Revista Brasileira de Ciências Agrárias
ISSN (on line) 1981-0997
v.17, n.2, e1353, 2022
Recife, PE, UFRPE. www.agraria.pro.br
DOI: 10.5039/agraria.v17i2a1353 - Protocol 1353
Subm. 25/08/2021 • App. 17/04/2022 • Pub. 20/06/2022

# Detection and counting of plants via deep learning using images collected by RPA

Kelly Lais Wiggers[1]*, Carlos Daniel Pohlod[1], Regiane Orlovski[2], Rodrigo Ferreira[3], Thais Amanda Santos[1]

[1] Universidade Estadual do Centro Oeste, Guarapuava, PR, Brasil. E-mail: kellyl@unicentro.br; carlospohlod@gmail.com; thais.amandas18@gmail.com
[2] Faculdade Guairacá, Guarapuava, PR, E-mail: rorlovski@unicentro.br
[3] Fundação Agrária de Pesquisa Agropecuária, Guarapuava, PR, Brasil. E-mail: rfagronomy@hotmail.com

**ABSTRACT:** Plant counting and location are essential to provide better control and production estimates in agricultural regions. Techniques based on deep learning have promising results in several application domains, including image analysis collected by RPA. This paper proposes the use of a deep learning model to detect and count plants in RGB images acquired by an unmanned aerial vehicle. The results were obtained via the YOLO model, with validation performed on manually annotated images. The experimental results of the trained model, considering an overlap greater than or equal to 50%, had an average precision of 84.8% and a recall of 89% for images where training and tests were performed in the same field. Experiments were also carried out with the trained model on images from different regions of the training, demonstrating effective results in detecting plants.

**Key words:** agriculture; production estimation; RGB images; YOLO

# Detecção e contagem de plantas via aprendizagem profunda usando imagens coletadas por RPA

**RESUMO:** Contagem de plantas e localização são essenciais para proporcionar melhor controle e estimativas de produção em regiões agrícolas. Técnicas baseadas em aprendizagem profunda tem se destacado em diversos domínios de aplicação, incluindo análises em imagens coletadas por RPA. Este artigo propõe a utilização de um modelo de aprendizado profundo para detectar e contar plantas de feijão em imagens RGB adquiridas por um veículo aéreo não tripulado. Os resultados foram obtidos via modelo YOLO, com validação realizada em imagens anotadas manualmente. Os resultados experimentais do modelo treinado, considerando sobreposição maior ou igual a 50%, teve precisão média de 84.8% e recall de 89% para imagens onde o treinamento e os testes foram realizados no mesmo campo. Também foram realizados experimentos com o modelo treinado em imagens de regiões diferentes do treinamento, demonstrando resultados efetivos na detecção de contagem de plantas.

**Palavras-chave:** agricultura; estimativa de produção; imagens RGB; YOLO

* Kelly Lais Wiggers - E-mail: kellyl@unicentro.br (Corresponding author)
Associate Editor: Edna Maria Bonfim Silva

## Introduction

The recognition of objects on the physical surface is important in different areas of knowledge, as well as for various purposes in agriculture, which, with the integration of mathematical and statistical techniques, makes it possible to discriminate features using remote sensing images. New technological methods based on Multirotor and Fixed Wing (RPA) leverage the precision agriculture approach that includes crop monitoring that provides farmers with real-time data on plant health and crop spraying chemicals in the field. This innovative approach can help farmers save their crops and maximize yields from their fields (Pederi & Cheporniuk, 2015).

Several researchers have explored alternative image based approaches. Geotechnologies emerge as a prominent tool for enforcement, resource analysis, and environmental monitoring. Studies focused on planting monitoring, planting estimates (Rahnemoonfar & Sheppard, 2017), weed detection (Milioto et al., 2017; Ampatzidis & Partel, 2019), or even plant counts (Karami & Crawford, 2020) stand out.

Machine learning based technologies are increasingly prominent in agriculture oriented applications, and are favorable in a range of applications, with significant potential for agriculture (Castro et al., 2018; Karami & Crawford, 2020). Noteworthy are works using Convolutional Neural Networks (CNN) (Kalantar et al., 2020; Valente et al., 2020; Xu et al., 2020), and approaches that propose more effective performance, such as R-CNN, applied by Ho et al. (2019) in counting watermelon stalks and Neupane et al. (2019) for counting banana stalks. The YOLO (You Look Only Once) algorithm, has recently been used in research for tree detection (Ampatzidis & Partel, 2019), cotton counting (Oh et al., 2020), showing good performance.

However, using aerial imagery via RPA to perform estimations, plant counts, or even general classifications, and considering the use of technologies that require several process runs or even integrate several types of algorithms to find feasible solutions, can become expensive and challenging. In this regard, it is important to use approaches that involve few processing steps and still result in good performance.

Thus, in this paper a method is proposed for detecting and counting plants in aerial images captured by RPA using a deep learning approach. To this end, a convolutional neural network based on the YOLO algorithm was tuned, responsible for the steps of candidate object generation, feature extraction, and object detection. In addition, the results are evaluated and the count of correctly detected objects is presented. Several experiments have been performed to determine the parameters for the YOLO algorithm, defining different types and formats of images. The contributions of this work were: object detection in images of agricultural areas, model for setting parameters to adjust the network for object detection in different types of images.

## Materials and Methods

### Study area

The study site is located in the municipality of Guarapuava, in the central-western region of the state of Paraná, Brazil, comprising an agricultural region with several types of crops. The agricultural area containing the bean crop was installed within the experimental field of the Fundação Agrária de Pesquisa Agropecuária (FAPA) on February 26, 2021, in a direct sowing system over straw. Sowing was done on 25/02/2021 and the emergence date was 02/03/2021. Figure 1 illustrates the location of the study area. It shows the map of Paraná, focusing on the Guarapuava region and the location of FAPA experimental field.

The images were captured by the FAPA company from Guarapuava on March 19, 2021, at 3:00 p.m. Due to the partnership with the research project, the images were made available for this study. In total, 68 images were captured via RPA of a region with bean planting, at a height of 20 m and focal length of 10.26 mm. The RPA (DJI Mavic Pro quad rotor)

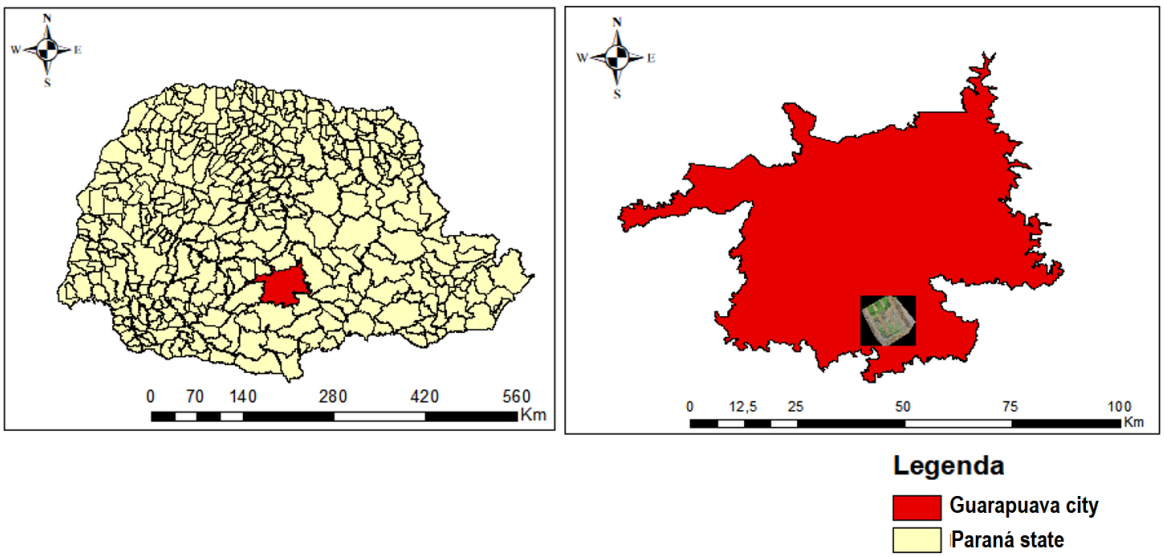



**Figure 1.** Identification of the study area.


**Figure 2.** Examples of images captured by RPA in the study area.

and the Camera are the property of FAPA. The camera has the following basic parameters: model Hasselblad L1D-20c CMOS 1/2.3", effective pixels: 12.35 Mb, FOV 78.8° and 26 mm f/2.2 lens, distortion < 1.5%, focus from 0.5 m to ∞. Photographic data: aperture 1/320, f/5, focal length (35 mm): 28.

In addition, these images have 3 spectral bands, Red, Green and Blue. Figure 2 has examples of the captured images that can be used for labeling. In addition, they highlight some difficulties of the collection points, since there is variability in plant growth in some regions that were captured.

In the planting regions captured by the imaging, the crop was 17 days old. However, some variability is observed in some regions. It is possible to observe in Figure 2A, a good object detection region. Figures 2B and 2C have examples of noise that make identification difficult. For example, Figure 2C has planting rows that are in an advanced stage of growth, as the imagery also captured some areas of the contour of the study region. This makes it impossible to identify feet separately to perform the count estimation. In addition, Figures 2B and 2C have regions that do not characterize the objects that define the research objective.

## Orthomosaic image

To enable the visualization of the whole area where the images were captured, considering the height 20 m and overlapping 70% frontal and lateral, it was necessary to create an orthomosaic image. An image in orthomosaic format is a map composed of several orthophotos joined together, and can carry various information about the region of interest. The orthomosaic image shown in Figure 3 was generated from the 68 images captured by the RPA using the test version of the pixel4D software (https://www.pix4d.com). The mosaic parameters were: Ground Sample Distance (GSD) of 0.19 cm/pixel, file size 19,235 × 17,420 pixels, 96 dpi, size 611.2 mb in geotiff format, with total area 0.0757 ha.

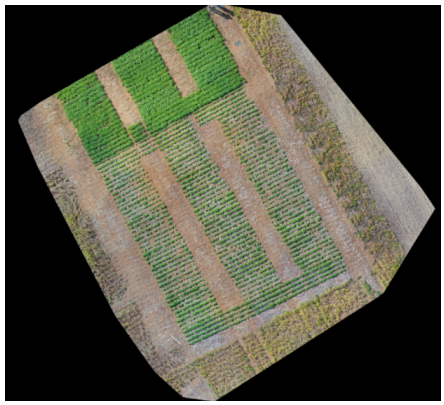


**Figure 3.** Orthomosaic image of the images captured via RPA.

## Overview of the proposed model

As can be seen in Figure 4, the first step for training was the selection of images captured from the RPA to be used in

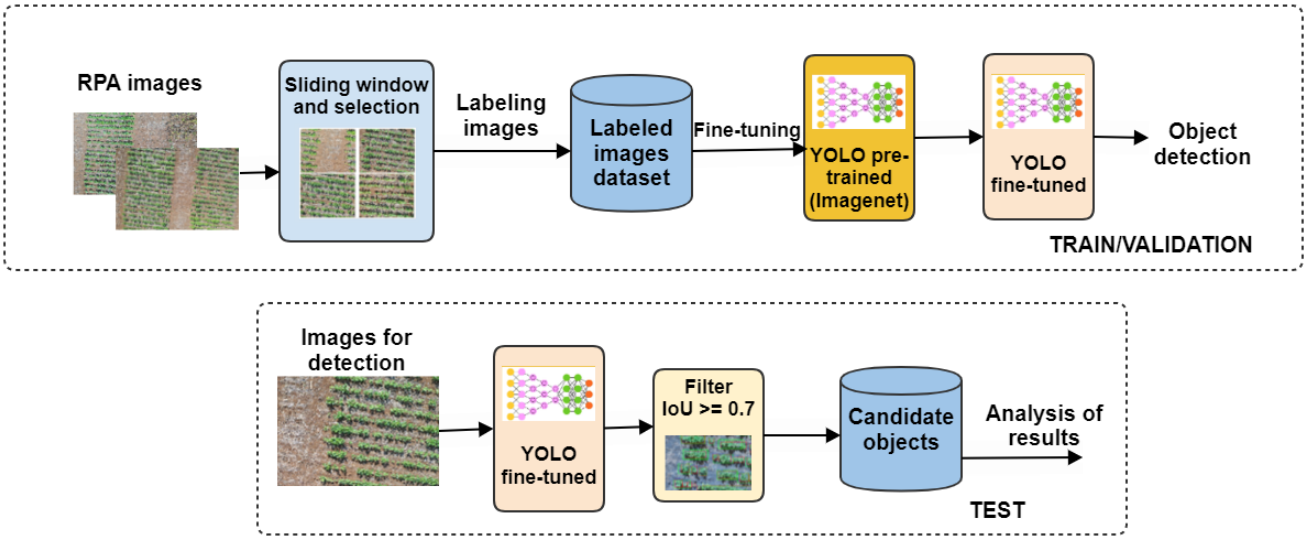



**Figure 4.** Overview of the proposed model for object detection based on YOLO architecture.

training the neural network. The images were taken at 3:00 p.m. by the RPA and are large in size, 5,472 × 3,648 pixels, totaling approximately 14.2 Mb. In the image base, some shading of the plants is observed, but evenly across all images. This image base with this time was chosen because, of the 11 flights made (mainly between 12:00 and 2:00 p.m.) this was the only one totally cloud-free. The images were worse on the earlier flights because of the alternating light and shadow due to the high cloudiness. A perfect flight was performed at 12:25 p.m. on 06/04/2021, completely cloudless, but at too advanced a crop stage to enable AI training.

After selection, the images were partitioned into quadrants for selection of images to be labeled. After labeling the objects, a pre-trained YOLO model network was used, and a fit (fine-tuned) was performed to the database parameters and images. Thus, the model returns the objects that have been detected.

In the testing stage, as can be seen in Figure 4, the model trained in the previous stage is used to identify objects from images that were not used in training, in order to observe the generalization of the model. Thus, detected images are selected according to an IoU threshold, which, for this project, includes comparisons with IoU 0.5, 0.6, and 0.7. At the end, the results are analyzed and discussed.

### Image partitioning and labeling

For each image a partitioning into 4 regions, called quadrants, was initially performed, and some quadrants were chosen for use in this project. An example of the partitioning can be seen in Figure 5, where Q1, Q2, Q3, and Q4 are the representation of the quadrants generated in each image. After partitioning, each image was saved and named by the pattern: "dji_nomeimagem_quadranteX.jpg", where X is the quadrant number.

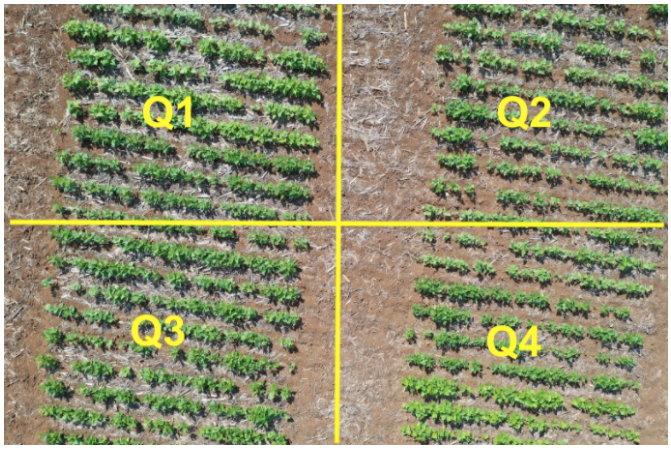



**Figure 5.** Example of partitioning to generate image quadrants.

To create the base with labeled images, 14 images were selected and annotated from the RPA image base. In this selection images that did not contain information relevant to the project were discarded. That is, images that contained only exposed soil or other types of plantings, as well as images that present regions with high growth stage of each plant, making it impossible to label the objects separately. Examples of discarded images are shown in Figure 6.

The selected images were partitioned to 2,736 × 1,824 size and labeling was done using LabelImg software (https://github.com/tzutalin/labelImg). Two types of labeling were used to identify the bean stalks, the first (Figure 7A) being that they are single and the second (Figure 7B) for stalks that are multiple (contain 2 or more stalks together but still make labeling possible). This approach was necessary due to the existing differences in aspect ratio of each object, and therefore creating a lot of confusion among the objects. Examples of these objects are shown in Figure 7.

For the training phase, a total of 636 bounding boxes were labeled manually, with 405 (64.7%) being separated for training and 231 (36.3%) for testing. The training percentage is not accurate because these bounding boxes are inside the quadrant images, so the values are not accurate. This labeling was performed by the research team of the present project. In addition, 190 bounding boxes were selected to validate the model (without duplicating from the training images). For each annotation the pattern "class x y width height" was adopted, saved in .txt format, as shown in Figure 8.


**Figure 7.** Examples of samples collected and labeled.

### YOLO based object detection

Object detectors with good performance have excelled at using deep learning algorithms. Usually these models are pre-trained on the Imagenet database and then adjusted to

**Figure 6.** Examples of images from the base that were discarded.

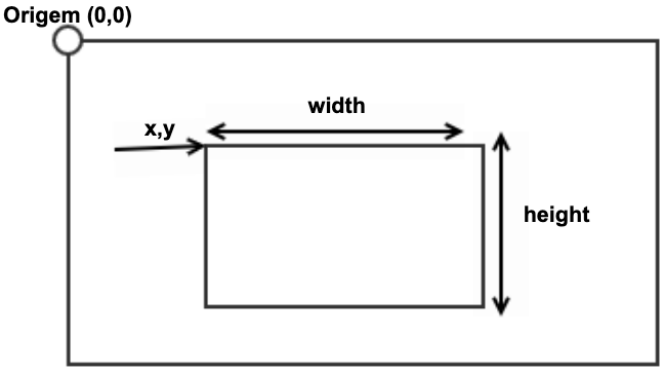



**Figure 8.** Object labeling standard.

perform the predictions and detect image bounding boxes. The YOLO algorithm has been used by some authors to detect objects in agricultural images. Noteworthy is the research of Ampatzidis & Partel (2019), who used CNN for tree detection to assess phenotypic traits in citrus crops, with an accuracy of 98%. Oh et al. (2020), on the other hand, applied the YOLO method in order to evaluate cotton plant yield estimation and foot count. They performed several experiments, with the best result being 88.16%. Therefore, the YOLO model allows with network training to receive an image as input and provide a prediction of bounding boxes and the class labels as output.

For this network architecture, an input image is divided into several grid cells, where each cell has the function of predicting a bounding box. Each cell is represented by vectors, with the coordinates height, width, x-position and y-position.

There are many variations of YOLO implementation that have been developed by researchers, and recently, the YOLOv3 and YOLOv4 versions have shown promising results.

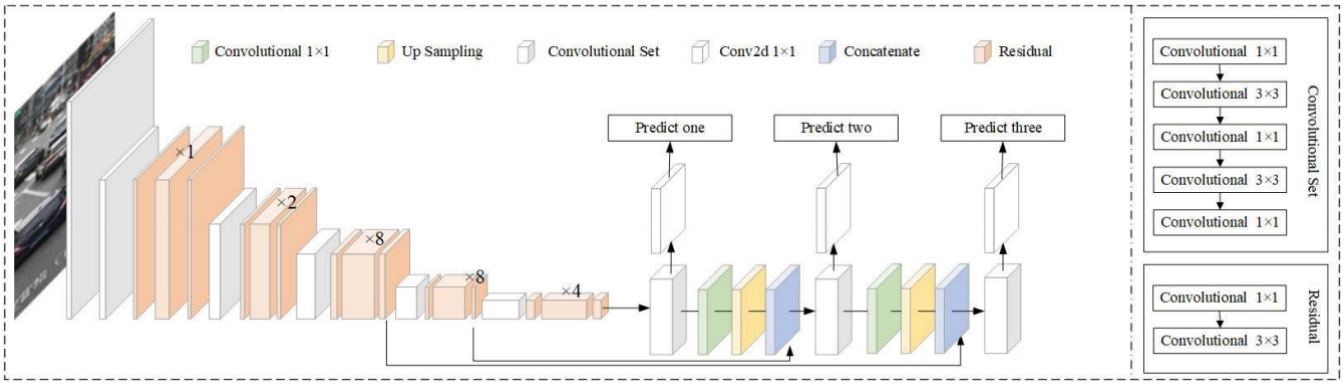
YOLOv3 (Redmon & Farhadi, 2018) is an object detector that takes the detection procedure as a regression task. This method increases the detection speed and accepts input images with different sizes. YOLOv3 uses Darknet-53 (Wang et al., 2020) to perform feature extraction. In addition, the model uses multi-scale prediction, which means that feature maps are detected at various scales. For this reason, the accuracy of object detection is increased. The detail of its structure is shown in Figure 9.

The testing process of YOLOv3 is defined in Table 1 (Mao et al., 2019).

The YOLOv4 architecture (Bochkovskiy et al., 2020) is an improvement on the YOLOv3 algorithm. It also includes the modules:

- backbone: model CSPDarknet53 (Wang et al., 2020);
- neck: Spatial Pyramid Pooling (Kaiming et al., 2015) and Path aggregation network (Liu et al., 2018); and,
- head: YOLOv3 (Redmon & Farhadi, 2018).

For the present study, the models YOLOv3 and YOLOv4 were used as models for training and making comparisons.




Source: Mao et al. (2019).

**Figure 9.** Detailed structure of YOLOv3.

**Table 1.** Stages for developing a network model based on YOLOv3.

| Stages | Description |
|---|---|
| 1 | Insert the image and scale the image to the default size. |
| 2 | Divide the input image into 13 × 13, 26 × 26, and 52 × 52 grids of three scales. If the center point of an object falls on the grid unit, the grid unit predicts the object. |
| 3 | Use k-means clustering to determine the a priori bounding box in each grid unit. There are three clusters in each grid unit. Because of the three scales, there are a total of 9 clusters per grid unit. |
| 4 | Insert the image into the grid for feature extraction. The model first produces a small-scale 13 × 13 feature map. |
| 5 | The 13 × 13 feature map is first submitted in the convolutional set and the upsampling rate is 2 times. And then connect it to the 26 × 26 feature map and produce the prediction result. |
| 6 | The 26 × 26 feature map output from step 5 is connected to a convolutional array and the upsampling rate is 2. Then connect the map to the sequential 52 × 52 feature map and show the output of the prediction result. |
| 7 | Step to merge the feature maps of the three predicted outputs. Using a probability score as a threshold to filter out most objects with low scores. It then uses Non-Maxima Suppression (NMS) (Neubeck & Van Gool, 2006) for post-processing, making the bounding boxes more accurate. |

Considering training with two types of labeling (single objects and multiple objects), since the objects have different shapes and sizes, the number of filters in the architecture was adapted to 21. The size of the input images should be an integer multiple of 32 (such as 320 × 320, 416 × 416, and 608 × 608), since in this model, each image is partitioned into 32 × 32 pixels windows to perform the feature vector extraction. The size of the images was standardized at 416 × 416 for training the models.

For training both models, a pre-trained model, darknet53.conv.74 (YOLOv3) and yolo4.conv.137 (YOLOv4), with a mini batch size of 64, max_batches of 6,000 and subdivisions of 64 on 1 GPU, a momentum of 0.9 and a weight reduction of 0.0005 were fitted. Experiments were conducted to evaluate the number of iterations to define the best model for the experiments. By default, the multi-step learning rate approach (policy = steps) was adopted with a base learning rate of 0.001, a step value of [4,800, 5,400], since in the YOLO model it indicates setting the steps to 80 and 90%, respectively, of the number of iterations. The scales defined for learning rate were [0.1, 0.1].

### Evaluation metrics

The trained network should be evaluated on a test dataset, considering different images and bounding boxes than those used for training and validation of the YOLO model. For each classified object in the image, its precise location can be evaluated by considering the Intersection over Union (IoU) metric - Equation 1. For this, the overlap between the classified object (given its location in the image) and the location defined in the label and ground truth is evaluated. The overlay considers the position of the labeled image (x, y) and its area q1. The positions of the candidate object (x1, y1) are also considered, as well as its area o1, as described in Equation 1. Commonly the value IoU ≥ 0.5 is set in object detection techniques (Nowozin, 2014). In the experiments, the IoU values were set to 0.5, 0.6, and 0.7 to evaluate the performance.

$$U_{iou}(x,y) = \frac{ql \cap ol}{ql \cup ol} \quad (1)$$

In order to statistically evaluate the experiments performed, Precision (PR) and Recall can be used for each set of objects in each class. These metrics are often used to evaluate classification and object detection performance, Xu et al. (2020) used to evaluate CNN classification performance, Sarwar et al. (2018) applied for performance in R-CNN, Kestur et al. (2018) evaluated in extreme learning machine (ELM).

The PR evaluates, out of all classified objects, how many have actually been classified correctly. Recall, on the other hand, evaluates whether all objects that should have been rated have actually been rated, i.e. the frequency of ratings. Then the AP metric can be calculated. The AP is the area under the accuracy and recall curve for each object to be identified in the image, which correspond to Equations 2 and 3, respectively (Powers, 2011):

$$\text{precision} = \frac{\text{truePositives}(\text{TP})}{\text{truePositives}(\text{TP}) + \text{falsePositives}(\text{FP})} \quad (2)$$

$$\text{recall} = \frac{\text{truePositives}(\text{TP})}{\text{truePositives}(\text{TP}) + \text{falseNegatives}(\text{FN})} \quad (3)$$

## Results and Discussion

The proposed method based on YOLOv3 and YOLOv4 was implemented using the Darknet tool (http://pjreddie.com/darknet). As mentioned, in the training stage the transfer learning mechanism was adopted. Thus, in the experiments a pre-trained network model from the ImageNet image database (Deng et al., 2009) was used and then all layers of the network were tuned using the data set of this study.

### Training and validation

To test the performance of the CNN-based deep learning network model, systematic convergence studies were conducted with respect to the number of iterations, showing some representative results when it reached 3,000 iterations. Thus, the weights from this model were saved for use in the experiments. For the same training set and test set, the count of correctly classified objects, AP, recall with the IoU values at 0.5, 0.6 and 0.7 were evaluated to observe the model performance. These results are presented in Table 2.

It can be seen from Table 1 that, given the same input image size (416 × 416) and the same training parameters, the accuracy performance of the YOLOv3 model was inferior to that of the YOLOv4 model for all IoU values. In particular, IoU results ≥ 0.5 enabled better results in the YOLOv4 model, considering 84.8% AP and 89% recall. Therefore, the model is able to identify that there is an object in the x and y coordinates of the image with at least 50% overlap of a proposed object.

Regarding object count, considering the value of IoU ≥ 0.5, the YOLOv4 model correctly returned 158 of the 231 objects that were labeled in the test images, this equates that approximately 69% of the crop was estimated in terms of object count. The YOLOv3 model, on the other hand, returned for IoU ≥ 0.5 only 103 objects in the estimate, demonstrating that several of the labeled objects were not detected.

Figure 10 shows the Precision × Recall curves for the YOLOv4 model that showed the best results of all experiments. For the YOLOv3 model, the accuracy came in between 0.8 and 0.9 with recall close to 0.2. As for recall values with 0.6, the precision was between 0.6 and 0.7. It can be seen that for the

**Table 2.** Training/validation results for object counting, AP and Recall.

| Model | IOU ≥ 0.5 | | IOU ≥ 0.6 | | IOU ≥ 0.7 | |
|---|---|---|---|---|---|---|
| | AP | Recall | AP | Recall | AP | Recall |
| | (%) | | | | | |
| YOLOv3 | 48.3 | 58 | 35.1 | 46 | 27.1 | 37 |
| YOLOv4 | 84.8 | 89 | 65.2 | 73 | 47.7 | 56 |

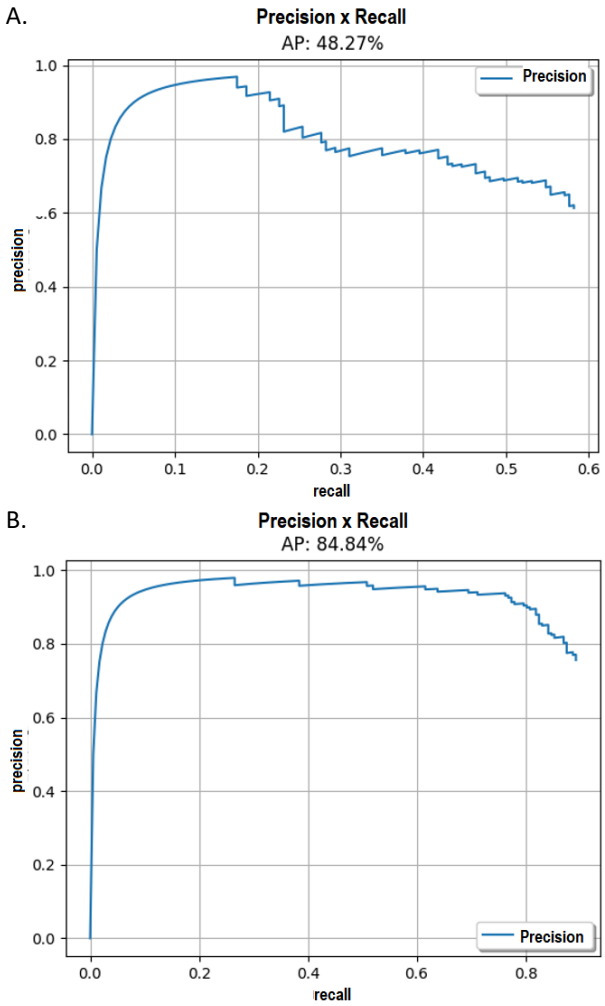



**Figure 10.** Accuracy × Recall curve for IoU ≥ 0.5.

YOLOv4 model the accuracy has in some cases come close to 1.0, but with recall between 0.2 and 0.4. For a recall 0.8, the precision is also close to 0.8.

In order to demonstrate a qualitative analysis of the detection of objects, you can see in Figure 11, the detection of some objects contained in the validation images defined in the training. It is possible to observe in Figure 11A the overlapping of the objects, where in red is the labeled image and in green the image detected by the network. It can be seen that, even with some differences in shape, the model was able to identify the presence of an object, given the features extracted. In Figure 11B, you can see some difficulties that the image base imposed. The planting trails are on the transverse, being quite different from the position and angles of the images in Figure 11A. Thus, the labeling was already challenging, and yet it is observed that the model was able, in some cases, to detect the presence and overlap of the labeled objects. Figure 11C shows an overview of how the visual analysis of all detected objects per quadrant tested looks like.

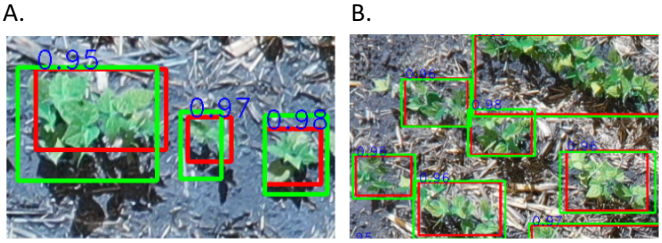



**Figure 11.** Qualitative analysis of some labeled and detected objects using the YOLOv4 model.

## Classification of unknown data using the trained model

To test the generalization ability of the best performing model, YOLOv4, a test set with 190 bounding boxes that are labeled on 3 unduplicated images from the training set was selected. In total 144 objects were correctly identified, amounting to approximately 64% AP and 75% recall with IoU of 0.5. Figure 12 presents the results of these experiments,

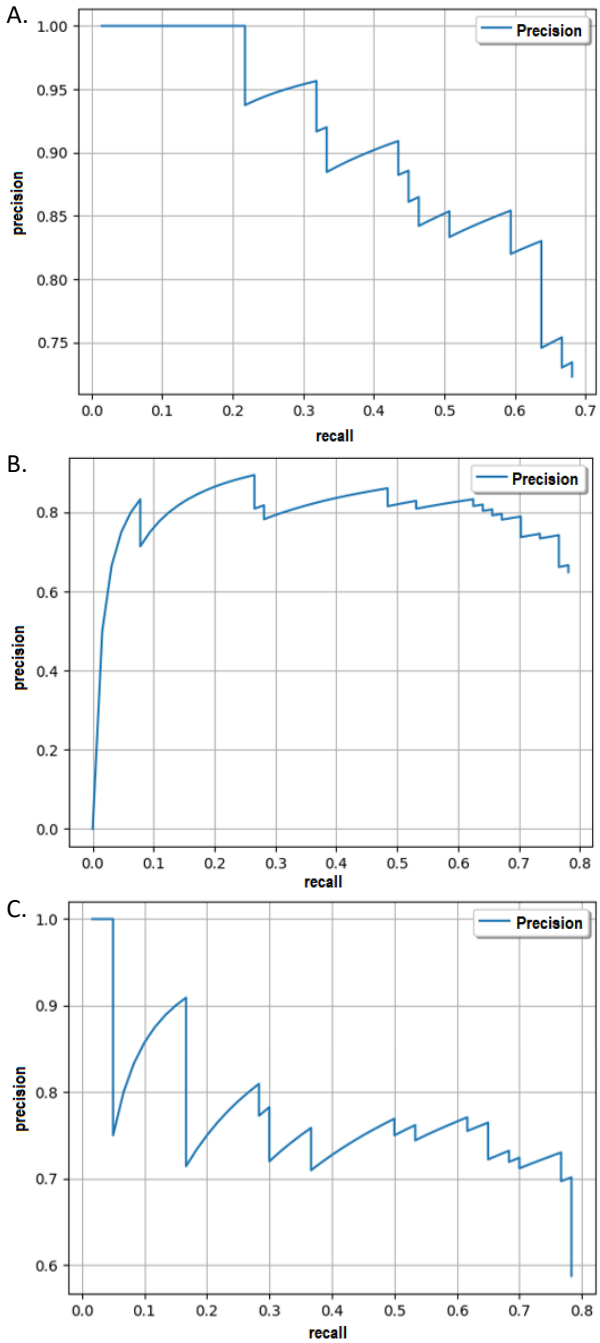



**Figure 12.** Accuracy × Recall of detected images.

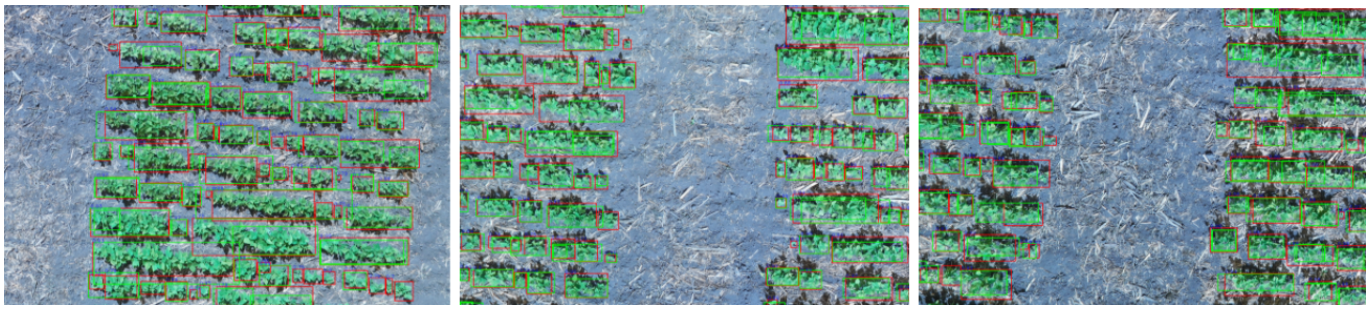


**Figure 13.** Qualitative analysis of the results using YOLOv4.

considering each verified image. It can be seen that Figures 12A and 12C showed greater confusions between the objects. Figure 12B, on the other hand, showed a good relationship between precision and recall.

For qualitative visualization of the results, the 3 images evaluated in Figure 12 were considered. Figure 13 shows the results, where the predicted objects are in green and the labeled one is in red.

Thus, it was possible to observe that the YOLOv4 model performs effectively in detecting plant feet via area images of agricultural environments, showing only a few confusions involving bounding box positioning. These confusions are due to the format in which the images were collected, since they have shadows due to the time of flight (3:00 p.m.) and advanced growth stage in some plants. Thus, the ideal for this same model is to evaluate it with images that are in the initial growth phase, preferably considering the 12:00 p.m. for capturing the images. This can prevent the appearance of shadows that increase confusion in the detection of objects. However, as explained earlier, perfect capture at the early growth stage was not possible due to the high cloudiness during the period of these experiments.

As future work, we intend to evaluate the model with new agricultural image captures, reducing the noise and problems detected in this work. Also improve the robustness of the algorithms in selecting training parameters to eliminate small errors in object detection. It is intended to evaluate the possibility of inserting pre-processing techniques to highlight the objects to be detected as well as including the trained model in a functional system that will be developed.

## Conclusions

With the exploration of the YOLO deep learning architecture two versions were selected with satisfactory experimental results, with 75% recall, although it is notable that more work is needed with the deep learning algorithms to obtain more effective results.

Among the main difficulties encountered in the execution of the experiments was the labeling of the data, since the plants were in an advanced stage of growth, as well as the noise found in the images due to the time and period of image collection. However, it was observed in the qualitative analysis that the detection of the objects was very close to the real thing, indicating that the methods selected are adequate for this proposal.

## Compliance with Ethical Standards

**Author contributions:** Conceptualization: KLW, RO, RF; Data curation: KLW, RF; Formal analysis: KLW, CDP, TAS; Investigation: KLW, CDP, TAS; Methodology: KLW; Software: CDP, TAS; Supervision: KLW, RO; Validation: KLW, RO, RF; Writing-Original Draf: KLW; Writing – review & editing: RO, RF.

**Conflict of interest:** The authors declare that they have no conflict of interest.

**Financing source:** This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

## Literature Cited

Ampatzidis, Y., Partel, V. UAV-based high throughput phenotyping in citrus utilizing multispectral imaging and artificial intelligence. Remote Sensing, v.11, n.4, e410, 2019. https://doi.org/10.3390/rs11040410.

Bochkovskiy, A. Wang, C., Liao, H. M. YOLOv4: optimal speed and accuracy of object detection. arXiv:2004.10934v1 [cs.CV], 2020. https://doi.org/10.48550/arXiv.2004.10934.

Castro, A. I de; Torres-Sánchez, J.; Peña, J. M.; Jiménez-Brenes, F. M.; Csillik, O.; López-Granados, F. An automatic random forest-OBIA algorithm for early weed mapping between and within crop rows using UAV imagery. Remote Sensing, v.10, n.2, e285, 2018. https://doi.org/10.3390/rs10020285.

Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; Li, F-F. ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, Miami. Proceedings... Miami: IEEE, 2009. p.248-255. https://doi.org/10.1109/CVPR.2009.5206848.

Ho, M.; Lin, Y.; Hsu, H.; Sun, T. An Efficient recognition method for watermelon using faster R-CNN with post-processing. In: International Conference on Innovation, Communication and Engineering, 8., 2019, Zhengzhou. Proceedings... Zhengzhou: IEEE, 2019. p.86-89. https://doi.org/10.1109/ICICE49024.2019.9117374.

Kaiming, He; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, v.37, n.9, p.1904–1916, 2015. https://doi.org/10.1109/TPAMI.2015.2389824.

Kalantar, A., Edan, Y., Gur, A.; Klapp, I. A deep learning system for single and overall weight estimation of melons using unmanned aerial vehicle images. Computers and Electronics in Agriculture, v.178, e105748, 2020. https://doi.org/10.1016/j.compag.2020.105748.

Karami, A.; Crawford, M. Automatic plant counting and location based on a few-shot learning technique. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, v. 13, p.5872-5886, 2020. https://doi.org/10.1109/JSTARS.2020.3025790.

Kestur, R.; Angural, A.; Bashir,B.; Omkar, S. N.; Anand, G.; Meenavathi, M. B. Tree crown detection, delineation and counting in UAV remote sensed images: a neural network based spectral–spatial method. Journal of the Indian Society of Remote Sensing, v.46, p.991-1004, 2018. https://doi.org/10.1007/s12524-018-0756-4.

Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. et. al. Path aggregation network for instance segmentation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, Salt Lake City. Proceedings... Salt Lake City: IEEE, 2018. p.8759–8768. https://doi.org/10.1109/CVPR.2018.00913.

Mao, Q.-C.; Sun, H.-M.; Lih, Y.-B.; Jia, R.-S. Mini-YOLOv3: real-time object detector for embedded applications. IEEE Access, v. 7, p. 133529-133538, 2019. https://doi.org/10.1109/ACCESS.2019.2941547.

Milioto, A.; Lottes, P.; Stachniss, C. Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, v. IV-2/W3, p. 41–48, 2017. https://doi.org/10.5194/isprs-annals-IV-2-W3-41-2017.

Neubeck, A.; Van Gool, L. Efficient non-maximum suppression. In: International Conference on Pattern Recognition, 18., 2006, Hong Kong. Proceedings... Hong Kong: IEEE, 2006. p. 850-855. https://doi.org/10.1109/ICPR.2006.479

Neupane, B., Horanont, T.; Hung, N.D. Deep learning based banana plant detection and counting using high-resolution red-green-blue (RGB) images collected from unmanned aerial vehicle (UAV). PLoS ONE, v.14, n.10, e0223906, 2019. https://doi.org/10.1371/journal.pone.0223906.

Nowozin, S. Optimal decisions from probabilistic models: the intersection-over-union case. In: Computer Vision and Pattern Recognition (CVPR), 2014, Columbus. Proceedings... Columbus: IEEE, 2014. p. 548-555. https://doi.org/10.1109/CVPR.2014.77.

Oh, S.; Chang, A.; Ashapure, A.; Jung, J.; Dube, N.; Maeda, M.; Gonzalez, D.; Landivar, J. Plant counting of cotton from UAS imagery using deep learning-based object detection framework. Remote Sensing, v.12, n.18, e2981, 2020. https://doi.org/10.3390/rs12182981.

Pederi, Y.A.; Cheporniuk, H.S. Unmanned Aerial Vehicles and new technological methods of monitoring and crop protection in precision agriculture. In: IEEE International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD), 2015, Kyiv. Proceedings... Kyiv: IEEE, 2015. p. 298-301. https://doi.org/10.1109/APUAVD.2015.7346625.

Powers, D. M. W. Evaluation: from precision, recall and f-measure to ROC, informedness, markedness & correlation. Journal of Machine Learning Technologies, v.2, n.1, p.37-63, 2011. http://www.bioinfo.in/contents.php?id=51. 22 Jun. 2021.

Rahnemoonfar, M.; Sheppard, C. Real-time yield estimation based on deep learning. In: SPIE 10218 Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping, 2., 2017, Anaheim. Proceedings... Anaheim: SPIE, 2017. e1021809. https://doi.org/10.1117/12.2263097.

Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. arXiv:1804.02767v1 [cs.CV], 2018. https://doi.org/10.48550/arXiv.1804.02767.

Sarwar, F. S.; Griffin, A.; Periasamy, P.; Portas, K.; Law, J. et al. Detecting and counting sheep with a convolutional neural network. In: 2018 IEEE International Conference on Advanced Video and Signal Based Surveillance, 15., 2018, Auckland. Proceedings... Auckland: IEEE, 2018. p.1-6. https://doi.org/10.1109/AVSS.2018.8639306.

Valente, J., Sari, B., Kooistra, L., Kramer, H.; Mücher, S. Automated crop plant counting from very high-resolution aerial imagery. Precision Agriculture, v.21, p.1366-1384, 2020. https://doi.org/10.1007/s11119-020-09725-3.

Wang, C.Y.; Liao, H.-Y. M.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W.; Yeh, I-H. CSPNet: A new backbone that can enhance learning capability of CNN. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop (CVPRW), 2020, Seattle. Proceedings... Seattle: IEEE, 2020. p. 1571-1580. https://doi.org/10.1109/CVPRW50498.2020.00203.

Xu, X.; Li, H.; Xi, L.; Qiao, H., Ma, Z.; Shen, S.; Jiang, B.; Ma, X. Wheat ear counting using K-means clustering segmentation and convolutional neural network. Plant Methods, v.16, e102, 2020. https://doi.org/10.1186/s13007-020-00648-8.