Uso de redes neurais convolucionais para detecção de laranjas no campo

João Camargo Neto[1], Sônia Ternes[1], Kleber Xavier Sampaio de Souza[1], Inácio Henrique Yano[1] , Leonardo Ribeiro Queiros[1]

[1] Embrapa Informática Agropecuária, Campinas, São Paulo, Brasil,

joao.camargo@embrapa.br, sonia.ternes@embrapa.br, kleber.sampaio@embrapa.br,

inacio.yano@embrapa.br, leonardo.queiros@embrapa.br

RESUMO

A laranja e seus derivados são um dos principais produtos do agronegócio brasileiro, além de uma das cadeias produtivas que mais emprega mão de obra por hectare, o que mostra o alto grau de impacto econômico e social desta cultura para o país. Uma estimativa de produção eficiente pode auxiliar os produtores tanto no manejo de sua lavoura quanto na adoção de estratégias de vendas com a indústria. Este trabalho descreve o processo de treinamento e teste de uma rede neural de aprendizado profundo para a detecção e contagem de frutos verdes a partir de imagens digitais de pés de laranja obtidas no campo. Os resultados para as imagens de teste apresentaram índice de mais de 90% de precisão, com cerca de 90% de revocação para a rede neural. Isso indica que a metodologia utilizada é bastante promissora.

PALAVRAS-CHAVE: Visão computacional, Aprendizado profundo, Yolo-v3, Citros.

ABSTRACT

Oranges and derivatives are one of the main products of Brazilian agribusiness and one of the sectors that employs more labor force per hectare, which shows the high degree of economic and social impact of this crop for the country. An efficient yield estimation can assist producers in managing their crop and in adopting sales strategies with industry. This work describes the process of training and testing a convolutional neural network for detecting and counting green fruits from digital images of orange trees obtained in the field. The results for the test images showed an index of more than 90% precision, with about 90% recall to the neural network. This indicates that the methodology used is very promising.

KEYWORDS: Computer vision, Deep learning, Yolo-v3, Citrus.

INTRODUÇÃO

Segundo o Levantamento Sistemático da Produção Agrícola (IBGE, 2019) a área plantada de laranja em 2018 foi de 605.752 ha e a estimativa para 2019 é de 594.541 ha, a produção em 2018 foi de 16.677.091 t e estimativa de 16.730.652 t para 2019.

Esses números demonstram a importância econômica e social que a citricultura tem para o país, como também relatado por Neves e Trombim (2017), que verificaram que em 2017 a citricultura foi responsável por gerar cerca de 200 mil empregos diretos e indiretos, além do setor gerar um PIB de US$ 6,5 bilhões de dólares em todos elos de sua cadeia produtiva. Por ser uma atividade que exige grande quantidade de mão de obra, manejos de insumos e superação de desafios agronômicos, a citricultura tem um terreno fértil para desenvolvimento e uso de tecnologias, dentro da filosofia da Agricultura de Precisão, que permitam automatizar e otimizar tarefas dentro do cultivo de plantas de laranjas, buscando melhorar a eficiência de produção e retorno econômico.

Diferente de outras *commodities* como milho e soja, a produção da laranja não pode ficar estocada na propriedade aguardando melhores condições de comercialização. Por esse motivo, uma estimativa de produção eficiente pode auxiliar os produtores em estratégias de vendas com a indústria. Um grande problema é o alto custo dessa estimativa que atualmente é intensiva em mão de obra e tempo.

Buscando contribuir para solucionar essa problemática enfrentada pelos citricultores, o objetivo deste trabalho é usar técnicas de visão computacional baseadas em redes neurais de aprendizado profundo para detecção e contagem de laranjas verdes.

A utilização de redes neurais de aprendizado profundo para identificação e contagem de frutos visíveis na copa da árvore pode ser o primeiro passo para a construção de um sistema robusto de estimativa do número de frutos no pomar. As redes neurais de aprendizado profundo vem sendo utilizado com sucesso em diversos tipos de aplicações. Na área agrícola para detecção de frutas utilizando a rede neural convolucional - Faster R-CNN (SA *et al*., 2016), para a contagem do número da panícula por unidade de área em experimentos de melhoramento genético de sorgo (GHOSAL *et al*., 2019), na detecção e classificação (gramíneas e folhas largas) de plantas daninhas na cultura de soja (FERREIRA *et al*., 2017). Em Tian *et al*. (2019) relata-se o uso de rede YOLO-V3 para identificação de maçãs em vários estádios de desenvolvimento da fruta para estimativa de produção. Já em Koirala *et al*. (2019b) são feitas identificações de mangas utilizando-se diversas redes neurais convolucionais, entre as quais uma versão modificada da YOLOv3, denominada MangoYOLO, todas as redes testadas obtiveram resultados superiores a 90% de taxa de

acerto, a partir de imagens noturnas com iluminação artificial, este recurso foi utilizado para minimizar os efeitos das variações na luminosidade nas fotos, causadas pela iluminação natural.

Apesar de existirem alguns artigos na literatura relatando a eficiência do uso de redes neurais convolucionais de aprendizado profundo em aplicações agrícolas, ainda existem vários desafios a serem superados para disponibilizar esta tecnologia em aplicações reais no campo. A metodologia aqui relatada busca avaliar o potencial do uso de rede neurais para detecção de frutos verdes a partir de fotos de árvores tomadas em condições reais de campo, sem preocupação com iluminação específica, tipo de equipamento ou resolução das imagens.

MATERIAL E MÉTODOS

Sistema de detecção usando as redes convolucionais YOLO

O sistema usado neste trabalho para a detecção de frutos diretamente a partir de imagens coletadas em campo foi um tipo de rede neural convolutiva batizada de YOLO (*You Only Look Once*) pelo seu criador Joseph Redmon (REDMON, 2016). Em particular, usou-se uma versão modificada da YOLO-v3 (REDMON; FARHADI, 2018), que ao invés das 53 camadas convolucionais originais, possui 75. A rede possui ainda outras 31 camadas dos tipos *shortcut*, *yolo*, *route* e *upsample*, perfazendo 106 camadas.

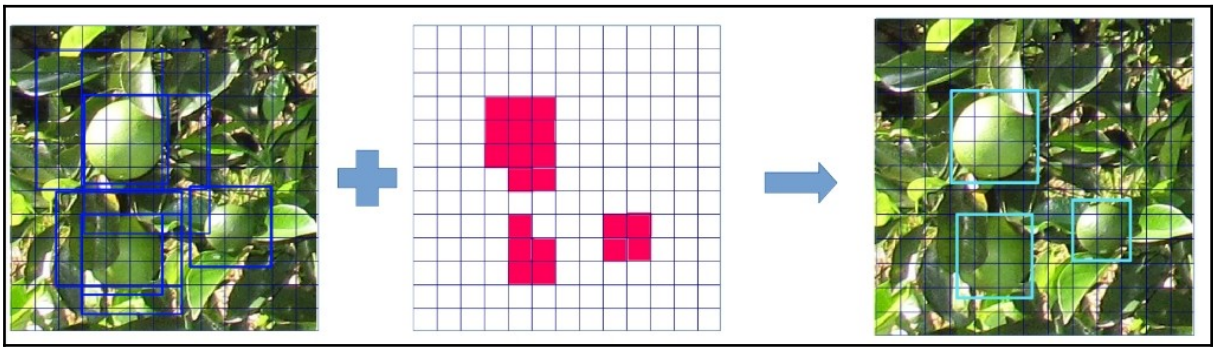
Existem dois repositórios de onde podem ser baixados os fontes da YOLO-v3 para serem compilados, um deles desenvolvido por Joseph Redmon (REDMON, 2019) e outro, que é um *fork* deste primeiro, desenvolvido por Alexey A. B. (ALEXEY, 2019). Preferimos usar esta última implementação, pois ela apresenta várias melhorias em relação à original, tais como a melhora no desempenho da detecção, redução no tempo de treinamento, otimização da alocação de memória e cálculo de medidas de desempenho dos modelos, dentre outras. Estes cálculos foram essenciais para produzirmos os gráficos da próxima seção.

O tamanho do *kernel* de detecção é determinado pela fórmula N x N x (B x (5+C)), onde N é a dimensão da imagem na última camada de convolução, B é o número de *bounding boxes*, e C é o número de classes. O número 5 vem do fato de que para cada ponto onde ocorre a detecção, tem-se cinco medidas: as quatro coordenadas de cada *bounding box* e um valor de probabilidade de haver um objeto naquela área. Como temos uma única classe (laranja) no projeto e usamos 3 *bounding boxes*, o dimensionamento do *kernel* de detecção é N x N x 18. O número N é variável conforme a escala de detecção. Para melhorar a precisão, o YOLO v3 implementa 3 escalas. A menor escala ocorre quando a rede realizou sua subamostragem máxima e a imagem foi reduzida em 32 vezes, de 416 x 416 para 13 x 13.

Este detector fica na 82ª camada e tem dimensão 3.042 (13 x 13 x 18). Após este ponto, a rede volta a aumentar (*up-sampling*), até atingir 26 x 26 e, para melhorar o desempenho, é concatenada com a rede de mesmo tamanho lá da camada 61 e a nova detecção ocorre na camada 94, agora em uma grade de 26 x 26. Nesta detecção, o *kernel* é um tensor de dimensão 12.168. O processo se repete para a última escala de detecção, com novo aumento de 26 x 26 para 52 x 52. Novamente há uma concatenação com as camadas de mesma dimensão lá do início, na camada 36, e novas convoluções, levando a uma detecção final na 106ª camada. Para esta última detecção, o *kernel* é um tensor de dimensão 48.672.

Como exemplo de funcionamento do YOLO, na primeira detecção, com grade 13 x 13, na Figura 1 à esquerda tem-se o sistema testando todos os *bounding boxes* para cada uma das células da grade. Para cada camada de detecção existem 3 possíveis *bounding boxes*. Os *bounding boxes* com maior probabilidade de haver um objeto de interesse naquela região recebem valores maiores. O sistema também estabelece a probabilidade de haver um fruto naquela região, ilustrado na parte central da Figura 1. Por fim, o sistema aplica a regra Bayesiana P(fruto) = P(objeto) x P(fruto/objeto). Toda convergência é guiada pela otimização de uma função de perda, cujos detalhes podem ser encontrados na literatura.

Figura 1 – Funcionamento da rede neural convolucional YOLO



Fonte: gerada pelo Autores

Aquisição e tratamento das imagens

O sistema foi treinado com 2.035 imagens, utilizando placa gráfica GPU, divididas em 1.832 para treino e 203 para validação (10% do conjunto de treinamento) durante a etapa de treinamento. Para teste das redes treinadas usou-se um outro conjunto de dados com 1.030 imagens. Cada uma dessas imagens tem dimensão de 416x416 pixels e foram recortadas das imagens originais com maior tamanho. As imagens originais foram capturadas usando diferentes dispositivos e resoluções, tais como câmeras fotográficas e celulares, sendo sua grande maioria fornecida pelo Programa de Estimativa de Safra (PES) do Fundo de Defesa da Citricultura (Fundecitrus). A diversidade de meios de captura se justifica por querermos testar

o sistema em uma situação real de campo, sem impor condições específicas de equipamentos, de resolução ou de condições de iluminação. Adicionalmente, os frutos capturados nas imagens são de diferentes variedades de laranjas e, embora predominantemente verdes, também representam vários graus de maturação.

Anotação das imagens

Após recortadas, cada uma das imagens foi manualmente anotada e verificada por usuários humanos em um sistema desenvolvido para este fim. Neste sistema desenha-se um retângulo ao redor da área que se identifica como um fruto ou parte visível deste. A Figura 2 ilustra duas imagens reais com 5 e 4 frutos anotados, respectivamente. Como se pode perceber, a primeira imagem está muito mais iluminada que a segunda, situação em que o reflexo da luz do sol tanto nas laranjas quanto nas folhas pode confundir o sistema de identificação neural. Outro complicador é que existem frutos em tons de verde mais escuro localizados em região de sombra (mais escura) e frutos parcialmente oclusos pela folhagem.

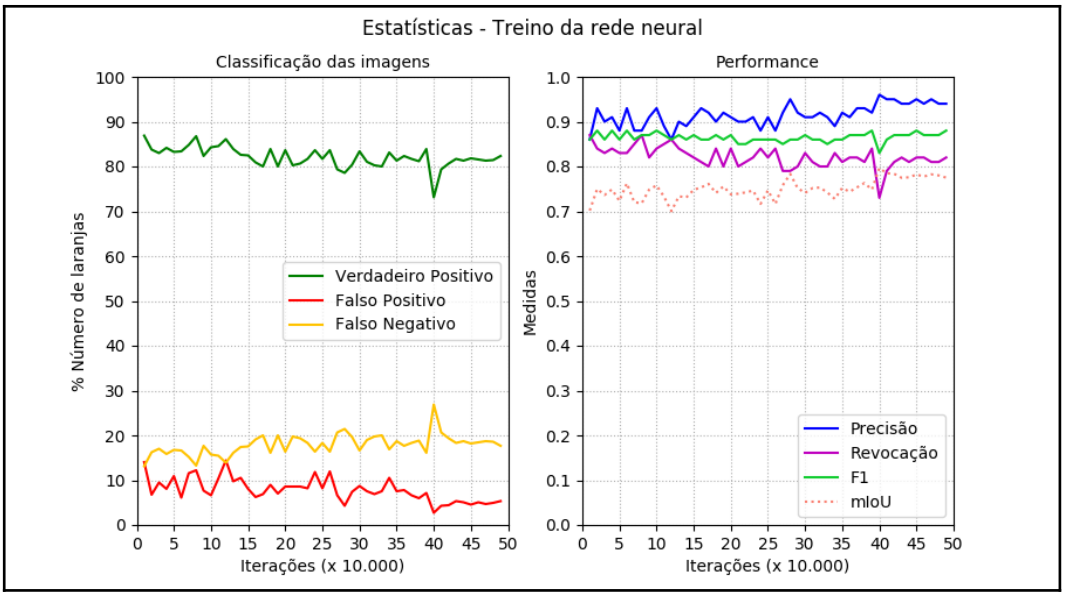
Figura 2 – Exemplo de duas imagens da base de treino com os frutos anotados

Fonte: gerada a partir da versão original cedida pelo PES/Fundecitrus

RESULTADOS E DISCUSSÃO

Durante o treinamento (base contendo 2.035 imagens) foram executadas 490.000 iterações da rede neural, permitindo observar por meio de gráficos diversos aspectos do comportamento dos modelos. A cada 1.000 iterações o modelo foi salvo, ou seja, foram geradas 490 redes neurais para serem testadas. Na Figura 3 são apresentadas as estatísticas de desempenho dessas redes frente à base de validação (203 imagens contendo 770 frutos anotados) durante a etapa de treinamento.

Figura 3 – Estatísticas obtidas durante a fase de treino da rede neural



Fonte: gerada pelo Autores

No gráfico à esquerda são apresentadas as porcentagens de frutos identificados corretamente (verdadeiro positivo ou VP), frutos não identificados pela rede (falso negativo ou FN) e partes de folhas/galhos identificados erroneamente como frutos (falso positivo ou FP). Ao final do treinamento obteve-se a identificação correta de mais de 80% dos frutos existentes na base.

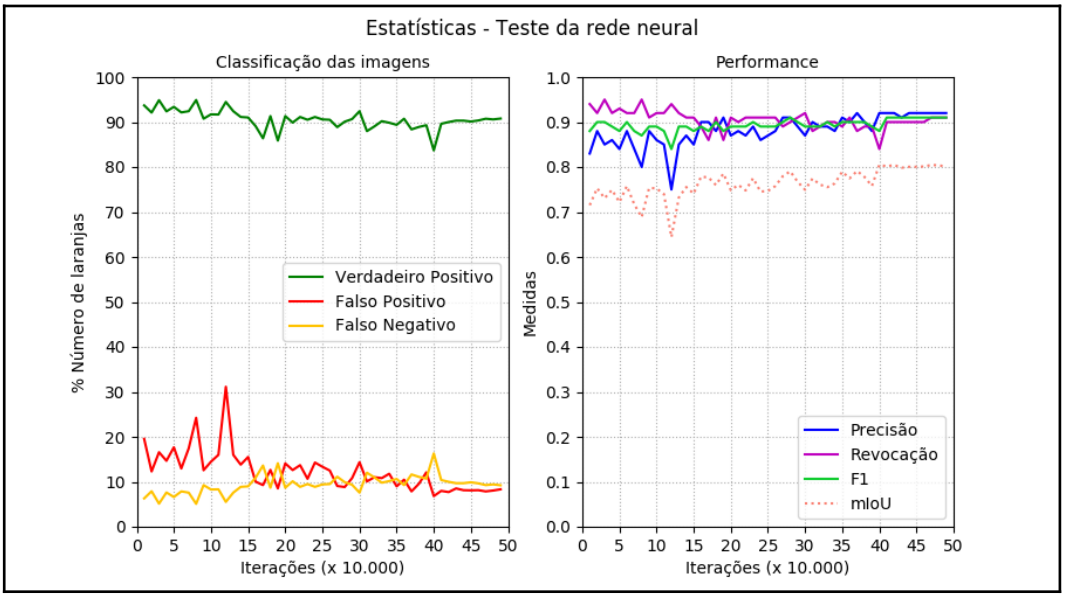
No gráfico à direita são apresentadas as curvas relativas à performance da rede. A curva azul indica a precisão do modelo, dada pela fração VP/(VP+FP), ou seja, quantas detecções corretas o modelo produziu. A curva roxa relaciona-se à métrica de revocação que é calculada pela fração VP/(VP+FN), indicando a frequência com que o modelo identifica corretamente uma laranja. Observando o gráfico percebe-se uma tendência de melhora na precisão, com pequena queda na revocação. Em geral, com o aumento da precisão ocorre uma perda na revocação, pois quanto mais preciso o sistema se torna em identificar a classe alvo (laranja), menos ele detecta alguns objetos que tenham menor probabilidade de serem verdadeiros positivos. Quem estabelece o compromisso entre a precisão e a revocação é a medida F1 (curva verde), dada por: F1 = 2 * (Precisão * Revocação)/(Precisão + Revocação). Percebe-se que essa medida, após a estabilização, teve uma melhoria por volta de 450.000 iterações. Por fim, percebe-se uma estabilização a partir de 400.000 iterações do valor da métrica *mean intersect over union* ou mIoU (curva pontilhada rosa), que fornece a medida de que há um objeto de interesse nas várias regiões identificadas pelos *bounding boxes*.

Ressalta-se que as quedas em todas as curvas na iteração 400.000 se dá pela mudança automática no valor do passo do gradiente de otimização da função de perda citada na seção

anterior. Após um ajuste inicial, a rede retoma seu equilíbrio.

Na Figura 4 estão apresentados os resultados das estatísticas obtidas do processo de detecção para a base de teste (1.030 imagens não usadas no treinamento).

Figura 4 – Estatísticas obtidas para a fase de teste da rede neural



Fonte: gerada pelo Autores

Observando as curvas do gráfico à esquerda, embora os modelos obtidos com menor número de iterações tenha uma alta porcentagem de laranjas identificadas (verdadeiro positivo), há ainda um número expressivo de falsos positivos. A situação melhora quando se testa os modelos obtidos com maior número de iterações, pois há um aumento da precisão sem implicar numa queda substancial na revocação, conforme mostrado no gráfico à direita. Por exemplo, a partir de 200.000 iterações, quando os modelos tendem a ser mais estáveis, e em especial, após 400.000 iterações, há uma queda considerável da porcentagem de falsos positivos e falsos negativos, com valores de precisão e revocação acima de 0,9.

O modelo correspondente à iteração 450.000, escolhido com base nos resultados mostrados na Figura 4, foi utilizado para teste da detecção dos frutos em imagens contendo uma árvore inteira. Para este modelo, os valores alcançados nas estatísticas da Figura 4 são: precisão = 0,95; revocação = 0,82; F1 = 0,88; mIoU = 78,2%. A Figura 5 mostra a imagem usada no teste, que possui tamanho original de 2.068x3.333 pixels.

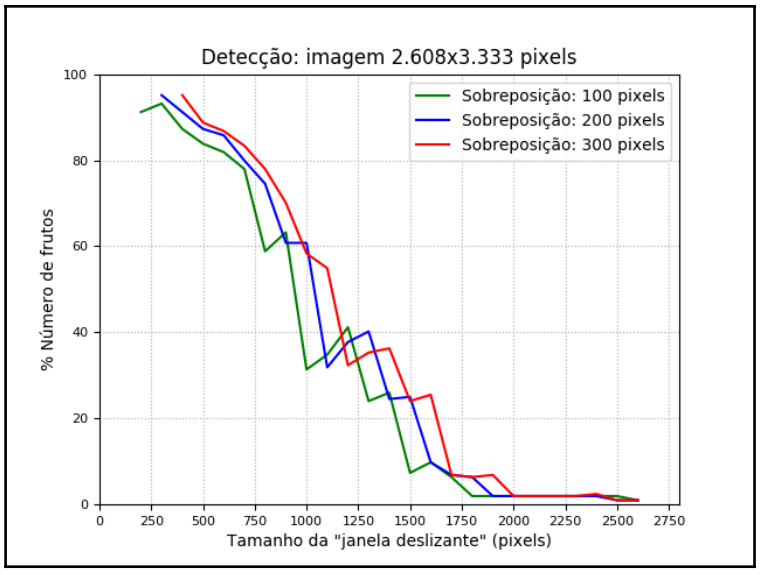
Um experimento foi realizado utilizando uma janela deslizante, que percorre toda a imagem. A Figura 6 mostra o resultado para diversos tamanhos de janela deslizante. No eixo das abcissas é apresentado o tamanho da janela deslizante e no eixo das ordenadas a porcentagem de frutos detectados com relação ao total real de 206 frutos contidos na imagem.

Figura 5 – Imagem original usada para testes de detecção de frutos por planta

Fonte: PES/Fundecitrus

Como pode ser observado, conforme o tamanho dessa janela tende às dimensões da entrada da rede (416x416 pixels), mais frutos são detectados. Isto se deve ao fato que imagens com dimensões muito maiores que a configuração de entrada da rede são reduzidas, e neste processo de redução o objeto a ser detectado pela rede é eliminado ou reduzido a tamanhos que impossibilitam a sua identificação. Quando é apresentada à rede uma imagem maior que 1.700x1.700 pixels praticamente não é detectado nenhum fruto. Para imagens menores que 1.500x1.500 pixels a rede já começa a detectar frutos, sendo que a maioria dos frutos são detectados com dimensões de imagens próximas da configuração da rede.

Figura 6 – Porcentagem de detecção de frutos em função do tamanho da janela deslizante



Fonte: gerada pelos Autores

A Figura 7a mostra o resultado da detecção dos frutos utilizando uma janela deslizante de 416x416 pixels com uma sobreposição entres janelas de 300 pixels. Nesta árvore os frutos estão totalmente verdes e há uma maior quantidade de frutos na parte interna da árvore, como mostrado em detalhes na Figura 7b, dificultando a detecção devido à oclusão. Neste exemplo foram detectados 208 frutos, nenhum falso positivo e 21 falsos negativos.

Figura 7 – Resultado da detecção de frutos verdes (a); e frutos na parte interna da árvore (b)

(a)

(b)

CONCLUSÕES

Os resultados descritos mostram o alto potencial de uso da rede neural de aprendizado profundo aqui descrita para a identificação de frutos verdes em pés de laranja. Sabe-se entretanto, que a variedade do pé de laranja também tem influência no resultado geral da detecção, pois algumas espécies têm a folhagem mais intensa, aumentando a oclusão de frutos. Assim, pretende-se buscar uma melhora nos resultados aumentando o número de imagens nas bases de treino e teste da rede neural, contemplando diferentes variedades de citros. Será avaliado também o impacto de diferentes tamanhos de *bounding boxes* em novos testes de detecção em árvores inteiras.

AGRADECIMENTOS

Agradecemos aos coordenadores do PES/Fundecitrus pela disponibilização de fotos de árvores e à NVIDIA Corporation pela doação da placa GeForce usada no processamento e teste da rede neural.

REFERÊNCIAS

Instituto Brasileiro de Geografia e Estatística - IBGE. **Levantamento Sistemático da Produção Agrícola.** https://www.ibge.gov.br/estatisticas/economicas/agricultura-e-pecuaria/9201-levantamento-sistematico-da-producao-agricola.html?=&t=resultados/ Acesso em 04 de julho de 2018.

ALEXEY A. B. Yolo-v3 and Yolo-v2 for Windows and Linux. https://github.com/AlexeyAB/darknet#yolo-v3-in-other-frameworks. Acessado em 05/07/2019.

FERREIRA, A. S.; FREITAS, D.M.; SILVA, G. G.; PISTORI, H.; FOLHES; M.T. Weed detection in soybean crops using ConvNets. Comput. Electron. Agr, v. 143, p. 314-324, 2017.

KOIRALA, A., WALSH, K.B., WANG, Z. *et al*. Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of 'MangoYOLO. Precision Agric (2019). https://doi.org/10.1007/s11119-019-09642-0

SA, I.; GE, Z.; DAYOUB, F.; UPCROFT, B.; PEREZ, T.; MCCOOL, C. Deepfruits: A fruit detection system using deep neural network. Sensors, v. 16, n. 8, 2016.

GHOSAL, S.; ZHENG, B.; CHAPMAN, S.; POTGIETERo, A.; JORDAN, D.; WANG, X.; SINGH, A.; SINGH, A.; HIRAFUJI, M.; NINOMIYA, S.; GANAPATHYSUBRAMANIAN, B.; SARKAR, S.; GUO, W. A Weakly Supervised Deep Learning Framework for Sorghum Head Detection and Counting. Plant Phenomics, v. 2019, 14p., 2019. https://doi.org/10.34133/2019/1525874.

NEVES, M. F.; TROMBIM, V. G. Anuário da citricultura 2017. 1. ed. SÃO PAULO CITRUSBR, 2017. 60 p.

REDMON, J. *et al*.You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, Las Vegas. IEEE: p. 779-788, 2016.

REDMON, J.; FARHADI, A. YOLOv3: An incremental improvement. arXiv. Technical Report preprint arXiv:1804.02767, 2018.

REDMON, J. Darknet: Open Source Neural Networks in C. https://pjreddie.com/darknet/ Acessado em 05/07/2019.

TIAN, Y.; YANG, G.; WANG, Z.; WANG, H.; LI, E.; LIANG, Z. "Apple detection during different growth stages in orchards using the improved YOLO-V3 model," Computers and Electronics in Agriculture, vol. 157, pp. 417–426, 2019.