DETECÇÃO DE PALMEIRAS EM IMAGENS AÉREAS COM YOLO E DARKNET

FELIPPE DE OLIVEIRA LIMA[1], JOSÉ BRUNO SANTOS PINHEIRO[2], HIDEO ARAKI[3]

RESUMO

O consumo global de óleo de palma vem crescendo nos últimos anos passando de 22.5 milhões de toneladas em 2010 para 40 milhões de toneladas em 2020. O aumento é devido ao uso do óleo da palma em diversas finalidades econômicas como farmacêutica, cosmética, dentre muitas outas. O número exato de palmeiras em uma área de plantio é uma importante informação para monitoramento e controle para prever a produtividade. Nesse trabalho, propõe-se a verificar se técnica *You Only Look Once* (YOLO) pode ser utilizada para detecção de palmeiras através de imagens aéreas. Foi utilizado um conjunto de imagens de palmeiras para a realização do treinamento. Para o processamento, usou-se os arquivos oriundos da YOLOv4 em conjunto com a arquitetura Darknet (mesmo nome do framework) dentro do ambiente do Google Colab. No total, foram utilizadas 3500 imagens de treinamento e 500 imagens de validação para o teste do modelo. Ao todo, foram realizadas 2.800 épocas na fase de treinamento com um *Mean Avarage Precision* (MAP) de 85,49%.

Palavras-chave: Detecção de objetos; Palmeiras; *Deep Learning*.

1 INTRODUÇÃO

A aprendizagem profunda (*Deep Learning*, em inglês) tem sido amplamente aplicada para detecção de objetos devido ao seu sucesso na extração de padrões em imagens. Cada vez mais técnicas e arquiteturas são empregas para a identificação de objetos com o objetivo de melhoria de acurácia e redução de tempo de processamento.

Diferentes conjuntos de dados são aplicados no campo da visão computacional para o treinamento em *Deep Learning* (DL). Para se construir um modelo de aprendizado robusto para visão computacional, é necessário aplicar conjuntos de dados de alta qualidade na fase de treinamento. Alguns desses conjunto de dados são amplamente aplicados como o CIFAR-10. Este conjunto de dados possui 60.000 imagens em 10 diferentes classes. Outro exemplo é o *dataset* MPII *Humam Pose*. Possui cerca de 25 mil imagens contendo mais de 40 mil pessoas com articulações corporais anotadas com imagens extraídas diretamente de vídeos do YouTube. O Quadro 1 mostra um resumo dos dados dos principais *datasets*.

[1] Universidade Federal do Paraná, felippeufrrj94@gmail.com
[2] Universidade Federal do Paraná, jbrunopinheiro@hotmail.com
[3] Universidade Federal do Paraná, araki.hideo@gmail.com

Quadro 1 - Comparativo dos principais *datasets* em visão computacional para detecção de objetos.

Dataset	Exemplos de Classes	Quantidade de Classes	Número de Imagens	Dimensão das Imagens (pixels)
VOC	Pessoa, gato, cachorro, carro e cadeira.	20	11.530	500 x 334
ImageNet	Avião, pássaro, carro, cadeira e capacete	200	1.200.000	64 x 64
COCO	Faca, cachorro, cavalo, barco e bicicleta	80	123.287	Variado
CIFAR – 10	Gato, veado, cavalo, cachorro e caminhão.	10	60.000	32 x 32
MPII Humam Pose	Exercícios de condicionamento, esportes e atividades domésticas	410	25.000	200 x 200

Fonte: Autoria própria.

Antes do advento do DL, o procedimento padrão de detecção de objetos dentro do campo da visão computacional era feito em três diferentes passos: (i) detecção da região de interesse, (ii) extração de características das regiões/alvos e (iii) implementação de um classificador supervisionado. Esse procedimento metodológico, apesar de mostrar bons resultados, a maioria das vezes é incapaz de apresentarem melhorias (Pham et al., 2020).

O campo da visão computacional é uma área da inteligência artificial que tem como objetivo interpretar informações a partir de imagens. Assim, a tarefa de detecção de objetos consiste em determinar o local na imagem onde um determinado alvo está presente, bem como classificar esses objetos. Ou seja, localizar o objeto junto com a classe é chamado de detecção de objetos. Nesse sentido, a técnica *You Only Look Once* (YOLO) é um tipo de técnica para detecção.

A ideia principal da técnica YOLO é otimizar o cálculo de previsões em várias posições da imagem de entrada sem utilizar o método de janelas deslizantes (ZAFAR et al., 2018). A YOLO, faz todo o processo com uma rede única, como indica o nome. Ela divide a imagem em uma grade S x S e cada grade faz a previsão de um determinado número de caixas delimitadoras com 4 componentes: as coordenadas (bx, by) representam o centro da caixa,

em relação à localização da célula da grade, as dimensões da caixa (bw, by), probabilidade de existir um objeto dentro da caixa delimitadora (pc) e a classe (c) correspondente ao objeto (HUANG; PEDOEEM; CHEN, 2018).

Independentemente do número de caixas delimitadoras encontradas, ela detecta apenas um objeto e prevê uma probabilidade condicional por classe. As caixas delimitadoras na representação vão ter a espessura modificada de acordo com o seu grau de confiança de detecção de um objeto (REDMON et al., 2016).

A YOLO é capaz de contribuir com o desenvolvimento do campo de detecção de objetos associado a imagens aéreas de alta resolução, apesar de detecções erradas ou com baixa acurácia ainda acontecerem devido ao tamanho dos objetos em relação a resolução espacial (objetos pequenos) e uma base de dados para treinamento ser pequena e limitada. Em geral, a detecção de um determinado objeto depende da correspondente base de dados de imagens de treino para uma melhor performance da detecção.

Entretanto, poucos conjuntos de dados estão disponíveis relativos a imagens aéreas, como o *Vehicle Aerial Imaging from Drone* (VAID) que contém mais de 6.000 imagens aéreas de diferentes ângulos, iluminações e tipos de veículos (LIN; TU; LI, 2020). Além do mais, devido a variações de sombra, iluminação e altura de voo, a detecção de objetos por imagens aéreas continua a ser um problema desafiador.

Hoeser e Bachofer (2020) fizeram um levantamento de dados de aplicações, técnicas e arquiteturas de redes neurais convolucionais sobre trabalhos aplicando DL no âmbito do sensoriamento remoto. No total foram levantados um total de 429 trabalhos divididos em aplicações distintas como agricultura, mudança de uso e cobertura da terra e transporte. Pode ser notado que dentre as principais aplicações, a área de transporte se sobressai com aproximadamente 27% dos estudos. Por outro lado, menos de 10% dos trabalhos são aplicados em agricultura. No total, apenas aproximadamente 1% foi aplicado em palmeiras no levantamento do artigo até o ano de 2019.

Devido a aplicações na esfera da detecção, o presente projeto pretende contribuir para a construção de uma abordagem metodológica para a detecção de palmeira-de-dendê a partir da técnica YOLO v4, tendo como resultado uma detecção automática baseada em DL servindo de suporte e tendo como consequência o melhoramento no gerenciamento na produção de palmeiras.

2 MATERIAIS E MÉTODOS

A coleta de dados para o treinamento do modelo envolve primeiramente a ferramenta LabelImg, disponível em repositório no GitHub. Rotular uma imagem é a primeira parte e mais significativa para a detecção de objetos. A rotulagem é um processo lento e manual, mas quanto mais for feita de maneira precisa e rigorosa, melhor será o modelo.

O conjunto de dados foi dividido inicialmente em 3.500 imagens de treinamento e 500 imagens para validação obtidas a partir de divisões iguais de uma única imagem da região de estudo. Cada imagem possui o tamanho de 224 x 244 pixels e possui um arquivo correspondente com a anotação de cada palmeira catalogada contida na imagem. Todo o processamento foi gerado no Google Colab e os produtos gerados foram armazenados no serviço Google Drive.

Alguns cuidados devem ser tomados para a coleta de imagens para a criação do conjunto de dados (*datasets*). A parte manual não se limita a apenas a coletas de imagens de forma aleatória, é necessário um trabalho de padronização dos tamanhos das imagens para que se tenham dimensões iguais e de um tamanho que comporte o treinamento de acordo com a técnica. Assim sendo, as imagens devem ser catalogadas de formas diferentes, não agregando valor imagens repetidas. Se deve também realizar a rotação dos objetos alvos para uma melhor generalização dos dados. A Figura 1 mostra o ambiente da ferramenta LabelImg.

Figura 1 - Ambiente da ferramenta LabelImg

Fonte: Autoria própria.

Para a utilização do framework Darknet é necessário realizar o download do framework de mesmo nome e que pode ser obtido na página no endereço eletrônico em https://github.com/AlexeyAB/Darknet.

Um ponto importante para a agilidade no processo dos dados é utilizar a GPU do Google Colab. Tendo em vista o difícil acesso de computadores de alta capacidade de processamento de dados, o Google Colab surge como opção com acesso gratuito as GPUs com uma simples configuração interna. Assim, se torna mais ágil o desempenho do processamento em cálculos avançados de DL.

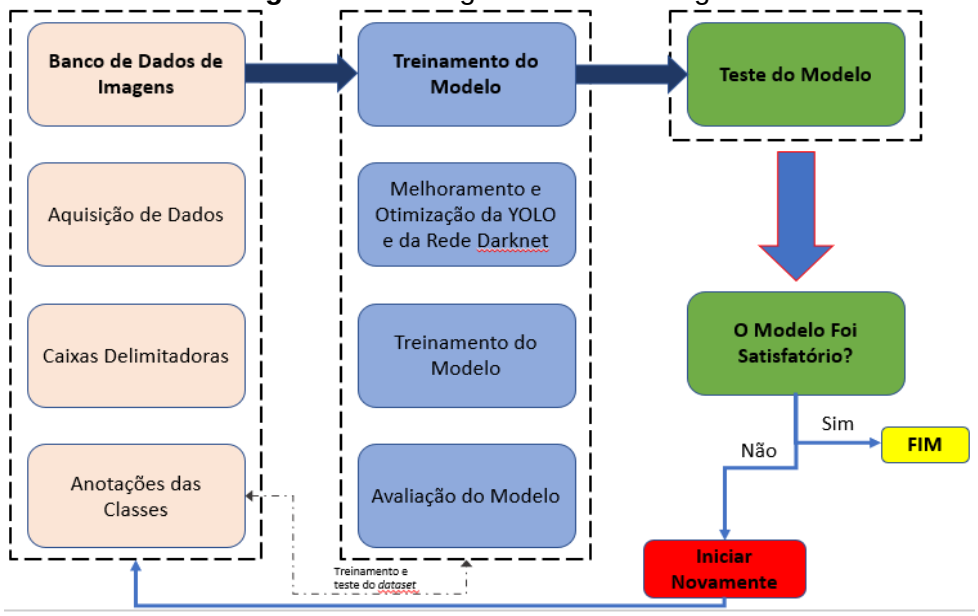
As bibliotecas utilizadas para o processamento dos dados envolveram a visualização dos dados, conexão com o google drive e bibliotecas de aprendizagem de máquina. A seguir são listadas as bibliotecas importadas no processamento:

- Tensorflow – utilização da GPU;
- Google Drive – armazenamento dos dados de entrada;
- OpenCV – visualização da imagem gerada com a detecção;
- Matplotlib – plotagem da imagem para visualização.

A presente pesquisa propõe o desenvolvimento de um sistema de detecção automatizado aplicado no sensoriamento remoto para detecção de palmeiras e que seja capaz de caracterizar as particularidades dos alvos de estudo em imagens na faixa do RGB e testar a qualidade do treinamento.

O modelo de detecção das palmeiras foi desenvolvido em três etapas principais (Figura 2). Em primeiro lugar, foi criado um *dataset* com as imagens das palmeiras em imagens aéreas de alta resolução. Os objetos (palmeiras) foram anotados através de caixas delimitadoras nas suas respectivas classes. Em seguida, a rede YOLO foi treinada e otimizada nos dados desenvolvidos. A cada 100 etapas de treinamento, as métricas de avaliação foram calculadas para validar o desempenho da detecção. Finalmente, o melhor conjunto de peso foi selecionado para a detecção de palmeiras a partir de imagens aéreas de alta resolução.

Figura 2 - Fluxo geral da metodologia.



Fonte: Autoria própria.

A primeira parte do fluxo metodológico se refere a preparação dos dados de entrada para o treinamento da rede. É necessário um banco de imagens de resolução adequada para atender a detecção de palmeiras. As caixas delimitadoras são feitas pelo operador de forma manual, fazendo a delimitação dos alvos de interesse e anotando a(s) classe(s) correspondentes.

A segunda parte do processo se refere ao treinamento em si do modelo. O Google Colab permite uma conexão com o google drive onde os dados podem ser armazenados. Portanto, o treinamento é feito com o YOLOv4 em conjunto com os dados do Darknet armazenados na nuvem.

A terceira e última parte do processo se faz pela verificação do conjunto de dados de teste. Após o treinamento, pode-se testar e visualizar em uma imagem de mesmas características, utilizando o OpenCV, que fazem parte do conjunto de teste (validação). Isso fornecerá o resultado visual da detecção através de caixas delimitadoras nas palmeiras. Além disso, a avaliação será feita através das métricas e do *score* da caixa delimitadora, fornecendo a probabilidade de uma palmeira ter sido encontrada de forma correta.

Uma das métricas importantes para avaliação é o Intersection Over Union (IoU). Esta métrica mede a sobreposição entre dois limites. No caso da detecção de objetos, usa-se para medir o quanto a caixa delimitadora prevista se sobrepõe sobre a caixa delimitadora real

(PADILLA; NETTO; DA SILVA, 2020). O valor 1 (100%) seria a condição perfeita, porém esse valor é quase impossível, pois a métrica vai penalizar qualquer deslocamento da imagem.

O treinamento com a YOLO utilizando a rede neural profunda, cuja arquitetura é chamada de Darknet, necessita de modificações de alguns arquivos que são adquiridos a partir do descarregamento dos seus dados a partir da sua página do GitHub. A mudança é necessária, pois os arquivos estão com as configurações padrões, sendo preciso mudar para cada dado personalizado.

Para ser realizado um treinamento com um bom desempenho, é necessário que se tenha um conjunto numeroso de imagens para realizar o treinamento. Porém, o número de imagens ainda é incerto e varia para cada caso específico. Li et al. (2020) em seu trabalho para detecção de objetos a partir de imagens orbitais, utilizou mais de 23.000 imagens para 20 classes. Já Xia et al. (2018) utiliza 2.806 imagens, porém com mais de 188.000 caixas delimitadoras. Portanto, a quantidade de imagens e objetos variam para cada estudo, sendo importante conseguir o maior número de objetos detectáveis possível.

A próxima etapa para o treinamento é a configuração do arquivo da rede neural (cgf) para ajustar ao modelo o conjunto de dados. Após a configuração da rede e de todo o conjunto de dados necessários mencionados anteriormente, existe uma necessidade de um processo de *transfer learning* para o processamento inicial e geração dos primeiros pesos da rede. O arquivo de pesos nessa fase é o yolov4.conv.137 obtido através do repositório do Darknet no site GitHub.

Para as 100 primeiras épocas é comum ter uma perda média (*average loss*) alta, pois o treinamento ainda está no início. Para cada conjunto de dados o valor ideal de perda média irá variar. O valor médio é 3 para dados mais complexos, porém a medida dependerá de diversas variáveis do treinamento (tamanho do conjunto de dados e quantidade de anotações).

3 RESULTADOS E DISCUSSÕES

A detecção de objetos através da YOLO em conjunto com sua CNN, chamado de Darknet, mostrou resultados satisfatórios de acordo com as métricas de avaliação e resultados visuais na detecção de palmeiras como F1-Score (80%) e o Recall em 89%.

Foram realizadas 2.800 épocas de treinamento com um MAP final em 85%. A Tabela 1 mostra os resultados finais detalhados das métricas de avaliação da YOLOv4. Estes

resultados de detecção mostram que o método proposto é eficaz para a detecção de palmeiras.

Tabela 1 - Resultados de detecção da YOLOv4

Métrica de Avaliação	Valor
Precisão	0,74
Recall	0,89
MAP	0,85
F1-Score	0,80
IOU	0,53
Avg. Loss	2,94

Fonte: Autoria própria.

Outro ponto a se destacar no treinamento é a evolução temporal durante as épocas que não mostrou sinal de *overfitting*, ou seja, o treinamento ainda poderia ser continuado observando o valor do MAP para prever melhorias do treinamento. As Figuras 3 e 4 mostram os resultados da detecção de palmeiras com suas caixas delimitadoras.

Figura 3 - Detecção de palmeiras com Yolov4 (Área 1)

Fonte: Autoria própria.

Figura 4 - Detecção de palmeiras com Yolov4 (Área 2)

Fonte: Autoria própria.

Em ambas das imagens os *scores* de probabilidade foram definidos com um valor mínimo de limiar de 30%. O que pode ser observado é que algumas palmeiras apenas aparecem em parte da imagem e mesmo assim o detector consegue identificar o alvo. O IoU (53%) apresentou resultados intermediários, mas um aumento número de épocas e generalização dos dados pode ser melhor o seu resultado. Entretanto as outras métricas de avaliação mostraram resultados satisfatórios na acurácia geral da rede treinada.

4 CONCLUSÔES

Nesta pesquisa, foi introduzido um conjunto de imagens aéreas para a avaliação da técnica YOLOv4 em conjunto com a Darknet para a detecção de palmeiras. O conjunto de dados possui 3.500 imagens de treinamento e 500 de validação capturadas em uma única condição de iluminação e altura de voo. A YOLOv4 foi capaz de interpretar características das palmeiras em regiões com outras árvores de mesma tonalidade de cor contidas na imagem. Outro aspecto positivo foi a detecção de palmeiras que estão visíveis parcialmente na imagem, concluindo que o treinamento foi eficaz em detectar o formato e as características da copa.

A generalização dos dados é algo importante em estudos de detecção de objetos em imagens aéreas, portanto, futuramente o conjunto de dados precisa ser aperfeiçoado para se adequar melhor em outras áreas e em diferentes situações.

REFERÊNCIAS

HOESER, Thorsten; BACHOFER, Felix; KUENZER, Claudia. Object detection and image segmentation with deep learning on Earth observation data: A review—Part II: Applications. **Remote Sensing**, v. 12, n. 18, p. 3053, 2020.

HUANG, Rachel; PEDOEEM, Jonathan; CHEN, Cuixian. YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers. In: **2018 IEEE International Conference on Big Data (Big Data)**. IEEE, 2018. p. 2503-2510.

LI, Ke et al. Object detection in optical remote sensing images: A survey and a new benchmark. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 159, p. 296-307, 2020.

LIN, Huei-Yung; TU, Kai-Chun; LI, Chih-Yi. VAID: An Aerial Image Dataset for Vehicle Detection and Classification. **IEEE Access**, v. 8, p. 212209-212219, 2020.

PADILLA, Rafael; NETTO, Sergio L.; DA SILVA, Eduardo AB. A survey on performance metrics for object-detection algorithms. In: **2020 International Conference on Systems, Signals and Image Processing (IWSSIP)**. IEEE, 2020. p. 237-242.

PHAM, Minh-Tan et al. YOLO-Fine: One-stage detector of small objects under various backgrounds in remote sensing images. **Remote Sensing**, v. 12, n. 15, p. 2501, 2020.

REDMON, Joseph et al. You only look once: Unified, real-time object detection. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. 2016. p. 779-788.

XIA, Gui-Song et al. DOTA: A large-scale dataset for object detection in aerial images. In**: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. 2018. p. 3974-3983.

ZAFAR, Iffat; TZANIDOU, Giounona; BURTON, Richard; PATEL, Nimesh; ARAUJO LEONARDO. Hands-On Convolutional Neural Networks with TensorFlow: Solve Computer Vision Problems with modeling in TensorFlow and Python. Packt, 2018. 274 p.