

Relatório Projeto 2 de Fundamentos de Sistemas Inteligentes

Vitor Vasconcelos de Oliveira

180114778

Professor: Dêbio

dept. de Ciência da Computação – Universidade de Brasília (UNB)

Brasília, Brasil

vitorvasconcelos05062000@gmail.com

Abstract—Esse Relatório é uma breve apresentação dos resultados do projeto 2 da disciplina de Fundamentos de Sistemas Inteligentes, o qual consiste em treinar uma rede YoloV4 para identificar a presença de infestações de Bemisia tabaci (white-flies) em folhas de mandioca, além disso também serão feitas comparações com os resultados do artigo Improving In-field Cassava Whitefly Pest Surveillance with Machine Learning [1].

Index Terms—YoloV4, Machine Learning

I. INTRODUÇÃO

O problema de detecção de objetos é amplamente reconhecido no meio de aprendizado de máquina e análise e processamento de imagens. O problema consiste em identificar tanto classes na imagem como a localização do objeto nessa imagem. Existem varias técnicas capazes de realizar essa atividade, R-CNN, SPP-net, Faster R-CNN mas aqui vamos focar principalmente na YOLO.

A YOLO foi desenvolvida por Joseph Redmon e Ali Farhadi em 2015 durante o período de seu doutorado e causou grande impacto. Trata-se de um método de detecção de objetos de passada única que utiliza uma rede neural convolucional como extrator de características. Sua principal diferenciação em relação a outras técnicas é justamente sua passada única que aumenta consideravelmente sua velocidade de detecção.

Neste projeto, objetiva-se desenvolver e treinar uma rede YoloV4 para identificar a presença de infestações de Bemisia tabaci (whiteflies) em folhas de mandioca além de classificar sua abundancia em 3 classificações, baixa, média e alta.

II. DATASET

O Dataset oferecido para o desenvolvimento do projeto conta com 3000 mil imagens de folhas de mandiocas infestadas com as chamadas whiteflies. Para cada imagem existe um respectivo arquivo do formato .xml com anotações das posições desta whitefly. Entretanto, para leitura da arquitetura YoloV4 foi necessária a transformação dessas anotações em arquivos .txt que pudessem ser compreendidos pelo modelo.

Com relação à sua separação, o dataset foi dividido seguindo o mesmo formato estabelecido pelo artigo [1], 2600 imagens para treinamento e 200 imagens para validação e outras 200 para teste.

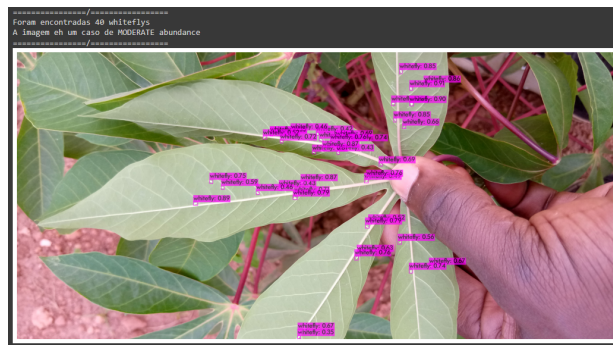


Fig. 1. Exemplo de predição gerada pela YoloV4, Moderate abundance

III. YOLOV4

Para aplicar a YoloV4 utilizou-se como ambiente de desenvolvimento o ambiente Google Colab. As principais referências de auxílio para sua aplicação foram:

- <https://medium.com/analytics-vidhya/train-a-custom-yolov4-object-detector-using-google-colab-61a659d4868>
- <https://github.com/AlexeyAB/darknet>

A rede Yolo desenvolvida conta com um total de 161 camadas, e foi construída em batches de 64, ou seja, são carregadas 64 imagens por iteração de treinamento, com subdivisões de 16 imagens sendo processadas por vez no batch. Sua funcionalidade principal é a detecção de regiões que delimitam focos da Bemisia tabaci, além de determinar a partir da quantidade de focos localizados, a classificação da imagem em Low ($n < 10$), Moderate ($10 \leq n \leq 100$) ou Super ($100 < n$) abundancia.

IV. RESULTADOS E COMPARAÇÕES

A partir da aplicação da função de mapeamento da YoloV4, que realiza as predições nos 200 arquivos de teste e retorna um pequeno relatório, foram obtidos seguintes resultados:

- True positives (TP) = 6224
- False positives (FP) = 7139
- False Negatives (FN) = 8239
- precision = 0.47
- recall = 0.43
- F1-score = 0.45

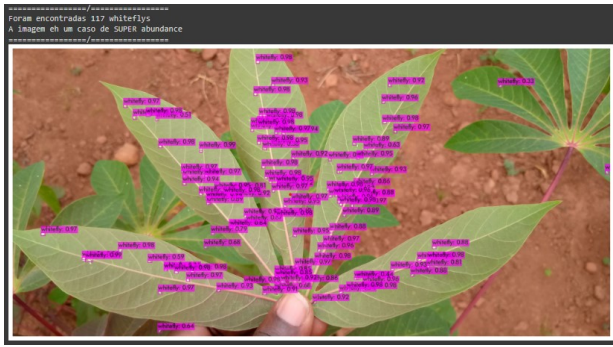


Fig. 2. Exemplo de predição gerada pela YoloV4, Moderate abundance

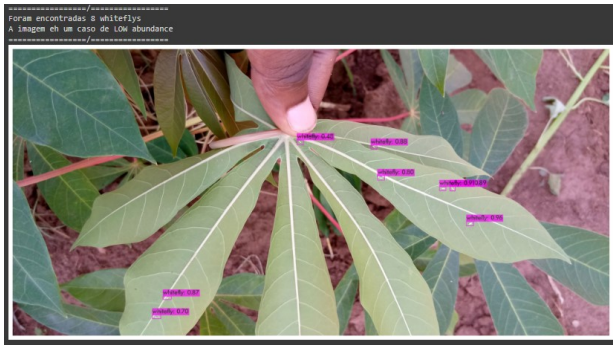


Fig. 3. Exemplo de predição gerada pela YoloV4, Moderate abundance

Em comparação com os resultados obtidos no artigo [1], temos a tabela IV:

Podemos observar que apesar de boas predições realizadas em imagens, a YoloV4 ainda apresenta uma alta taxa de erros. São varias predições corretas, os positivos verdadeiros (TP), mais até mesmo que Faster-RCNN e Haar Cascade, mas também são altas as taxas de erros, com falsos negativos (FN) e falsos positivos (FP), por isso, a YoloV4 apresenta valores de precisão, revogação e F1-score inferiores.

Portanto, é argumentável que a rede YoloV4 treinada está "chutando" muito e por isso acertando bastante, isso resulta em um alto número de TP, porém, subsequentemente aos acertos também ocorrem erros com frequência semelhante ou superior, o que implica em altos FN e FP. Isso demonstra a ocorrência de uma grande variância na rede, refletida nos valores de Precision, Recall e F1-Score, que são inferiores aos das outras técnicas.

TABLE I
TABELA DE COMPARAÇÃO ENTRE AS TRÊS TÉCNICAS

	Haar Cascade	Faster-RCNN	YoloV4
TP	3950	5627	6224
FP	831	140	7139
FN	2997	1330	8239
Precision	0.83	0.98	0.47
Recall	0.57	0.81	0.43
F1-score	0.68	0.89	0.45

V. CONCLUSÃO

Tendo em vista os Resultados do experimentos anteriores, podemos apenas observar o potencial da YoloV4. Apesar de rápida em suas predições, em média 33 milissegundos por imagem, é evidente que a YoloV4 apresentada apenas reflete pouco do potencial da técnica, sendo necessário o aprimoramento dos parâmetros de treinamento e possivelmente mais tempo de treinamento para rede.

REFERENCES

- [1] J. F. Tusubira, S. Nsumba, F. Ninsiima, B. Akera, G. Acellam, J. Nakatumba, E. Mwebaze, J. Quinn, and T. Oyana, "Improving in-field cassava whitefly pest surveillance with machine learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.