O número de *frames* consecutivos para confirmar uma trajetória, o número de *frames* em que um objeto pode se manter desaparecido antes de ser excluído e o limiar de similaridade para associação de objetos. A ferramenta *MMTracking* disponibiliza uma forma de buscar os melhores parâmetros, nela são testadas diferentes valores e exibe-se os resultados. Usando essa ferramenta foi possível selecionar os melhores valores para os parâmetros com base nos resultados obtidos com cada método de rastreamento. A quantidade de *frames* contínuos selecionada para confirmar a trajetória três *frames* para todos os métodos de rastreamento. O número de *frames* em que um objeto pode se manter desaparecido, foi diferente para cada rastreador, o *Tracktor* obteve dez *frames*, o *Byte Tracker* trinta *frames*, o *Deep Sort* cem *frames* e o Método Proposto cinquenta *frames*. Durante a análise do melhor valor para o limiar de similaridade para associação de objetos, entre os valores analisados o percentual que alcançou os melhores resultados foi o valor de 30% para todos os rastreadores.

Os resultados foram divididos em Pontuações e Contagem. As pontuações, são: HOTA, DetA, AssA e LocA. As pontuações variam de 0 à 100, sendo 0 o pior resultado e 100 o melhor resultado. As Contagens possuem a quantidade de IDs (Indentificações), Dets (Detecções) geradas pelo método de rastreamento e também os valores de IDs e Dets da anotação, chamados verdade fundamental (GT). Para cada detector utilizado existe uma tabela com os resultados em cada um dos rastreadores aplicados, além disso existem os resultados gerais e resultados individuais de cada corte dos vídeos.

Na Tabela 4.2 estão as pontuações HOTA obtidas com o método de detecção *Faster R-CNN + ResNet101* e os rastreadores de objetos. Além da pontuação HOTA, temos as pontuações DetA, AssA e LocA, exibidas nas Tabelas 4.3, 4.4

e 4.5, com essas pontuações é possível analisar o desempenho dos métodos em contextos individuais. Nos resultados exibidos é possível visualizar que o método *Byte Tracker* obteve as melhores pontuações em comparação aos outros métodos de rastreamento, ainda assim os resultados não variam muito, se mantendo bem próximos. Outros resultados importantes são as contagens de IDs e Detecções, estes resultados são exibidos nas Tabelas 4.7 e 4.6. Existem dois pontos interessantes nas Tabelas 4.7 e 4.6. Primeiro, apesar do *Byte Tracker* obter pontuações superiores a contagem apresentou um cenário totalmente diferente com uma quantidade de IDs muito superior aos valores da anotação. Segundo, o Método Proposto alcançou a contagem mais próxima da anotação e com pontuações HOTA um pouco abaixo do método *Byte Tracker*. A relação entre pontuação e contagem varia pois a avaliação não considera apenas a contagem de objetos e sim a detecção, localização e associação do objeto no vídeo com o objeto na anotação fazendo com que até mesmo os rastreadores que geraram muitos IDs, em algum momento detectaram e rastrearam o objeto de acordo com a anotação.

Corte	Tracktor	Byte Tracker	Deep Sort	Método Proposto
GPBK0197_Corte1	45.126	47.339	42.210	42.295
GPBK0197_Corte2	31.899	33.848	28.099	28.584
GPBK0197_Corte3	46.798	49.246	50.918	39.001
GPBK0197_Corte4	40.567	42.950	41.583	40.254
GPBK0197_Corte5	42.938	56.226	43.827	55.237
Geral GPBK0197	41.447	44.241	40.958	38.636

Tabela 4.2: Resultados HOTA: Detector *Faster R-CNN + ResNet101*

Corte	Tracktor	Byte Tracker	Deep Sort	Método Proposto
GPBK0197_Corte1	41.698	43.085	42.843	37.641
GPBK0197_Corte2	25.324	27.598	26.061	20.542
GPBK0197_Corte3	37.270	41.250	39.402	31.585
GPBK0197_Corte4	30.589	32.596	31.577	29.356
GPBK0197_Corte5	48.065	50.021	49.112	48.948
Geral GPBK0197	33.911	36.164	35.050	29.932

Tabela 4.3: Resultados DetA: Detector *Faster R-CNN + ResNet101*

Corte	Tracktor	Byte Tracker	Deep Sort	Método Proposto
GPBK0197_Corte1	48.847	52.021	41.633	47.535
GPBK0197_Corte2	40.245	41.621	30.326	39.931
GPBK0197_Corte3	58.776	58.801	65.822	48.166
GPBK0197_Corte4	53.816	56.607	54.773	55.214
GPBK0197_Corte5	38.608	63.208	39.388	62.342
Geral GPBK0197	50.719	54.192	48.008	49.933

Tabela 4.4: Resultados AssA: Detector *Faster R-CNN* + *ResNet101*

Corte	Tracktor	Byte Tracker	Deep Sort	Método Proposto
GPBK0197_Corte1	76.311	77.260	77.284	77.186
GPBK0197_Corte2	70.677	71.515	71.518	71.499
GPBK0197_Corte3	78.270	79.755	79.755	79.755
GPBK0197_Corte4	79.505	80.254	80.239	80.254
GPBK0197_Corte5	80.312	80.571	81.002	80.571
Geral GPBK0197	76.238	77.118	77.165	77.068

Tabela 4.5: Resultados LocA: Detector *Faster R-CNN* + *ResNet101*

Corte	GT_Dets	Tracktor	Byte Tracker	Deep Sort	Método Proposto
GPBK0197_Corte1	356	541	536	537	638
GPBK0197_Corte2	251	526	511	523	715
GPBK0197_Corte3	195	379	353	370	464
GPBK0197_Corte4	235	549	538	542	599
GPBK0197_Corte5	66	80	81	80	83
Geral GPBK0197	1103	2075	2019	2052	2499

Tabela 4.6: Contagem de Dets: Detector *Faster R-CNN* + *ResNet101*

Corte	GT_IDs	Tracktor	Byte Tracker	Deep Sort	Método Proposto
GPBK0197_Corte1	2	18	25	26	6
GPBK0197_Corte2	2	22	20	38	9
GPBK0197_Corte3	1	19	18	27	6
GPBK0197_Corte4	1	12	21	23	6
GPBK0197_Corte5	1	4	3	4	3
Geral GPBK0197	7	75	87	118	30

Tabela 4.7: Contagem de IDs: Detector *Faster R-CNN* + *ResNet101*

Diferente do *Faster R-CNN* o método *RetinaNet* + *ResNet101*, não permite a implementação do método *Tracktor*, pois necessita dos recursos de região de interesse que apenas o *Faster R-CNN* possui, por este motivo não foi possível avaliá-lo com outros detectores. Analisando os resultados da Tabela 4.8

vê-se que os métodos *Byte Tracker* e Método Proposto obtiveram os melhores resultados quando avaliados. Outro ponto interessante é o aumento na pontuação em comparação com os resultados do detector *Faster R-CNN*, em concordância a avaliação do método de detecção na Tabela 4.1, com isso podemos notar que o desempenho do método de rastreamento está atrelado ao desempenho do detector de objetos. Na contagem de IDs exibida na Tabela 4.13 é possível notar a diminuição da contagem de IDs em comparação com os resultados do detector *Faster R-CNN* juntamente com o bom desempenho do Método Proposto, apresentando a menor contagem de objetos dentre os outros métodos.

Corte	Tracktor	Byte Tracker	Deep Sort	Método Proposto
GPBK0197_Corte1	-	53.004	43.426	47.120
GPBK0197_Corte2	-	42.681	36.010	39.688
GPBK0197_Corte3	-	64.141	49.733	47.032
GPBK0197_Corte4	-	53.705	35.421	47.589
GPBK0197_Corte5	-	56.567	44.574	51.320
Geral GPBK0197	-	53.653	41.599	46.400

Tabela 4.8: Resultados HOTA: Detector *RetinaNet* + *ResNet101*

Corte	Tracktor	Byte Tracker	Deep Sort	Método Proposto
GPBK0197_Corte1	-	48.062	42.273	41.689
GPBK0197_Corte2	-	42.671	31.546	31.948
GPBK0197_Corte3	-	63.318	42.557	38.2070
GPBK0197_Corte4	-	40.552	33.038	34.886
GPBK0197_Corte5	-	48.460	49.831	41.417
Geral GPBK0197	-	46.675	37.468	36.904

Tabela 4.9: Resultados DetA: Detector *RetinaNet* + *ResNet101*

Corte	Tracktor	Byte Tracker	Deep Sort	Método Proposto
GPBK0197_Corte1	-	58.462	44.648	53.268
GPBK0197_Corte2	-	42.694	41.123	49.316
GPBK0197_Corte3	-	64.975	58.119	57.906
GPBK0197_Corte4	-	71.156	38.063	64.948
GPBK0197_Corte5	-	66.039	40.013	63.608
Geral GPBK0197	-	61.915	46.483	58.508

Tabela 4.10: Resultados AssA: Detector *RetinaNet* + *ResNet101*

Corte	Tracktor	Byte Tracker	Deep Sort	Método Proposto
GPBK0197_Corte1	-	76.460	76.557	76.336
GPBK0197_Corte2	-	71.417	71.353	71.298
GPBK0197_Corte3	-	79.353	79.353	79.353
GPBK0197_Corte4	-	81.798	81.789	81.798
GPBK0197_Corte5	-	81.423	81.389	80.782
Geral GPBK0197	-	77.338	77.326	77.172

Tabela 4.11: Resultados LocA: Detector *RetinaNet* + *ResNet101*

Corte	GT_Dets	Tracktor	Byte Tracker	Deep Sort	Método Proposto
GPBK0197_Corte1	356	-	490	544	586
GPBK0197_Corte2	251	-	293	421	439
GPBK0197_Corte3	195	-	229	421	384
GPBK0197_Corte4	235	-	449	547	524
GPBK0197_Corte5	66	-	89	86	111
Geral GPBK0197	1103	-	1550	1942	2044

Tabela 4.12: Contagem de Dets: Detector *RetinaNet* + *ResNet101*

Corte	GT_IDs	Tracktor	Byte Tracker	Deep Sort	Método Proposto
GPBK0197_Corte1	2	-	7	19	3
GPBK0197_Corte2	2	-	7	44	6
GPBK0197_Corte3	1	-	5	25	3
GPBK0197_Corte4	1	-	8	21	4
GPBK0197_Corte5	1	-	4	5	3
Geral GPBK0197	7	-	31	114	19

Tabela 4.13: Contagem de IDs: Detector *RetinaNet* + *ResNet101*

Por último temos o *VFNet* + *ResNet101*. Na Tabela 4.14 vê-se os resultados HOTA obtidos para os diferentes métodos de rastreamento, visualizando as diferentes pontuações é possível notar que os resultados estão bem próximos aos resultados do detector *Faster R-CNN*. Novamente podemos notar que o Método Proposto se destaca dos outros métodos de rastreamento apresentando uma contagem menor, como visto na Tabela 4.19.

Corte	Tracktor	Byte Tracker	Deep Sort	Método Proposto
GPBK0197_Corte1	-	46.582	37.090	39.098
GPBK0197_Corte2	-	36.663	24.865	28.769
GPBK0197_Corte3	-	45.177	39.889	29.430
GPBK0197_Corte4	-	45.518	33.877	35.977
GPBK0197_Corte5	-	54.137	40.775	31.475
Geral GPBK0197	-	44.730	34.686	33.698

Tabela 4.14: Resultados HOTA: Detector *VFNet* + *ResNet101*

Corte	Tracktor	Byte Tracker	Deep Sort	Método Proposto
GPBK0197_Corte1	-	41.454	35.896	34.816
GPBK0197_Corte2	-	30.740	23.077	20.208
GPBK0197_Corte3	-	36.737	29.389	18.086
GPBK0197_Corte4	-	33.161	31.022	22.318
GPBK0197_Corte5	-	46.137	43.626	16.470
Geral GPBK0197	-	35.990	30.316	22.892

Tabela 4.15: Resultados DetA: Detector *VFNet* + *ResNet101*

Corte	Tracktor	Byte Tracker	Deep Sort	Método Proposto
GPBK0197_Corte1	-	52.405	38.388	44.038
GPBK0197_Corte2	-	43.748	26.817	41.069
GPBK0197_Corte3	-	55.562	54.153	47.906
GPBK0197_Corte4	-	62.502	37.009	58.034
GPBK0197_Corte5	-	63.536	38.164	60.233
Geral GPBK0197	-	55.730	39.852	49.716

Tabela 4.16: Resultados AssA: Detector *VFNet* + *ResNet101*

Corte	Tracktor	Byte Tracker	Deep Sort	Método Proposto
GPBK0197_Corte1	-	76.165	76.327	76.156
GPBK0197_Corte2	-	71.950	71.963	71.870
GPBK0197_Corte3	-	79.300	79.300	79.300
GPBK0197_Corte4	-	81.889	81.889	81.889
GPBK0197_Corte5	-	79.058	79.850	78.863
Geral GPBK0197	-	77.181	77.294	77.114

Tabela 4.17: Resultados LocA: Detector *VFNet* + *ResNet101*

Corte	GT_Dets	Tracktor	Byte Tracker	Deep Sort	Método Proposto
GPBK0197_Corte1	356	-	585	637	701
GPBK0197_Corte2	251	-	451	609	734
GPBK0197_Corte3	195	-	401	503	822
GPBK0197_Corte4	235	-	554	593	829
GPBK0197_Corte5	66	-	90	90	271
Geral GPBK0197	1103	-	2081	2432	3357

Tabela 4.18: Contagem de Dets: Detector *VFNet* + *ResNet101*

Corte	GT_IDs	Tracktor	Byte Tracker	Deep Sort	Método Proposto
GPBK0197_Corte1	2	-	11	22	6
GPBK0197_Corte2	2	-	38	54	6
GPBK0197_Corte3	1	-	13	36	4
GPBK0197_Corte4	1	-	22	30	7
GPBK0197_Corte5	1	-	8	12	4
Geral GPBK0197	7	-	92	154	27

Tabela 4.19: Contagem de IDs: Detector *VFNet* + *ResNet101*

Observando os resultados de contagem é possível visualizar que o número de IDs contados em alguns casos é alto, em relação a GT. Isso pode ocorrer por dois motivos, primeiro motivo, durante o processo de rastreamento caso o método pode contar de forma errada um objeto mais de uma vez, criando uma nova identidade, esse fenômeno é considerado um Falso Positivo. Segundo motivo, formigueiros que não foram incluídos na anotação, por estarem muito distantes ou serem muito pequenos, são detectados com uma pontuação alta e acabam sendo rastreados, porém fazem a contagem superar o valor da anotação, esse fenômeno é considerado um Falso Negativo. O número alto de objetos contados aparecem em todos os métodos de rastreamento, em alguns casos muito altos e em outros os valores chegam mais próximos do valor GT. Com o intuito de diferenciar IDs Falsos Positivos, de IDs Falsos Negativos temos as Figuras 4.4 e 4.5, através delas é possível visualizar a quantidade de IDs por *frames* do vídeo. Como exibido na Figura 4.4, o rastreamento cria novos objetos que não estão próximos ao GT, uma forma de corrigir o problema é reavaliar os objetos que devem ser incluídos na anotação, ou filtrar os resultados da detecção. Na Figura 4.5 vê-se que não existem muitos IDs gerados longe dos GTs, isso pode significar que o rastreador gerou diferentes identidades para os mesmos objetos causando uma contagem alta de objetos. Ao analisar os vídeos gerados com o rastreamento de objetos utilizando o Método Proposto percebe-se um cenário diferente, a contagem de objetos é baixo e a mudança de IDs entre os objetos é muito pequena, efetuando a contagem de

forma satisfatória.

Por fim, podemos observar que com um modelo de detecção bem treinado os resultados obtidos são bons em muitos dos casos, porém o método de rastreamento também influencia muito no resultado final, sendo possível observar que em vários casos os resultados variam em grande quantidade. Analisando os resultados obtidos é possível observar que o Método Proposto trás alguns benefícios, como, simplicidade de implementação, diferente de alguns rastreadores o Método Proposto não utiliza uma rede de re-identificação, além disso os métodos mais robustos como *Tracktor*, *Byte Tracker* e *Deep Sort*, muita vezes obtiveram resultados inferiores ao Método Proposto.

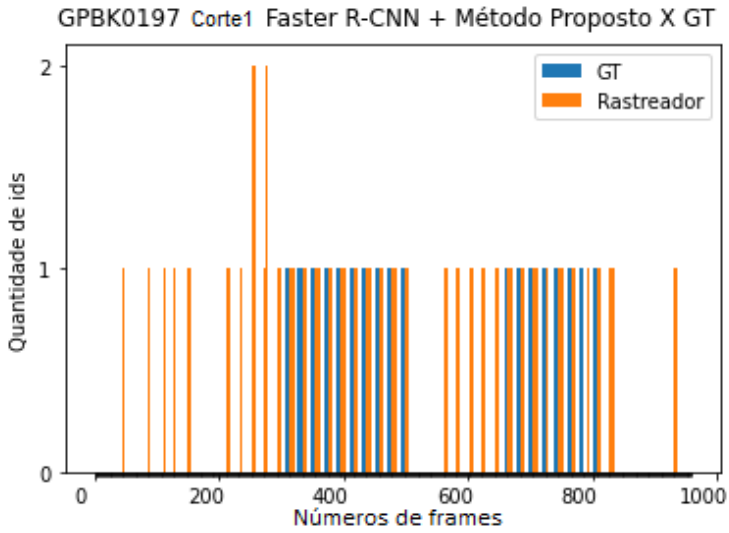

Figura 4.4: GPBK0197 Corte1 Faster R-CNN + Método Proposto X GT.

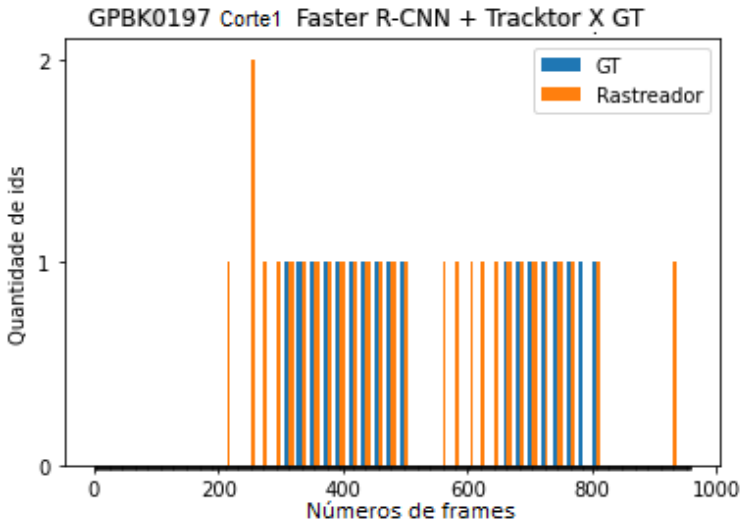

Figura 4.5: GPBK0197 Corte1 Faster R-CNN + *Tracktor* X GT.

CAPÍTULO

5

Conclusão

5.1 Conclusão

Dado o exposto, pôde-se perceber que a abordagem do modelo proposto apresentou bons resultados aplicado ao problema de rastreamento de formigueiro, em comparação com os outros métodos de rastreamento.

Ainda que possua resultados interessantes o rastreamento de objetos ainda necessita de mais atenção e estudos, visto que muitos dos problemas em que aplica-se a técnica de rastreamento é voltada para a aplicação em pedestres. Outro ponto importante é a dificuldade de informações em relação a anotação dos objetos, existem poucos exemplos de formato da base de dados para aplicação em rastreamento.

Por fim, acredita-se que este trabalho incentive e estimule novas pesquisas a respeito da detecção e rastreamento de objetos.

5.2 Trabalhos Futuros

Como trabalhos futuros propõe-se a implementação da desativação da trajetória a partir do momento em que o objeto sai do campo de visão pelas laterais da imagem, tendo em vista que o objeto só retorna ao campo de visão em casos de oclusão pelas árvores.

Outra proposta seria a implementação do detector de objetos *YOLOX* juntamente ao método *Byte Tracker* afim de avaliar a configuração sugerida para o rastreador *Byte Tracker*.

Outra proposta seria a aplicação de novos métodos de detecção, junta-

mente com o Método Proposto afim de melhorar a contagem dos objetos.

E por fim outra proposta seria buscar as melhores formas de avaliar os métodos MOT para o problema de rastreamento de formigueiros.

Referências Bibliográficas

[1] Instituto Brasileiro de Geografia e Estatística - IBGE. "Levantamento Sistemático da Produção Agrícola - LSPA"ago. 2020. Disponível em: https://biblioteca.ibge.gov.br/visualizacao/periodicos/2415/epag_2020_ago.pdf Acesso em: 20 set. 2020. Citado na página 2.

[2] IBÁ - Indústria Brasileira de Árvores. Dados estatísticos. Disponível em: https://www.iba.org/dados-estatisticos. Acesso em: 25 set. 2020. Citado na página 2.

[3] IBGE - Instituto Brasileiro de Geografia e Estatística. Produção da extração vegetal e da silvicultura - PEVS. Disponível em:<https://sidra.ibge.gov.br/pesquisa/pevs/tabelas> Acesso em: 04 set. 2022. Citado na página 2.

[4] OLIVEIRA A. D., Barcelos, J. A. V., Moraes, E. D., & Freitas, G. D. Um estudo de caso: o sistema de monitoramento e controle de formigas cortadeiras na Mannesmann Fi-EL Florestal Ltda. As formigas cortadeiras, 1993. Citado na página 2.

[5] ZANETTI, R., Zanuncio, J. C., Mayhé-Nunes, A. J., Medeiros, A. G. B., & Souza-Silva, A. Combate sistemático de formigas-cortadeiras com iscas granuladas, em eucaliptais com cultivo mínimo. Revista Árvore, v. 27, n. 3, p. 387-392, 2003. Citado na página 2.

[6] REIS, M. D. A., Zanetti, R., Scolforo, J. R. S., & Ferreira, M. Z. Amostragem de formigas-cortadeiras (Hymenoptera: Formicidae) em eucaliptais pelos métodos de transectos em faixa e em linha. Revista Árvore, v. 34, n. 6, p. 1101-1108, 2010. Citado na página 2.

[7] STAFFORD, John V. Implementing precision agriculture in the 21st century. Journal of Agricultural Engineering Research, v. 76, n. 3, p. 267-275, 2000. Citado na página 3.

[8] BROSNAN, Tadhg; SUN, Da-Wen. Improving quality inspection of food products by computer vision a review. Journal of food engineering, v. 61, n. 1, p. 3-16, 2004. Citado nas páginas 3 e 10.

[9] (GONZALEZ & WOODS, 2000) GONZALEZ, R. C.; WOODS, R. E. Processamento de imagens digitais. São Paulo: Edgard Blucher, 2000. Citado na página 6.

[10] SOVIANY, Petru; IONESCU, Radu Tudor. Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction. In: 2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC). IEEE, 2018. p. 209-214. Citado na página 7.

[11] LUO, W., Xing, J., Milan, A., Zhang, X., Liu, W., & Kim, T. K. Multiple object tracking: A literature review. Artificial Intelligence, p. 103448, 2020. Citado na página 7.

[12] BRAGA, A. P., Carvalho, A. P. L. F., & Ludemir, T. B. Redes neurais artificiais: teoria e aplicação. Rio de Janeiro: LTC, 2007. Citado na página 8.

[13] KHAN, S., Rahmani, H., Shah, S. A. A., & Bennamoun, M. A guide to convolutional neural networks for computer vision. Synthesis Lectures on Computer Vision, v. 8, n. 1, p. 1-207, 2018 Citado na página 10.

[14] PATRÍCIO, Diego Inácio; RIEDER, Rafael. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. Computers and electronics in agriculture, v. 153, p. 69-81, 2018. Citado na página 11.

[15] LI, Z., Li, Y., Yang, Y., Guo, R., Yang, J., Yue, J., & Wang, Y. A high-precision detection method of hydroponic lettuce seedlings status based on improved Faster RCNN. Computers and Electronics in Agriculture, v. 182, p. 106054, 2021. Citado na página 11.

[16] AMPATZIDIS, Yiannis; PARTEL, Victor; COSTA, Lucas. Agroview: Cloud-based application to process, analyze and visualize UAV-collected data for precision agriculture applications utilizing artificial intelligence. Computers and Electronics in Agriculture, v. 174, p. 105457, 2020. Citado na página 12.

[17] HAMIDISEPEHR, Ali; MIRNEZAMI, Seyed Vahid; WARD, Jason K. Comparison of Object Detection Methods for Corn Damage Assessment using Deep Learning. Transactions of the ASABE, p. 0, 2020. Citado na página 12.

[18] BAYRAKTAR, Ertugrul; BASARKAN, Muhammed Enes; CELEBI, Numan. A low-cost UAV framework towards ornamental plant detection and counting in the wild. ISPRS Journal of Photogrammetry and Remote Sensing, v. 167, p. 1-11, 2020. Citado na página 13.

[19] Pratama, M. T., Kim, S., Ozawa, S., Ohkawa, T., Chona, Y., Tsuji, H., & Murakami, N. Deep Learning-based Object Detection for Crop Monitoring in Soybean Fields. In: 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020. p. 1-7. Citado na página 13.

[20] Kapania, S., Saini, D., Goyal, S., Thakur, N., Jain, R., & Nagrath, P. Multi object tracking with UAVs using deep SORT and YOLOv3 RetinaNet detection framework. In: Proceedings of the 1st ACM Workshop on Autonomous and Intelligent Mobile Systems. 2020. p. 1-6. Citado na página 14.

[21] OSMAN, Youssef; DENNIS, Reed; ELGAZZAR, Khalid. Yield Estimation and Visualization Solution for Precision Agriculture. Sensors, v. 21, n. 19, p. 6657, 2021. Citado na página 14.

[22] CONTRIBUTORS, M. MMTracking: OpenMMLab video perception toolbox and benchmark. 2020. Citado na página 19.

[23] Ren, S., He, K., Girshick, R., & Sun, J. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence, v. 39, n. 6, p. 1137-1149, 2016. Citado nas páginas 18 e 19.

[24] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. 2017. p. 2980-2988. Citado nas páginas 18 e 19.

[25] Ahmed, A., Tangri, P., Panda, A., Ramani, D., & Karmakar, S. VFNet: A Convolutional Architecture for Accent Classification. In: 2019 IEEE 16th India Council International Conference (INDICON). IEEE, 2019. p. 1-4. Citado nas páginas 18 e 20.

[26] Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. Simple online and realtime tracking. In: 2016 IEEE international conference on image processing (ICIP). IEEE, 2016. p. 3464-3468. Citado nas páginas 21 e 29.

[27] WOJKE, Nicolai; BEWLEY, Alex; PAULUS, Dietrich. Simple online and realtime tracking with a deep association metric. In: 2017 IEEE international conference on image processing (ICIP). IEEE, 2017. p. 3645-3649. Citado nas páginas 22 e 29.

[28] BERGMANN, Philipp; MEINHARDT, Tim; LEAL-TAIXE, Laura. Tracking without bells and whistles. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019. p. 941-951. Citado nas páginas 22 e 29.

[29] Zhang, Y., Sun, P., Jiang, Y., Yu, D., Yuan, Z., Luo, P., ... & Wang, X. Bytetrack: Multi-object tracking by associating every detection box. arXiv preprint arXiv:2110.06864, 2021. Citado na página 23.

[30] Luiten, J., Osep, A., Dendorfer, P., Torr, P., Geiger, A., Leal-Taixé, L., & Leibe, B. Hota: A higher order metric for evaluating multi-object tracking. International journal of computer vision, v. 129, n. 2, p. 548-578, 2021. Citado nas páginas 26 e 27.

[31] Rezatofighi, Hamid and Tsoi, Nathan and Gwak, JunYoung and Sadeghian, Amir and Reid, Ian and Savarese, Silvio, Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019 pp. 658-666. Citado na página 24.