



UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL

TALIS ULISSES DA SILVA

**DETECÇÃO DE CUPINZEIROS EM PLANTAÇÃO DE EUCALIPTO VIA
MÉTODOS DE DETECÇÃO DE OBJETOS EM IMAGENS**

PONTA PORÃ – MS

2023



DETECÇÃO DE CUPINZEIROS EM PLANTAÇÃO DE EUCALIPTO VIA MÉTODOS DE DETECÇÃO DE OBJETOS EM IMAGENS

Dissertação apresentada à Universidade Federal de Mato Grosso do Sul no campus de Ponta Porã, como requisito para a obtenção do bacharel em sistemas de informação.

Orientador: Prof. Dr. Jonathan Andrade Silva

PONTA PORÃ – MS

2023

LISTA DE FIGURAS

Figura 1 – Vídeo gravado usando a lente “olho de peixe”	13
Figura 2 - Anotação do <i>Roboflow</i>	14
Figura 3 – Posicionamento dos cupinzeiros e proposta de detecção	15
Figura 4 - Exemplo de overfitting	16
Figura 5 - Gráfico do algoritmo DETR com o <i>learning rate</i> a 0.002	18
Figura 6 - Exemplo de imagem com um número alto de <i>False Positive</i>	19
Figura 7 - Exemplo de <i>True Positive</i> (TP)	20
Figura 8 - Exemplo de <i>False Positive</i> (FP)	20
Figura 9 - Imagem do Retinanet r101.	22
Figura 10 - Imagens do algoritmo AutoAssign	25
Figura 11 - Imagens do algoritmo TOOD	25

LISTA DE QUADROS

Quadro 1 – Algoritmos utilizados – Época do <i>Early Stop</i>	16
Quadro 2 - Algoritmos com o <i>lr</i> utilizado	17
Quadro 3 - Quadros dos algoritmos e sua precisão de acordo com o IoU 0.50	22
Quadro 4 – Quadro dos algoritmos e sua precisão de acordo com o IoU 0.75	23
Quadro 5 – Quadro com a taxa de Recall dos algoritmos	24

RESUMO

O objetivo do trabalho é procurar dentre alguns algoritmos selecionados, qual obtém o melhor desempenho na detecção de cupinzeiros em uma plantação de eucalipto, utilizando o *Average Precision* (AP) e recall, tentando evitar que o mesmo sofra *overfitting*, para que possa ser usado em outros datasets do mesmo objeto.

O conjunto de imagens adequadamente preparadas para o treinamento de uma IA, é possível detectar uma diversidade de objetos ao mesmo tempo. Dando a um algoritmo uma imagem é esperado que o mesmo nos devolva a mesma imagem com um ou vários possíveis resultados para o objeto procurado, que pode vir a ser identificado mesmo que com algum tipo de obstrução visual. Como a detecção de objetos vem sendo cada vez mais utilizados por empresas dos mais variados setores nos últimos anos, os algoritmos recebendo atualizações e variantes frequentemente e se tornado muito acessíveis, com essa acessibilidade que temos atualmente, a proposta é treinar alguns algoritmos novos e antigos, com o objetivo de encontrar um algoritmo que consiga se adequar a um ambiente de fazenda e sendo utilizado para identificar cupinzeiros, observando o desempenho e tentando entender quais características dos algoritmos o levam a ter um melhor resultado no ambiente proposto.

Palavras - Chave: Algoritmos; *Average Precision*; *Overfitting*; Imagens; Detecção de Objetos.

ABSTRACT

The objective of the work is to look among some selected algorithms, which obtains the best performance in detecting termites in a eucalyptus plantation, using average precision (AP) and recall, trying to prevent it from being overfit, so that it can be used in other datasets of the same object.

Object detection has achieved incredible performances in recent years, with the emergence of new detection algorithms and methods, it has become increasingly easy to training a machine to identify objects. With the set of Images properly prepared for the training of an AI, you can detect a diversity of objects at the same time. Giving an algorithm an image is expected to return the same image to us with one or more possible results for the searched object, which can be identified even if with some kind of visual obstruction. With object detection being used by a lot of companies from different areas, algorithms being updated and getting new variations and becoming more accessible, the proposal is to train new and old algorithms, with the objective to find an algorithm that can adapt to a farm ambient and find termites, analyze their results and try to find which characteristics makes an algorithm to perform better.

Keywords: Algorithms; Average Precision; Overfitting; Images; Object Detection.

SUMÁRIO

INTRODUÇÃO	8
METODOLOGIA	13
RESULTADOS E DISCUSSÃO	22
CONCLUSÃO	26
REFERÊNCIAS	27

1 INTRODUÇÃO

As áreas florestais são manejadas intensivamente e uma ou duas espécies da mesma idade periodicamente são plantadas para fins de produção (FAO, 2020). Dentre as mais utilizadas, destacam-se os plantios de eucalipto (*Eucalyptus spp.*), que já existem em todos os continentes e apresentam a maior produtividade (Li et al., 2018). No Brasil, esse gênero ocupa 5,5 milhões de hectares e 73 % da área de florestas colocadas do país, principalmente em Minas Gerais, São Paulo, Mato Grosso do Sul e Bahia (IBÁ, 2019).

O eucalipto pertence a uma grande família de plantas chamada *Myrtaceae*. Esta família inclui atualmente mais de 5000 espécies (cerca de 800 delas são eucaliptos), distribuídas principalmente em regiões tropicais ou subtropicais do Hemisfério Sul. Particularmente na Austrália as *Myrtaceae* dominam a flora existente com 75 gêneros (o mais conhecido é o *Eucalyptus*) e mais de 1500 espécies, a maioria das quais endêmicas (MOÇO,2019).

O eucalipto tem grande importância comercial na economia brasileira. Segundo a indústria madeireira brasileira, esse gênero está plantado em 5,5 milhões de hectares, com produtividade média de 39 m³/ha / ano. No entanto, a produtividade depende de vários fatores, como local de plantio, tratos culturais e insumos disponíveis. Em geral, as espécies de eucalipto têm sido favorecidas para os diversos usos de sua madeira devido ao seu veloz crescimento, adaptabilidade a diferentes regiões ecológicas e potencial econômico (PENTEADO; PICHIELLI; SOARES,2019).

A alta produtividade da árvore, os menores custos e o maior retorno do investimento tornam o cultivo do eucalipto extremamente atrativo, garantindo alta competitividade de seus produtos no mercado interno e externo (PENTEADO; PICHIELLI; SOARES,2019).

De acordo com Ferreira, Oliveira e Lavor Lopes (2017), o eucalipto é considerado uma planta regeneradora do solo. Por ter raízes profundos, busca nutrientes minerais nas camadas mais baixas do solo, inacessíveis às raízes superficiais. Conseqüentemente, o eucalipto pode controlar a erosão do solo, ocupar áreas impróprios para a agricultura e, a longo prazo, reconstruir as reservas de água subterrânea do solo.

Incluiu matérias-primas para diversos fins como carpinteiro, apicultura (FERREIRA; OLIVEIRA; LAVOR LOPES,2017).

De acordo ainda com os autores acima, o plantio de eucalipto gera uma gama de produtos, que vêm diretamente sem transformação como a lenha na matriz energética de uso, a madeira serrada transformada mecanicamente é transformada em compensado, aglomerado, tábuas, chapas, chapas e caules aplicados na fabricação de móveis, embalagens; pode ser utilizado estruturalmente na construção civil, construção de moradias, redes elétricas e telefônicas, postes, em tratamentos termoquímicos na fabricação de celulose e derivados. E na produção de óleos para perfumaria, cosméticos e farmacêuticos. Além do mercado de crédito de carbono, os hectares cultivados com eucalipto capturaram gás carbônico do meio ambiente (FERREIRA; OLIVEIRA; LAVOR LOPES,2017).

Os cupins foram reclassificados e, de acordo com Beccaloni e Eggleton (2013), pertencem à ordem *Blattodea*, portanto não pertencem mais à antiga ordem *Isoptera*. Juntamente com abelhas, formigas e vespas, formam grupos de insetos *eusociais*, ou seja, figuram gerações sobrepostas, dividindo o trabalho e zelando da prole (CONSTANTINO, 1999; EGGLETON, 2010).

Embora os cupins sejam conhecidos pelos danos que causam, seja em áreas urbanas (prédios e móveis) ou rurais (perdas econômicas de safras), apenas 12,4 % são considerados destrutivos pela literatura. Os cupins tornam-se pragas quando seu *habitat* natural é alterado ou gradativamente substituído por áreas urbanas ou agrícolas, criando ambientes favoráveis ao assentamento (CHAGAS, 2016).

No Brasil, as espécies de cupins causam danos em áreas urbanas ou rurais, mas principalmente de forma indireta. As culturas de arroz, soja, milho, café, aipim, cana-de-açúcar e eucalipto têm documentado ataques de cupins, principalmente dos gêneros *Nasutitermes*, *Coptotermes*, *Heterotermes*, *Syntermes*, *Neocrapitermes*, *Cornitermes* e *Procornitermes* (CHAGAS, 2016).

Apesar de toda a mudança paisagística nos plantios de eucalipto, é possível observar a biodiversidade nessas áreas, principalmente quando há um remanescente de mata nativa próximo. Com base na conectividade da floresta plantada com os remanescentes, verifica-se a semelhança, em termos de fartura de espécies, com

florestas em regeneração (CHAGAS, 2016).

Com relação aos cupins a diversidade e riqueza de espécies dos plantios de eucalipto é importante, pois espécies de todos os grupos funcionais habitam esses ambientes. No entanto, a prevalência de carnívoros é maior e isso ocorre devido a distúrbios na vegetação antes do plantio principalmente no solo, o que causa diminuição de cupins humívoros (CHAGAS, 2016).

Na cultura de pastagem esta relação é contrária, onde a abundância e riqueza de cupins humívoros são maiores do que os xilófagos, pois a presença de madeira é baixa neste tipo de cultura. No que diz respeito a danos a cultura do eucalipto, espécies como *Anoplotermes pacificus*, *Neocapritermes opacus*, *Subulitermes sp.*, *Coptotermes testaceus*, *Cylindrotermes sp.*, *Cornitermes cumulans*, *Procornitermes araujo*, *Syntermes insidians* e *S. molestus* foram descritas como as principais pragas (CHAGAS, 2016).

Uma das alternativas para combater os cupins no campo tem sido tratar as plantas em viveiro, antes de encaminhá-las para o plantio por imersão em solução contendo um dos produtos feitos especificamente para o combate de cupins. A maioria dos cupins compreende a comer madeira morta (SERRA et. al.,2019).

Em áreas com cupins de solo (*Syntermes sp.*) a proteção é possível com produtos à base de *carbosulfan*, *fipronil* ou *clopirifós*. São produtos de baixa resistência que podem prevenir ataques de cupins e possivelmente outros insetos como *tripes*, pulgões, cigarrinhas, etc., além de reduzir o ataque de formigas cortadeiras (SERRA et. al.,2019).

A automação de processos visa simplificar os métodos de trabalho sem execução manual, abrindo caminho para uma atuação automática, padronizada e eficiente (MESKIV,2020).

As manufaturas florestais normalmente têm que entregar madeira com base em sua capacitância instalada de processamento e produção, o que requer planos de manejo bastante complexos, sofisticados e muitas vezes documentados. Nas pequenas propriedades rurais, por outro lado, são estabelecidos plantios florestais de pequena escala para maximizar o retorno monetário do investimento: nesse caso, a noção de floresta se confunde com a de parcela ou compartimento, e o planejamento pode ser feito em nível de unidade (MESKIV,2020).

O sucesso do investimento em plantações florestais, incluindo espécies de eucalipto, depende em grande medida do desenvolvimento de um plano de gestão adequado ao conjunto de circunstâncias que melhor descrevem e caracterizam a empresa. Um plano de manejo não é estático: em última análise, ao longo de sua execução, podem ser necessários ajustes e adequações, que devem ser levados em consideração levando em consideração imprevistos, novas situações e mudanças em fatores externos como as condições de mercado (MESKIV,2020).

Dessa maneira, com *softwares* de automação desenvolvidos especificamente para integrar sistemas e responder roboticamente a estímulos pré-determinados pela programação é possível automatizar tarefas repetitivas, tornando-as independentes do trabalho humano (MESKIV,2020).

Inspirado no funcionamento do sistema nervoso dos organismos vivos, redes profundas neurais convulsionais ou redes Neurais Convolucionais (CNNs) são algoritmos computacionais que seguem um modelo *feed-forward* onde todas as camadas são associadas a próxima camada, seguindo um caminho desde a entrada até a saída da rede. Essa classe de rede neural artificial tem sido aplicada com sucesso ao processamento e análise de imagens digitais, onde ao transformar as imagens em um espaço matricial numérico e atribuir pesos, torna-se possível simular sinapses, permitindo que o algoritmo mantenha o conhecimento adquirido (SILVA et. al., 2021).

Isso acontece por meio de ajustes de peso durante o treinamento, onde ocorre o processo de retro propagação, dividido em duas etapas. No primeiro caso, a amostra é passada para a rede para saída. Este resultado é comparado com o resultado desejado. A partir desta etapa, o erro é calculado e propagado da saída para a entrada e os pesos e limites são atualizados usando o gradiente descendente, a fim de reduzir a soma dos erros (SILVA et. al., 2021).

Usando esse conceito, as redes profundas têm a capacitância de extrair recursos, obter correlações entre variáveis para detectar padrões e aprender com a experiência por meio do ajuste automático de pesos e parâmetros para exemplos selecionados. Conseqüentemente, está no "estado da arte para problemas de categorização (SILVA et. al., 2021).

A proposta é utilizar as ferramentas disponibilizadas pelo MMDetection, pegando alguns dos frameworks de detecção de objetos suportados, que já possuem um pré-treinamento, tornando assim um segundo treinamento facilitado e com necessidade de menos épocas. Foram utilizados os seguintes AutoAssign, Fasterrcnn, NAS-FCOS, YOLOF, Retinanet, DETR, Cascade, TOOD, YOLOv3 e YOLOv3-Mixed, com esse último sendo apresentado pelo algoritmo YOLOv3 como um algoritmo com treino de precisão mista, e que de acordo com os treinos deles apresenta um Average Precision (AP) melhor que as outras versões do mesmo algoritmo. E treinar os algoritmos para identificar os cupinzeiros, tornando possível para fazendeiros tomar a devida medida para neutralizar ou até mesmo identificar algum cupinzeiro isolado e começar as medidas preventivas.

2 METODOLOGIA

2.1 Base de Dados

Foi utilizado o ambiente *do google collab*, uma máquina virtual rodando um sistema Linux, utilizando Python 3 e com acesso a uma GPU tesla T4 16GB, 12GB de memória ram e 107GB de espaço em disco.

Para o *set* do projeto foi utilizado 1 vídeo, obtido pelo laboratório de geomática da FAENG UFMS e gravado usando uma lente "Olho de peixe", que proporciona um maior ângulo de visão do ambiente, na resolução 2704 x 2624, em que os objetos ficam estáticos e a câmera se movimenta pelo ambiente (figura 1).

Figura 1 – Vídeo gravado usando a lente "olho de peixe".



Fonte: Elaborado pelo autor, 2023.

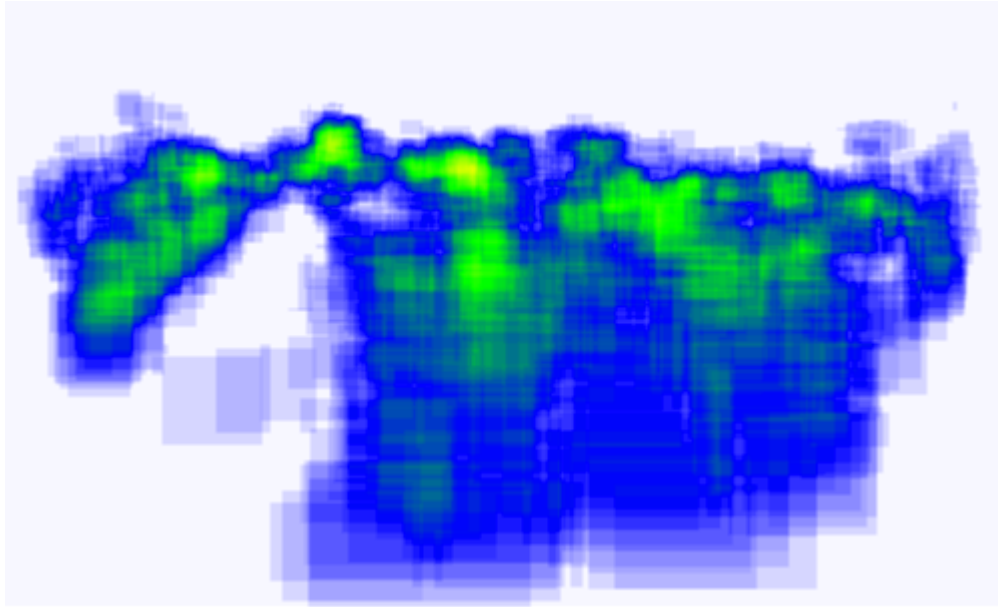
Foi utilizada a ferramenta de anotação do site *Roboflow*, que permite que as anotações sejam feitas manualmente pelo usuário pela página web e exportadas no formato coco, conforme pode ser observado na figura 2:

Figura 2 - Anotação do *Roboflow*

Fonte: Elaborado pelo autor, 2023.

Foram extraídas 1309 imagens, na qual foi retirada as imagens sem os objetos objetivo de treino, restando 822 imagens, separadas em 70% treino, 20% validação e 10% teste, que foram aleatoriamente separadas pela própria ferramenta do *Roboflow* para assim evitar que alguma etapa do treino tivesse uma sequência de imagens muito parecidas. As anotações foram feitas focando nos objetos próximos a câmera, e com uma visibilidade aproximada de x pixels, e não anotadas em caso de não visualização do objeto sem auxílio dos frames anteriores ou seguintes (figura 3).

Figura 3 – Posicionamento dos cupinzeiros e proposta de detectação



Fonte: Elaborado pelo autor, 2023.

2.2 Configuração Experimental

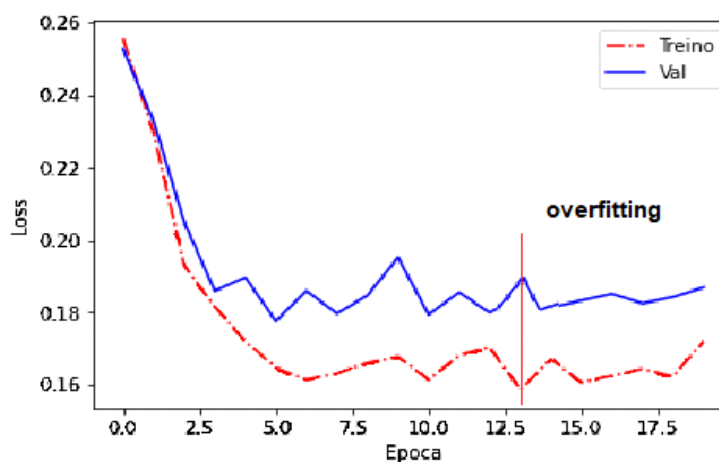
Foram testados os algoritmos em várias épocas, (10,12,20,24,30,48,50,60) a fim de encontrar um ponto de bom desempenho, mas que também possa vir a ser usado em outro *dataset*. Foram testadas algumas variações de *learning rate*, que foram sendo ajustadas de acordo com as anomalias apresentadas pelos algoritmos, que poderiam ser o algoritmo não retornar nenhuma marcação, muitas marcações de uma forma que o objeto do treinamento acabava sendo marcado por vários retângulos que abrangiam uma área muito grande ou um desempenho anormal na hora de detectar o objeto.

O próximo ponto tratado foi a regularização, onde era necessária uma forma de evitar que o treino sofresse de *overfitting*, que causaria seria quando o algoritmo ganha precisão no dataset de treino, porém quando exposto ao dataset de validação ele demonstra perda de desempenho por estar muito “especializado” nas imagens presentes no dataset de treino.

À medida que foi utilizada para tentar contornar o *overfitting* foi o *early stop* que é um método bem popular no *deep learning* e provavelmente o mais utilizado. A proposta do *early stop* é que o modelo treinado será utilizado na época em que a validação tenha o melhor desempenho antes do treino apresentar sinal de *overfitting*, mesmo com o

dataset de validação fazendo parte do mesmo conjunto de imagens que o de treino, o mesmo serve como uma forma de observar um possível overfitting (figura 4), já que as imagens são diferentes do dataset de treino. Com isso espera-se que o modelo treinado tenha um bom desempenho tanto com o dataset de treino quanto em um novo conjunto de imagens que venha a ser utilizado no futuro (GOODFELLOW; BENGIO; COURVILLE,2016).

Figura 4 - Exemplo de overfitting



Fonte: Elaborado pelo autor, 2023.

Quadro 1 – Algoritmos utilizados – Época do *Early Stop*

Algoritmos	Época
AutoAssign	12
Retinanet r101	12
TOOD	12
Retinanet r50	12
NAS-FCOS	12
YOLOv3 – Mixed	17
Fasterrcnn r50	13
Fasterrcnn r101	13
YOLOv3	20
YOLOF	20
Cascade r50	13
DETR	16

Fonte: Elaborado pelo autor, 2023.

Foram avaliados 1 versão de 5 algoritmos, *AutoAssign*, *Cascade*, *DETR*, *Nas-Fcos*, *TOOD*, 2 versões dos algoritmos *Fasterr* e *Retinanet*, em suas versões *r50* e *r101*, e 3 versões do algoritmo *YOLO*, sendo eles *YoloV3*, uma versão de treinamento misto do *YoloV3* que será utilizada a nomenclatura *YOLO-Mixed* e a versão *YOLOF*, uma modificação da versão original, que foi lançado em 2021.

Os algoritmos que foi utilizado apenas 1 versão foram preferencialmente usadas as versões o backbone R-50 com style pytorch quando disponível, com Lr schd sempre de 1x e com o maior box ap possível com tanto que as outras condições sejam atendidas.

A configuração de *learning rate* utilizada funcionava no formato $lr/8$ (como exemplo um algoritmo que era usado o *learning rate* de 0.001 era treinado com o *learning rate* de 0.000125) pois o algoritmo utilizado é originalmente feito para utilizar 8 placas gráficas, enquanto nos testes realizados só foi utilizado uma.

Quadro 2 - Algoritmos com o *lr* utilizado

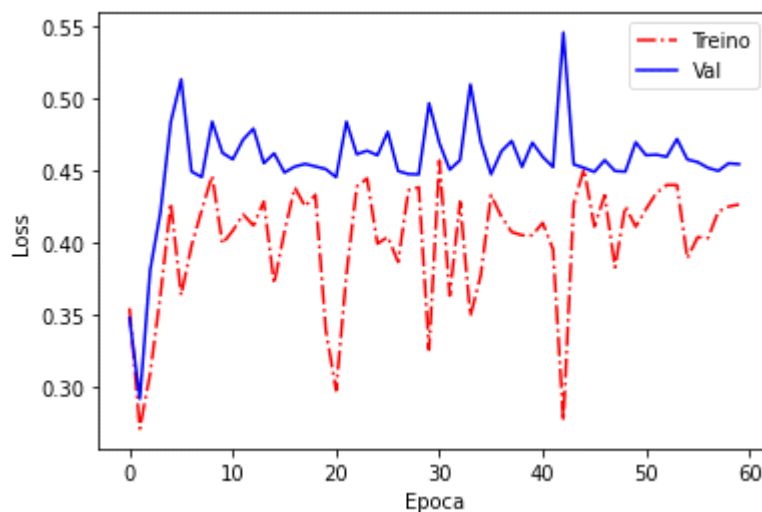
Algoritmos	Learning rate utilizado
AutoAssign	0.002
Retinanet r101	0.002
TOOD	0.001
Retinanet r50	0.002
NAS-FCOS	0.002
YOLOv3 – Mixed	0.008
Fasterrcnn r50	0.002
Fasterrcnn r101	0.002
YOLOv3	0.008
YOLOF	0.002
Cascade r50	0.0008
DETR	0.0008

Fonte: Elaborado pelo autor, 2023.

O *learning rate* dos algoritmos foram sendo modificados até chegar em um valor onde o algoritmo pudesse reconhecer o objeto treinado, visando reduzir a quantidade de *False Positive* (FP) e *False Negative* (FN), essas duas medidas afetam diretamente a

precisão que os algoritmos e poderiam não alcançar o valor mínimo estipulado quando chegar a hora de medir o *Intersection Over Union* (IoU). Para encontrar um *learning rate* compatível foi utilizado como base um treino de 48 épocas, onde era verificado se o resultado do treino apresentava um gráfico em que os dados faziam uma curva de melhoria progressiva de treino e validação.

Figura 5 - Gráfico do algoritmo DETR com o *learning rate* a 0.002



Fonte: Elaborado pelo autor, 2023.

Dessa forma era esperado que o algoritmo pelo menos conseguisse atingir o mínimo de IoU de 0.50, que não seria alcançado com o algoritmo gerando muitos False Positive (FP) resultante de um treino desregulado.

Figura 6 - Exemplo de imagem com um número alto de *False Positive*

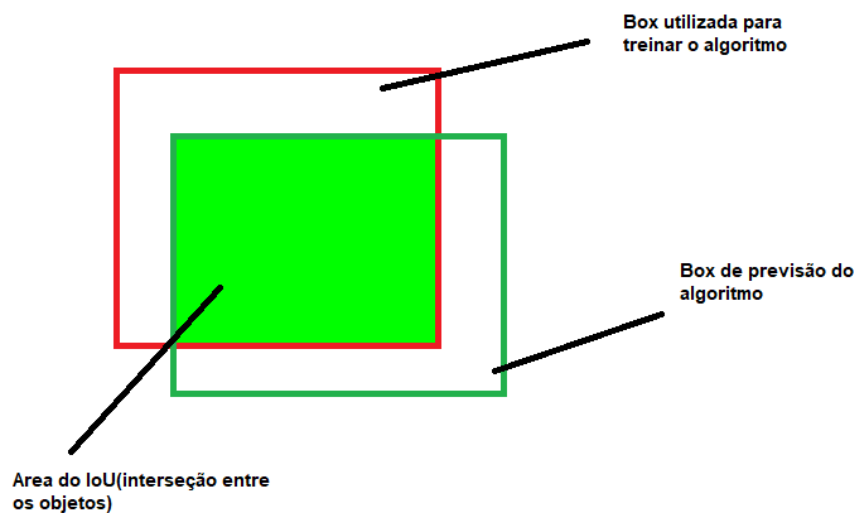


Fonte: Elaborado pelo autor, 2023.

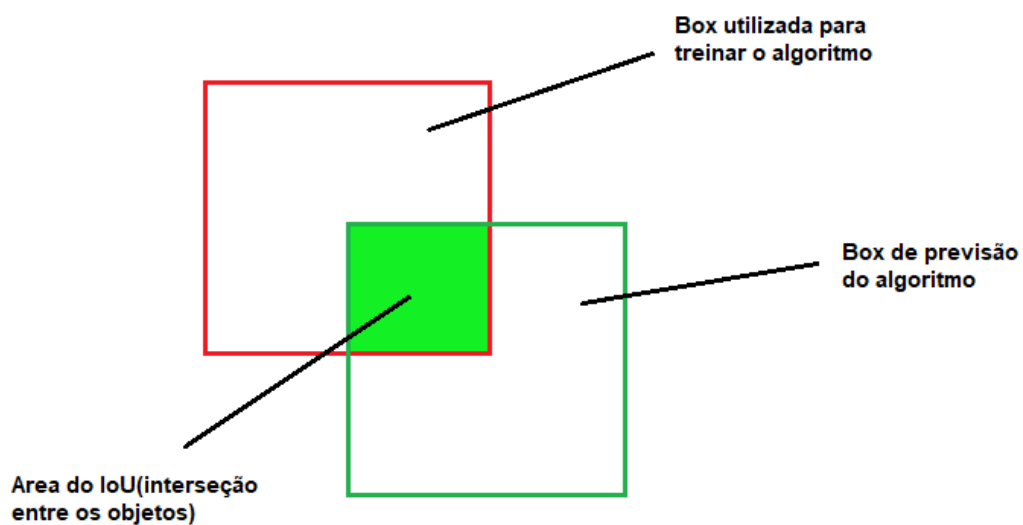
2.3 Métrica de Avaliação

O método de avaliação utilizado foi o *Average Precision (AP)*, que utiliza no seu cálculo o *Intersection Over Union (IoU)*, o IoU avalia a interseção entre a *box* de treino e a *box* de predição dessa forma, o valor que iremos obter vai depender da área que a *box* de predição do algoritmo ocupou em relação a área de união de ambas as *boxes*, assim caso o nosso valor de *IoU* seja de 0.65, significa que a área de interseção foi de 65% da união das duas *boxes*. Evitando que um algoritmo faça uma predição da imagem inteira, pois criaria uma área de união muito grande e com uma interseção pequena e que não alcançaria o valor mínimo imposto para que a imagem seja considerada *True Positive (TP)* (VAN ETTEN,2019).

No nosso caso o valor utilizado e comum em testes de detecção foi 0.50, significa que valores acima de 0.50 serão considerados *True Positive (TP)*, valores abaixo disso serão considerados *False Positive (FP)*.

Figura 7 - Exemplo de *True Positive* (TP)

Fonte: Elaborado pelo autor, 2023.

Figura 8 - Exemplo de *False Positive* (FP)

Fonte: Elaborado pelo autor, 2023.

Com essas informações é possível chegar no valor da precisão, conforme a equação 1:

$$Precisão = \frac{TP}{TP+FP} \quad (1)$$

Então toda vez que um objeto for marcado pelo algoritmo e sua interseção for maior que 0.5, teremos 1 contador no nosso TP, quando a interseção não alcançar o valor mínimo teremos 1 no contador de FP.

Outra medição utilizada será o *Recall*.

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

Com isso conseguiremos saber também se em algum momento o algoritmo não fez a marcação de um objeto que deveria ter sido marcado. Fazendo assim um contador no FN toda vez que o algoritmo deixar de marcar um objeto que era para ser marcado (GAD,2020).

3 RESULTADOS E DISCUSSÃO

Quando avaliado os resultados com o primeiro limitador do IoU (0.50) (quadro 3), temos um melhor desempenho apresentado pelos algoritmos *AutoAssign*, *TOOD* e a versão r101 do *retinanet*, que mesmo tendo ficado empatado com o *TOOD* quando a porcentagem de interseção avaliada era 50%, era perceptível que teria diferenças entre os dois com um IoU maior já que o *Retinanet* r101 apresentava muitos False Positive (FP).

Figura 9 - Imagem do Retinanet r101.



Fonte: Elaborado pelo autor, 2023.

Quadro 3 - Quadros dos algoritmos e sua precisão de acordo com o IoU 0.50

Algoritmos	IoU 0.50
AutoAssign	0.940
Retinanet r101	0.891
TOOD	0.891
Retinanet r50	0.876
NAS-FCOS	0.853
YOLOv3 – Mixed	0.853
Fasterrcnn r50	0.840
Fasterrcnn r101	0.833

YOLOv3	0.782
YOLOF	0.781
Cascade r50	0.077
DETR	0.695

Fonte: Elaborado pelo autor, 2023.

Foi então aplicado um limitador mais rigoroso, aumentando o *IoU* para 0.75 (quadro 4) onde o algoritmo *TOOD* apresentou uma AP maior que os demais, seguido pelo *AutoAssign* que obteve o melhor desempenho no *IoU* 0.50.

Quadro 4 - Quadros dos algoritmos e sua precisão de acordo com o *IoU* 0.75

Algoritmos	IoU 0.75
TOOD	0.350
AutoAssign	0.311
Retinanet r101	0.302
Fasterrcnn r101	0.282
Fasterrcnn r50	0.277
Cascade r50	0.252
Retinanet r50	0.249
NAS - FCOS	0.249
YOLOF	0.229
YOLOv3 - Mixed	0.155
YOLOv3	0.099
DETR	0.065

Fonte: Elaborado pelo autor, 2023.

Com os Dois algoritmos que obtiveram o melhor desempenho com o *IoU* 0.75 é bastante perceptível uma boa quantidade de *False Positive* (FP) e *False Negative* (FN), possivelmente gerado pela pequena quantidade de épocas de treino e pela coloração do ambiente, que apresenta áreas do terreno com colorações diferentes no solo e manchas em arvores com formatos e cores que podem algumas vezes se parecer com o objeto de treinamento. Mesmo com um valor baixo no AP do *IoU* 0.75 os algoritmos ainda são capazes de identificar os objetos treinados.

Dentre os algoritmos testados o *AutoAssign* foi o que obteve o melhor desempenho com o *IoU* de 0.5 e manteve um desempenho próximo do desempenho do algoritmo *TOOD* que foi o que melhor desempenhou com *IoU* 0.75, conseguindo um bom

desempenho no IoU 0.5 provavelmente pelo uso de um modelo de detecção que se adapta a aparência do objeto ao invés de apenas usar um formato pré-definido como é o caso do Retinanet (ZHU,2020).

Algoritmos	Recall
AutoAssign	0.566
Fasterrcnn r101	0.532
NAS-FCOS	0.514
Cascade r50	0.490
Fasterrcnn r50	0.473
DETR	0.467
TOOD	0.350
Retinanet r101	0.302
Retinanet r50	0.249
YOLOF	0.229
YOLOv3-Mixed	0.155
YOLOv3	0.099

Fonte: Elaborado pelo autor, 2023.

Com a tabela de recall temos os algoritmos que deixaram de marcar os cupinzeiros quando eles estavam presentes na imagem, False Negative (FN), então com um bom desempenho no IoU 0.50 e também o algoritmo AutoAssign também obteve um desempenho melhor que os outros avaliados.

Figura 10 - Imagens do algoritmo AutoAssign



Fonte: Elaborado pelo autor, 2023.

Figura 11 - Imagens do algoritmo TOOD



Fonte: Elaborado pelo autor, 2023.

4 CONCLUSÃO

Neste projeto foram treinados alguns algoritmos disponíveis no MMDetection utilizando a medida de early stop e comparado o desempenho deles utilizando a métrica de IoU 0.50, 0.75 e recall para saber qual desses algoritmos teriam o melhor desempenho em identificar cupinzeiros em uma plantação de eucalipto. Tendo como destaque o algoritmo AutoAssign, que demonstrou uma alta precisão especialmente na tabela de IoU de 0.50, e também sendo o algoritmo que menos deixou de marcar o objeto de estudo quando o mesmo estava presente, de acordo com a tabela de recall.

Considerando o tamanho do Brasil e da variedade de ecossistemas e cores a qual os algoritmos seriam expostos, é improvável que o algoritmo que obteve o melhor desempenho durante este teste também obtenha o melhor resultado em outros ambientes, principalmente porque as imagens utilizadas neste projeto foram tiradas em uma única época do ano, fazendo assim com que alguns algoritmos tivessem maior dificuldade de diferenciar a coloração do ambiente com a dos cupinzeiros. Mesmo assim a maioria dos algoritmos treinados conseguiram atingir alta precisão na identificação dos cupinzeiros.

Seria uma boa proposta realizar mais testes com um dataset de imagens mais diversas em coloração e formatos diferentes de fotográfica, como drones ou lentes alternativas, utilizando diferentes metodologias de treinos, que poderiam favorecer algoritmos que não obtiveram um desempenho tão bom durante esses testes e que são muito populares nos projetos de detecção de objetos, como o algoritmo Yolo.

5 REFERÊNCIAS

BECCALONI, G.; EGGLETON, P. Order Blattodea. **Zootaxa**, v.3703, ed.1, p.46-48, 2013.

CHAGAS, Thiago Xavier et al. Biodegradação de tocos de Eucalyptus sp. pela ação de cupins xilófagos. 2016.

CHEN, Kai et al. MMDetection: Open mmlab detection toolbox and benchmark. **arXiv preprint arXiv:1906.07155**, 2019.

CONSTANTINO, R. Chave ilustrada para identificação dos gêneros de cupins (Insecta: Isoptera) que ocorrem no Brasil. **Papéis Avulsos de Zoologia**, v.40, n.25, p.387-448, 1999.

EGGLETON, P. An Introduction to Termites: biology, taxonomy and functional Morphology. In: BIGNELL, D. E.; ROISIN, Y.; LO, N. (Eds). **Biology of Termites: A Modern Synthesis**, Springer, 2010, p.1-26.

FAO. **Global forest resources assessment 2020: key findings**. Roma: FAO, 2020. Disponível em: <https://doi.org/10.4060/ca8753en>. Acesso em: 26 dez. 2022.

FERREIRA, Patrícia Vieira; DE OLIVEIRA, Louise Antunes; DE LAVOR LOPES, Paloma. ANÁLISE DE INVESTIMENTO ECONÔMICO-FINANCEIRO ENTRE PLANTAÇÕES DE EUCALIPTO E PREVIDÊNCIA PRIVADA. **Episteme Transversalis**, v. 7, n. 1, 2017.

GAD, Ahmed Fawzy. Evaluating object detection models using mean average precision (mAP). **PaperspaceBlog**, 2020.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. Regularization for deep learning. **Deep learning**, p. 216-261, 2016.

INDÚSTRIA BRASILEIRA DE ÁRVORES. **Relatório 2019**. Brasília, DF: IBA, 2019. 80 p. Disponível em: <https://iba.org/datafiles/publicacoes/relatorios/iba-relatorioanual2019.pdf>. Acesso em: 26 dez. 2022.

LI, W. et al. A global yield dataset for major lignocellulosic bioenergy crops based on field measurements. **Scientific Data**, London, v. 5, p. 180169, 2018. Disponível em: <https://www.nature.com/articles/sdata2018169>. Acesso em: 26 dez. 2022.

MESKIV, Valdemir. **Automação de uma estufa agrícola destinada a produção de mudas de eucalipto**. 2020. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.

MOÇO, Vítor Emanuel de Oliveira Portugal Marques. **Impactes da intensidade de mobilização do solo em plantações de eucalipto**. 2019. Tese de Doutorado.

PENTEADO, J.; PICHIELLI, K.; SOARES, S. Eucalipto. Disponível em:< <https://www.embrapa.br/florestas/transferencia-de-tecnologia/eucalipto/perguntas-e-respostas>>. Acesso em: 26 dez. 2022.

SERRA, Ademar Pereira et al. Fundamentos técnicos para implantação de sistemas de integração lavoura-pecuária-floresta com eucalipto. **Embrapa Gado de Corte-Capítulo em livro científico (ALICE)**, 2019.

SILVA, João Gabriel Junqueira da et al. Redes neurais profundas para detecção e contagem automática de moscas brancas no contexto agrícola. 2021.

VAN ETTEN, Adam. Satellite imagery multiscale rapid detection with windowed networks. In: **2019 IEEE winter conference on applications of computer vision (WACV)**. IEEE, 2019. p. 735-743.

CHEN, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C. C., & Lin, D. MMDetection: Open MMLab Detection Toolbox and Benchmark. In: **arXiv preprint arXiv:1906.07155**. 2019.

ZHU, Benjin et al. AutoAssign: Differentiable Label Assignment for Dense Object Detection. In: **arXiv:2007.03496**, 2020.