

Article

Identification and Counting of Coffee Trees Based on Convolutional Neural Network Applied to RGB Images Obtained by RPA

Lucas Santos Santana ^{*}, Gabriel Araújo e Silva Ferraz ^{*}, Gabriel Henrique Ribeiro dos Santos, Nicole Lopes Bento  and Rafael de Oliveira Faria 

Department of Agricultural Engineering, Federal University of Lavras, Aqueanta Sol, Lavras 37200-900, MG, Brazil

^{*} Correspondence: lucas.unemat@hotmail.com (L.S.S.); gabriel.ferraz@ufla.br (G.A.e.S.F.)

Abstract: Computer vision algorithms for counting plants are an indispensable alternative in managing coffee growing. This research aimed to develop an algorithm for automatic counting of coffee plants and to determine the best age to carry out monitoring of plants using remotely piloted aircraft (RPA) images. This algorithm was based on a convolutional neural network (CNN) system and Open Source Computer Vision Library (OpenCV). The analyses were carried out in coffee-growing areas at the development stages three, six, and twelve months after planting. After obtaining images, the dataset was organized and inserted into a You Only Look Once (YOLOv3) neural network. The training stage was undertaken using 7458 plants aged three, six, and twelve months, reaching stability in the iterations between 3000 and 4000 it. Plant detection within twelve months was not possible due to crown unification. A counting accuracy of 86.5% was achieved with plants at three months of development. The plants' characteristics at this age may have influenced the reduction in accuracy, and the low uniformity of the canopy may have made it challenging for the neural network to define a pattern. In plantations with six months of development, 96.8% accuracy was obtained for counting plants automatically. This analysis enables the development of an algorithm for automated counting of coffee plants using RGB images obtained by remotely piloted aircraft and machine learning applications.

Keywords: remote sensing; deep learning; precision coffee-growing; digital agriculture; plant count



Citation: Santana, L.S.; Ferraz, G.A.e.S.; Santos, G.H.R.d.; Bento, N.L.; Faria, R.d.O. Identification and Counting of Coffee Trees Based on Convolutional Neural Network Applied to RGB Images Obtained by RPA. *Sustainability* **2023**, *15*, 820. <https://doi.org/10.3390/su15010820>

Academic Editor: Georgios Koubouris

Received: 6 December 2022

Revised: 24 December 2022

Accepted: 28 December 2022

Published: 2 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Technological applications in agriculture can contribute to the significant development of agribusiness [1,2]. Application of emerging technologies based on remote sensing to the monitoring of agricultural fields represents an important advance for agriculture [3,4], contributing to improvements in management and increased productivity [5,6]. These technologies involve image processing, artificial intelligence, geographic information systems, sensor networks, and global positioning systems [7]. The interaction of remote sensing technologies and digital agriculture is provided by techniques such as the IoT, cloud processing, big data analytics, machine learning, deep learning, and computer vision [8].

High-spatial-resolution images obtained by RPA enable observations of vegetative vigour and failures in agricultural fields [9]. The analysis of images using computer vision is essential in agricultural research. There are techniques that can be used to identify various characteristics of vegetation in agriculture [10,11].

Computer-vision agricultural monitoring has become an essential technology in crop management. Rico-Fernández et al. [12] characterized applications for classifying and detecting specific objects of interest in photos and videos by algorithm [13]. Algorithm learning allows the automatic discovery of representations necessary for detection and classification from raw data input into systems [14,15]. Heterogeneous landscapes, which are sometimes presented in agriculture, can present difficulties in object detection. Machine

learning models show better results in predicting and identifying anomalies. Convolutional neural networks (CNNs), long short-term memory (LSTM), and deep neural networks (DNNs) are the most commonly applied algorithms [16,17]. The use of machine learning with images obtained by RPA can be seen in research by Osco et al. [18], who employed a CNN for geolocation and counting of citrus plants. Lewis and Espineli [19] used a convolutional neural network to detect nutritional deficiencies in coffee plantations. Kerkech et al. [20] used deep learning with colourimetric spaces and vegetation indices to detect vine diseases.

Advanced algorithms for object detection use convolutional neural networks (CNNs) [21]. CNNs present remarkable performance when locating objects in images with complex backgrounds [22]. A CNN has a convolution layer in which the filtering process is related to different input parts [23]. Furthermore, many computer vision problems are mitigated by convolution neural networks [24].

Research on CNN applications with RPA images is usually performed using multispectral and hyperspectral sensors. These sensors have high acquisition costs, so the insertion of these technologies in agriculture faces resistance. Exploration using RPA with RGB sensors can be a low-cost alternative. The use of RGB images to identify plants in the agricultural field can be made viable by applying digital processing techniques and computer vision [25].

The integration of digital agriculture technology in coffee farming still requires improvements to enable productivity gains and crop profitability [26]. In coffee growing, plant identification through computer vision can contribute to field management [27]. A suitable coffee field formation is obtained by correctly establishing added plants. However, errors can occur in transplanting, leading to various cultivation field failures. Loss of plants in the initial development stage can occur due to factors linked to the mechanized transplanting system, defects in plant root systems, climatic factors, pests, and diseases [28]. Therefore, after culture implantation, it is necessary to replant the plants that have not survived. The number of plants missing from the cultivation stand is determined from visual samples obtained by walking through the field and marking the places where replanting is necessary. This method is a slow, costly, and imprecise method. In this context, the application of remote sensing and computer vision techniques can offer satisfactory results in identifying and counting plants [29].

The automatic detection and counting of plants in coffee farming can quickly and safely provide geo-referenced information on the points that need replanting. This information contributes to the determination of the amount of plant management required in each stand and the number of workers needed to carry out the replanting. Given the questions presented, this research aimed to develop a method for detecting and counting coffee plants based on the You Only Look Once (YOLOv3) CNN and OpenCV tools. The study's contributions are as follows: (i) development of a prototype for a coffee-plant counting algorithm based on pattern recognition; and (ii) identification of the ideal plant age for identification and counting.

2. Materials and Methods

2.1. Image Data Acquisition

Image capture was performed with a Phantom 4 Advance model remotely piloted aircraft (RPA) (Figure 1). This aircraft has a GPS/GLONASS global positioning system for automated missions and a 1" focal aperture RGB spectral sensor with a complementary metal–oxide–semiconductor (CMOS).

Flight plan settings were initiated with area inspection to define the takeoff point ("home"). In addition, climatic conditions were verified: cloud number, insolation levels, wind speed, and presence of birds. After checking these characteristics, the flight mission was defined as a height of 30 m, speed of 3 m/s, and lateral and longitudinal overlap of 80%, which would result in a spatial resolution of 1.68 cm in three spectral bands: red, green, and blue (RGB).



Figure 1. Equipment used to obtain RGB images. (a) Radio control and device for flight mission; (b) remotely piloted aircraft (RPA).

The coffee plantations were characterized by *Coffea arabica* L. The Catuaí Vermelho IAC 99 cultivar was used, planted with spacings of 3.5 m between rows and 0.5 m between plants. The flights were carried at three, six, and twelve months after implantation. This strategy made it possible to understand how the coffee plants' age interfered with the algorithm's accuracy in identifying plant numbers. Evaluation of the stages of growth (Figure 2) formed the test image bank.

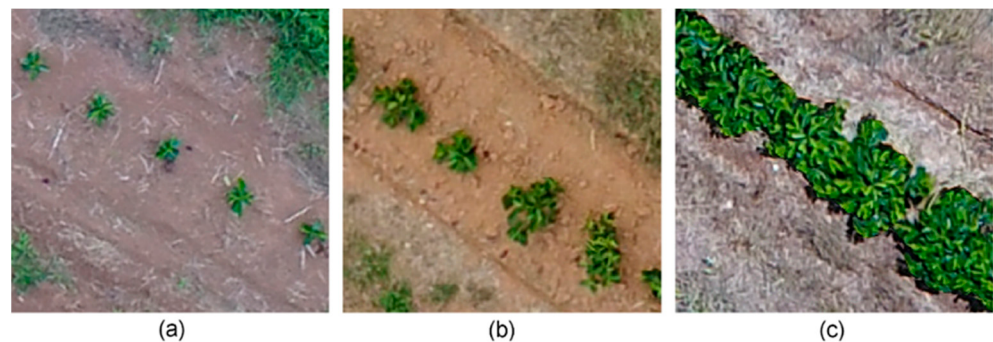


Figure 2. Example of evaluation of plants' age after planting: (a) three months, (b) six months, and (c) twelve months.

Aerial images were processed using Agisoft PhotoScan 1.4 software. The process and parameters used for mosaic formation and RGB band unification were as follows: align photos (high), build a dense cloud (medium), build mesh (medium), and build orthomosaic surface (mesh).

2.2. Image Processing

Large images allow for greater detection accuracy in neural networks, especially for smaller objects spread across the field of view [30]. However, they are rarely used, as they require high computational, time, and financial resources for processing [31]. In orthomosaics of agricultural fields, which represent the entire cultivation area, the scenes have expressive dimensions that create problems for computer-vision techniques. The windowing technique, consisting of cutting orthomosaic pieces to the same dimensions, can be used to address these limitations. Thus, the images were cut in dimensions of 512×512 pixels and submitted to a neural network.

Image training was performed using deep learning techniques. The learning of accurate models using deep learning can be limited by the large amounts of data required, which is a problem that can be mitigated by labelling [32]. Analysis interferences were improved using the insertion of samples through data augmentation. This technique

artificially increases the number of images in a database using geometric transformations (Figure 3). The process mirrors the images (horizontally and vertically) and changes their orientation (45° and 90°). Thus, a neural network will consider a mirrored or rotated image a new image, distinct from the original. As the degree of rotation increases, the data label is no longer preserved in the transformation [33].

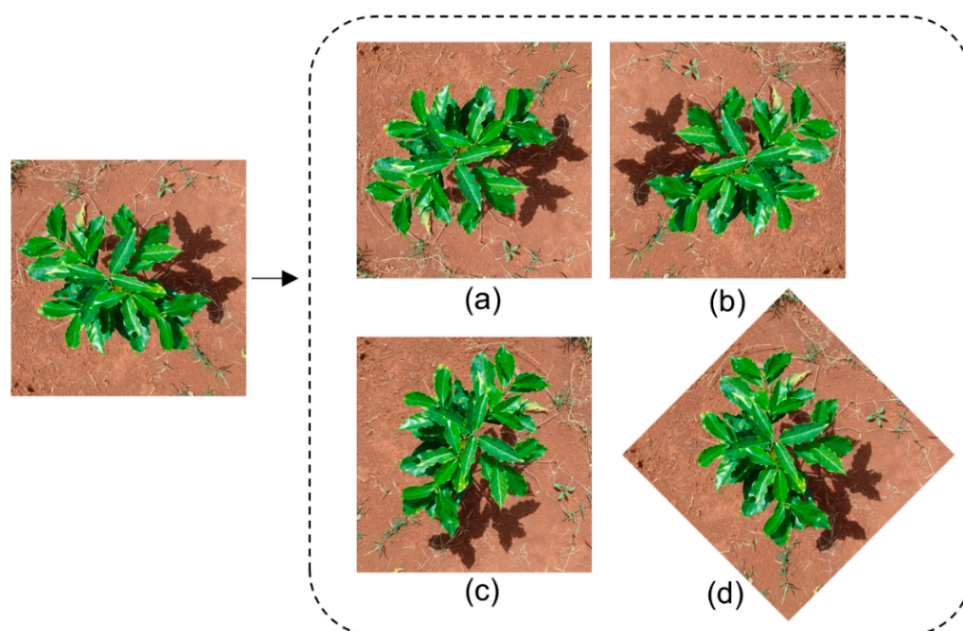


Figure 3. Data augmentation representation. (a) Vertical mirroring, (b) horizontal mirroring, (c) 90° rotation, and (d) 45° rotation.

The application of data augmentation made it possible to increase the number of training images to 1302 clippings totalling 7458 plants (Table 1). The number of plants in the images differed from the real number because the same plants were included in two or more clippings. Therefore, the plants' virtual numbers were higher.

Table 1. Final numbers of cuttings and plants identified in the dataset at coffee development stages.

Development Stage (Age)	Images (Cuts)	Objects (Plants)
Training	1302	7458
Three months ¹	187	931
Three months ²	161	811
Six months ¹	216	770
Six months ²	966	6216

¹ Area 1; ² area 2.

Enlarging images avoids the overfitting problem, which occurs when a statistical model overfits the dataset training process. This problem causes the model to only be accurate when tested with the training set and it will not be able to make correct predictions with unusual datasets [34].

The final preparation was the image labelling applied in the test training. The procedure consisted of inserting a text file containing the terrain truth parameters into each dataset slice. These parameters were represented by a rectangular bounding box parameterized according to the centre point, position, width, and height [35], parameters extracted from each plant present in the images. For the clippings that contained plants at twelve months of implantation, the labels were not produced due to the impossibility of individualizing plants.

2.3. Deep Learning

Algorithm learning was carried through network training based on the fundamental truth. In this step, the neural network knows the desired output result for the respective clipping. Thus, the errors obtained in the output are backpropagated (gradient descent algorithm) to adjust and reduce future errors. The connections between neurons are represented by numerical values responsible for weighting the signal transmitted to subsequent neurons; these values are called synaptic weights [36].

The network learning process involves changing the synaptic weights throughout the training until the best filter values for the dataset are found [37]. The synaptic weights are adjusted based on the error signals, bringing the actual response closer to the desired response [38]. This process aims to calculate the local error gradient (the direction in which the calculated average error value tends to increase) in order to correct the synaptic weights and introduce the opposite slope direction in the local minimum error search [39].

2.4. Detection Algorithm

Object recognition was performed using the third improvement of the You Only Look Once (YOLO) algorithm described by Redmon et al. [40] as a network architecture. The ability to perform class prediction and create bounding boxes simultaneously differentiates YOLOv3 from traditional algorithms. Furthermore, it only uses a neural network to predict bounding boxes and class probabilities [41].

The YOLO architecture transforms the detection problem into a regression problem, increasing detection speed compared to regional-based convolutional neural networks (R-CNNs) [42]. This makes the architecture completely optimisable, unlike the detectors in R-CNN-based architectures, where each stage needs to be trained separately [43]. YOLO is classified as a single-stage object detector, dividing the input image into a grid and then adding safety scores in the bounding boxes [44]. YOLO network models with 1000, 2000, 3000, and 4000 iterations were obtained during the training process, making it possible to compare results between the models.

The process for the YOLOv3-based coffee-plant detector consisted of several steps. The first step was to process the dataset and remove erroneous and blurred images. The remaining photos were labelled and enlarged, and the training and testing sets were allocated. The second step involved inserting the images into the YOLOv3 coffee-plant detector for training and model optimization. The third stage involved building the bounding box and the class score simultaneously, making the forecast images available.

Figure 4 shows the coffee-plant detector based on YOLOv3 processes, including internal structure convolution, the residual set, upsampling, and concatenation.

The YOLOv3 architecture training process involved a grid cell where the object centre was used to make the prediction. Each grid cell had three bounding boxes, which are known as anchor boxes (Figure 5). Anchor boxes have pre-selected sizes based on database objects, making the learning process easier. The network thus does not need to learn the geometric aspects from the start. It just adjusts the anchor boxes to the correct location.

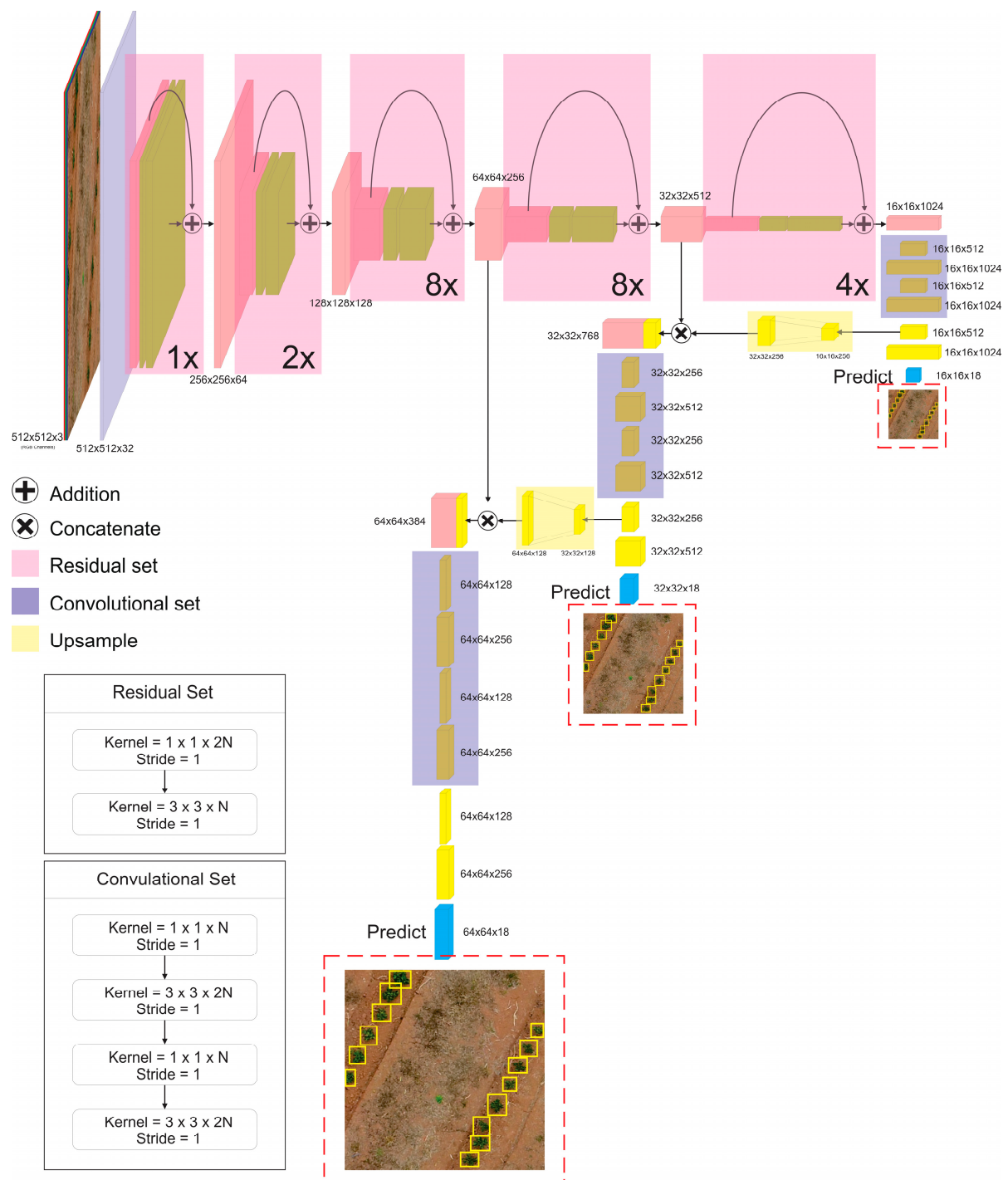


Figure 4. Structure of coffee-plant detector based on YOLOv3.

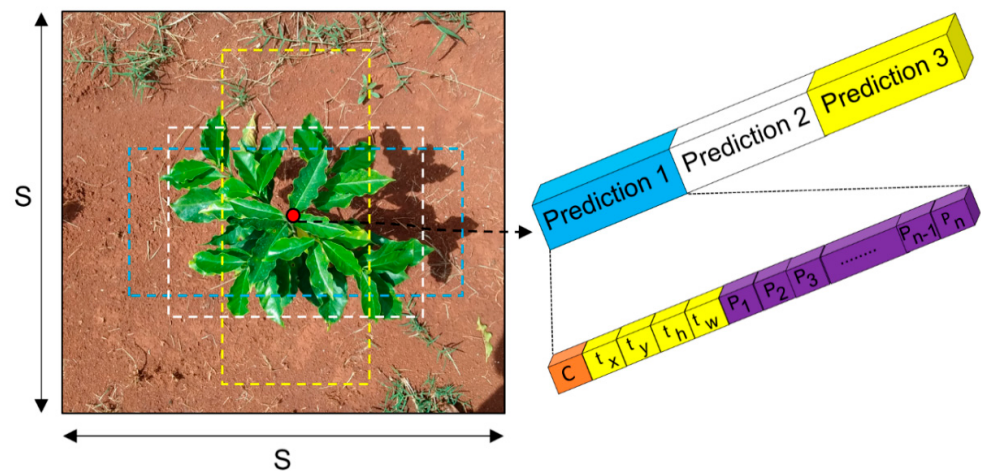


Figure 5. Structure for coffee plant identification using bounding boxes with the YOLOv3 network.

The prediction vector was represented by the respective bounding box confidence value and included an object (C), four values representing the bounding box (t_x , t_y , t_h , and t_w), and each class probability (p_1 , p_2 , p_3 , ..., p_{n-1} and p_n). Equation (1) was used to generate the predictions in the YOLOv3 output:

$$S \cdot S \cdot \{3 \cdot (1 + 4 + C)\} \quad (1)$$

where:

- S: grid dimension;
- C: number of classes in the database.

In practice, the network did not predict the absolute values of the coordinates and dimensions of bounding boxes. This was to ensure better network stability during training. Furthermore, the prediction values ranged from 0 to 1 so that the model would be better focused.

The following equations were used to transform the predicted values into absolute values:

$$\begin{aligned} b_x &= \sigma(t_x) + c_x \\ b_y &= \sigma(t_y) + c_y \\ b_w &= p_w e^{t_w} \\ b_h &= p_h e^{t_h} \end{aligned}$$

where (c_x, c_y) represent the cell displacement in the image and (p_h, p_w) the dimensions of the previously selected anchor boxes.

The loss function used with YOLOv3 to quantify network error predictions during training and minimize them through the gradient descent algorithm can be separated into three parts: loss of location ($Loss_{coord}$), loss of confidence ($Loss_{conf}$), and loss of classification ($Loss_{class}$) [30]:

$$Loos = Loss_{coord} + Loss_{conf} + Loss_{class}$$

Since it is a numerical regression problem, the location loss function calculations (coordinates and box dimensions) used the mean square error (MSE). If the ground truth of a coordinate prediction is $\hat{t}_*$, this subtraction with the predicted coordinate t_* is the error gradient (Equation (2)).

$$\begin{aligned} Loss_{coord} &= \sum_{i=1}^{S \times S} \sum_{j=1}^B 1_{ij}^{obj} (2 - w_i \cdot h_i) \left[\left(\sigma(t_{x_{ij}}) - \sigma(\hat{t}_{x_i}) \right)^2 + \left(\sigma(t_{y_{ij}}) - \sigma(\hat{t}_{y_i}) \right)^2 \right] \\ &\quad + \sum_{i=1}^{S \times S} \sum_{j=1}^B 1_{ij}^{obj} (2 - w_i \cdot h_i) \left[\left(t_{w_{ij}} - \hat{t}_{w_i} \right)^2 + \left(t_{h_{ij}} - \hat{t}_{h_i} \right)^2 \right] \end{aligned} \quad (2)$$

The MSE was multiplied by $(2 - w_i \cdot h_i)$, where w_i and h_i are the width and height of the ground truth at roughly the total image size, to increase the location error weight for smaller objects.

When calculating the confidence loss and the class function (Equations (3) and (4)), the binary cross entropy (BCE) function was used, which is more appropriate for situations in which one wishes to measure the proximity of a predicted probability distribution to reality.

$$Loss_{conf} = - \sum_{i=1}^{S \times S} \sum_{j=1}^B \mathbb{1}_{ij}^{obj} [\hat{C}_i \log(C_{ij}) + (1 - \hat{C}_i) \log(1 - C_{ij})] - \sum_{i=1}^{S \times S} \sum_{j=1}^B \mathbb{1}_{ij}^{noobj} [\hat{C}_i \log(C_{ij}) + (1 - \hat{C}_i) \log(1 - C_{ij})] \quad (3)$$

$$Loss_{class} = - \sum_{i=1}^{S \times S} \sum_{j=1}^B \mathbb{1}_{ij}^{obj} \sum_{c \in classes} [\hat{p}_i(c) \log(p_{ij}(c)) + (1 - \hat{p}_i(c)) \log(1 - p_{ij}(c))] \quad (4)$$

During training, the network was forced to use a single bounding box for each object. This was achieved by selecting from among the three boxes the one that had the highest intersection over union (IoU) value with regard to the object's genuine bounding box (ground truth). When this occurred, $\mathbb{1}_{ij}^{obj} = 1$; otherwise, $\mathbb{1}_{ij}^{obj} = 0$. Even with satisfactory results, the bounding box was ignored when it had an IoU value greater than 0.7. Boxes with IoU values below 0.7 were only penalized in the loss of confidence function for non-objects; therefore, $\mathbb{1}_{ij}^{noobj} = 1$ [30].

2.5. Validation

This step consisted of submitting clippings to the neural network for plant detection. Detection tests were performed for coffee plants at different age stages to verify the trained model's generalization power and determine the best age for application of the model. The plantations with plants at three and six months were tested in two replications. The tests were not performed for 12 month old plants since ground truth was needed to measure the quantity and quality of detections. Ground truth values were not obtained because the plants' canopy was mixing, making it impossible to build a bounding box.

The quantity and quality of the detection identifications were obtained by comparing the predicted and desired outputs present for the ground truth of each test clipping. This made it possible to identify:

True Positives (TPs)—objects that were coffee plants and were detected;
False Positives (FPs)—objects that were not coffee plants and were detected;
False Negatives (FNs)—objects that were coffee plants and were not detected.

The TP, FP, and FN detection classification was performed using the intersection over union (IoU) metric, defined as the ratio between the intersection and union of the predicted box with regard to the ground truth box (desired output) (Figure 6).

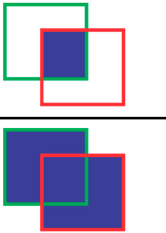
$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


Figure 6. Representation of intersection over union (IoU).

The validation metric was set to an IoU threshold of 0.5. Therefore, for detections with IoU values > 0.5 , the predicted box was considered to be a true positive (TP); otherwise, it was a false positive (FP) [45]. In addition, detections obtained when imaging with non-existent objects of interest were considered false positives. The TPs and FPs made it possible to calculate two essential metrics: accuracy and recall.

Precision refers to the detection accuracy obtained by a neural network and is characterized by the percentage of correct predictions, calculated as follows (Equation (5)):

$$Precision = \frac{TP}{TP + FP} \therefore Precision = \frac{(\text{correct detections})}{(\text{all detections performed})} \quad (5)$$

Recall refers to the ability of a neural network to detect all relevant cases in a test dataset (which is also known as model sensitivity), and it can be obtained through Equation (6):

$$Recall = \frac{TP}{TP + FN} \therefore Recall = \frac{(\text{correct detections})}{(\text{all plants from the test dataset})} \quad (6)$$

Models generally have proportionally inverse behaviour. This is because models favour a high hit rate and reduce the number of detections. In this way, sensitive models can detect objects that do not correspond to the object of interest, resulting in low precision. Satisfactory results are found in models that show equilibrium.

Balance assessment was performed using graphical analysis with a precision–recall (PR) curve to determine whether the model had a reasonable hit rate as sensitivity increased. In addition, models with different training iterations were compared using as a criterion the area below the PR curve, which was characterized as the average precision (AP).

2.6. Plant Count

The coffee-plant counting process was carried out with the results of the plant detection, so the detection quality contributed to the counting accuracy. The plant count was performed with tools provided by the OpenCV library using the Python language. Segmentation techniques were used to highlight the pixels representing the bounding boxes [46]. The neural network was previously configured to generate bounding boxes in cyan, the target colour in the segmentation. In this analysis, the areas with pixels in this colour were the bounding boxes kept in the orthomosaic, while black was used for the other pixels. Aggregation of areas of interest made it possible to binarize the orthomosaic into images with only the colours white and black, with the areas of interest being the white pixels.

After identification and counting training, the algorithm was applied to a commercial coffee crop for six months using the same flight parameters applied in the training and testing stages.

3. Results

3.1. Training

The results obtained during model training are shown in Figure 7. The dataset inserted into the YOLOv3 network achieved satisfactory results after adjustments. It is possible to observe an expressive evolution in plant detection errors during the learning iterations. Despite still decaying, the results after 3000 it were adequate for coffee plant identification.

As shown in Figure 7, the cost function decayed during training. This occurred because the backpropagation algorithm changed network weights based on the error-surface gradient of descent. This minimized the difference between the obtained and desired outputs at each training iteration. This network behaviour followed the surface slope direction created by the objective function (loss), a process of descent until stability was reached [47].

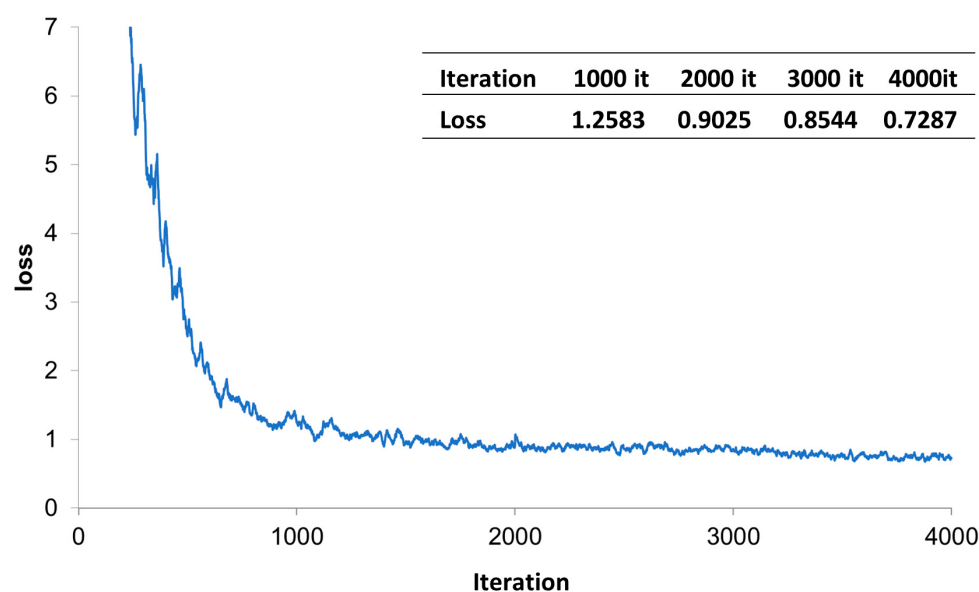


Figure 7. Training results with YOLOv3 network for coffee plant detection.

3.2. Coffee Plant Detection

The different coffee plant ages directly altered the plants' identification. Tests performed at various ages demonstrated the interference of these characteristics in plant detection (Table 2). Table 2 shows the performance with multiple iterations and the precision, average precision, and recall values.

Table 2. Performance with different iteration models at different plant ages.

Plant Age	Model	TP	FP	FN	Precision	Recall	AP
Three months ¹	1000 it.	282	220	529	0.562	0.348	0.36
	2000 it.	357	272	454	0.568	0.44	0.375
	3000 it.	438	318	373	0.579	0.54	0.463
	4000 it.	417	353	394	0.542	0.514	0.392
Three months ²	1000 it.	517	37	414	0.933	0.555	0.777
	2000 it.	707	57	224	0.925	0.759	0.842
	3000 it.	770	96	161	0.889	0.827	0.887
	4000 it.	705	66	226	0.914	0.757	0.872
Six months ¹	1000 it.	507	55	263	0.902	0.658	0.862
	2000 it.	593	94	177	0.863	0.77	0.853
	3000 it.	695	118	75	0.855	0.903	0.873
	4000 it.	705	109	65	0.866	0.916	0.874
Six months ²	1000 it.	4848	282	1368	0.945	0.78	0.943
	2000 it.	5351	399	865	0.931	0.861	0.951
	3000 it.	5664	544	552	0.912	0.911	0.944
	4000 it.	5899	532	317	0.917	0.949	0.955

TP: true positive, FP: false positive, FN: false negative, AP: average precision, recall: model sensitivity. ¹ Area 1, ² area 2.

Model plant detection achieved the best accuracy for plants within six months of development. As observed in Table 2, relevant detection results were found for images of plants six months of age between 3000 and 4000 iterations, as they presented precision, recall, and AP values above 0.88. The results found for plants of three months, even when applying 4000 it, were inferior. It can be observed that the values for TPs, FPs, and the AP were close.

The detection involved submission of an image to a neural network with objects of interest ("coffee plants") delimited by bounding boxes. The segmentation in the detection,

which determined the plant count, was altered and represented by the contour and filling bounding interior (Figure 8).

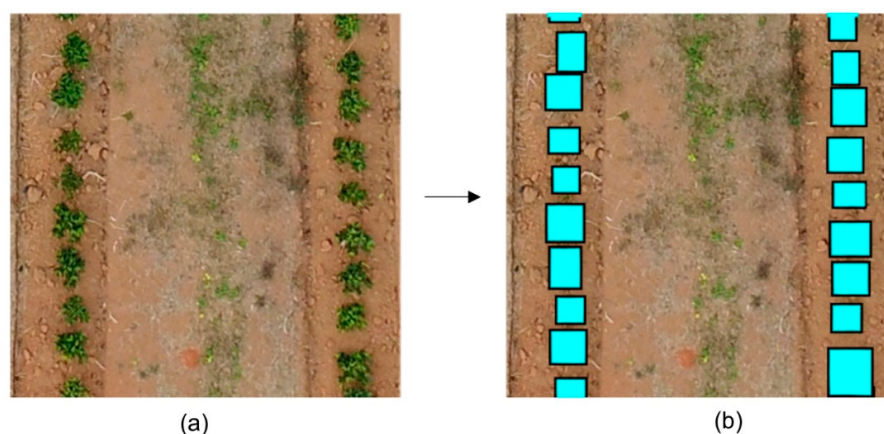


Figure 8. Cutout detections. (a) Input image and (b) identification result.

Formation of the bounding boxes was an important step (Figure 8). This detection process determined the image conditions for the next step, which was the plant count. Furthermore, the final output of the object detection model was a list of bounding boxes that would ideally contain all the plants and their relative locations. The main goal was for the box numbers to match the number of plants in the image.

3.3. Plant Count

The sequence using segmentation techniques from the OpenCV library in Python for counting plants is shown in Figure 9. The input image received the bounding boxes, and then the black backgrounds were applied, the noise was removed, and the area centre was determined.

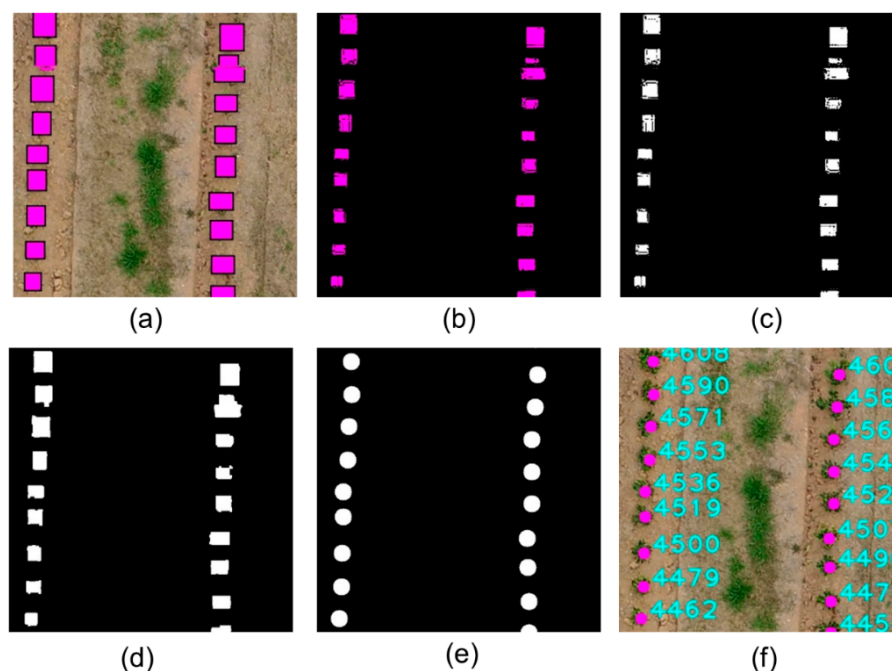


Figure 9. Segmentation process: detection of filled rectangles (a); cyan colour segmentation (b); binarization (c); dilation (d); determination of the centre of each area (e); circle count (f).

The final segmentation process (Figure 9) marked the plants with circles and determined their respective numbers in the orthomosaic. Furthermore, it was possible to identify

some failures in counting plants in this step. They were errors caused by the presence of false positives, in some cases compensating for the occurrence of false negatives.

The plant counting process depended directly on correct object identification. As can be observed, identification of plants within six months of age showed greater accuracy. Table 3 presents the plants' manual counts (three and six months of development) and the trained algorithm capacity with the best iteration tests (3000 and 4000 it).

Table 3. Ability to identify and count coffee plants of different ages.

Ages	Manual Count Count	Algorithm (4000 it.)		Algorithm (3000 it.)	
		Absolute Count	Error (%)	Absolute Count	Error (%)
Three months ¹	860	735	14.5	771	10.3
Three months ²	943	716	24.1	769	18.5
Six months ¹	713	690	3.2	674	5.5
Six months ²	5962	5687	4.6	5523	7.4

¹ Area 1, ² area 2.

The best automatic counting indexes achieved by the YOLOv3 algorithm were identified for plants with six months of development, presenting performance with 96.8% of identifications correct. This high accuracy may have been related to the uniformity of the plants in this period, facilitating object characterization.

3.4. Counting Prototype Performance

The final performance validation was undertaken by applying the algorithm in a commercial cultivation area. This step demonstrated the occurrence of several characteristics that made it possible to identify the errors and algorithm successes practically (Figure 10).

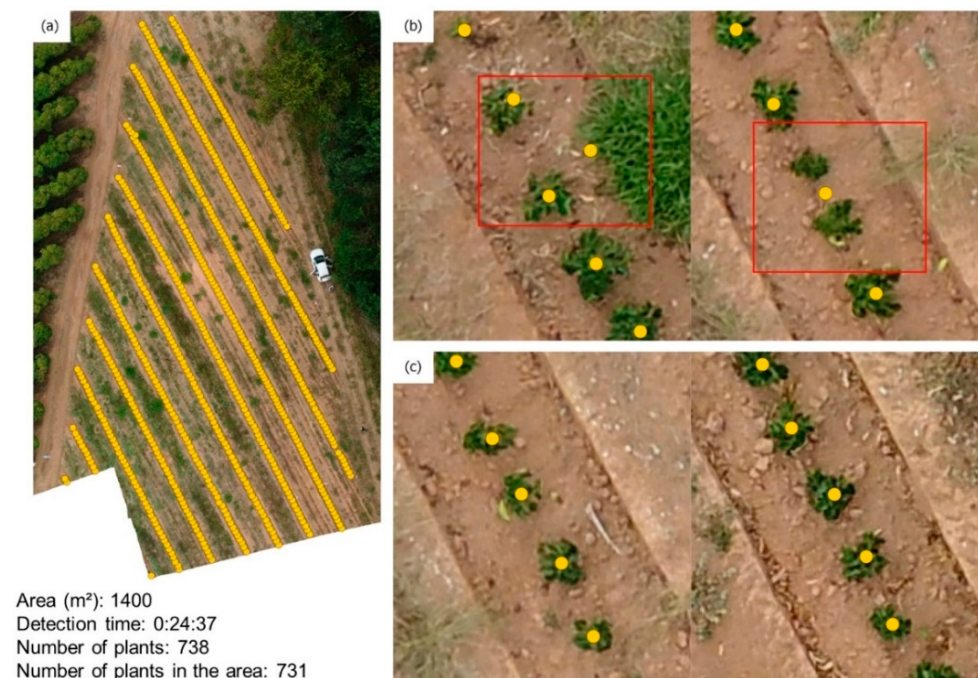


Figure 10. Application of plant-counting algorithm in the commercial planting area. (a) Cultivation area within six months of implantation, (b) errors during identification, and (c) correct identification and counting.

Despite the satisfactory performance in commercial plantations (Figure 10a), it was observed that, at some points, the plant-counting algorithm was influenced by errors in coffee-plant implantation. Two variations in the spacing between plants are presented in Figure 10b. As the algorithm worked with constant identifications, the abrupt variation in

spacing could cause detection errors. Therefore, it was essential to carry out planting with the correct spacing between the plants. In addition, the management of invasive plants would contribute to better detection.

4. Discussion

4.1. Training

Errors relating to reductions (losses) affected the network mainly below 1500 iterations. Above 1500, the results oscillated within a small range. The probable cause for this was stagnation due to the location of the backpropagation algorithm at a minimum surface error. At low numbers of iterations, the most apparent learning procedure disadvantage was that the error surface could contain local minima, so gradient descent could not be guaranteed to find a global minimum [48]. This suggests that training beyond 4000 iterations would not lead to better results than those already obtained. A local minimum solution is not always wrong. It can be very close to what a global minimum solution would be. The purpose of the optimization algorithm was to guide the search to a viable solution point where a prescribed criterion was satisfied, usually that some error measure was below a given tolerance [49].

In the presented training, the model quality, as measured only by the loss value, was sufficient. The training was considered satisfactory, since loss values resulted in exorbitant values in complex training problems. After completing the training with the backpropagation algorithm, the presentation of the test set made it possible to evaluate whether the solution found was acceptable or not [50]. The criteria to be satisfied are defined in the testing stage, among which there are metrics that better characterize network quality.

4.2. Coffee Plant Detection

The plant identification stage presented different characteristics for different coffee plant ages. Depending on the amount of training, significant evolution in the “Recall” model sensitivity occurred; this pattern was observed in all evaluations. This indicated higher detection-model specificity in the first iterations, with the generalization capacity increasing with iterations. The development of the generalization ability contributed to a loss in precision, but the loss was found to be small throughout the training. The maintenance of the AP at good values suggested that, at the end of 4000 iterations, a model with good sensitivity and precision had been obtained. Detection accuracy is the most critical parameter when evaluating a model’s performance [51].

The best results were obtained for the more developed plants in the six-month-old plant tests. Plants aged less than six months could be confused with other invasive plants, as they had smaller canopy sizes. The relationship between plant ages may vary depending on the culture, which indicates that training should be specified according to culture formation [52].

Difficulty in detecting a plant may be due to biological morphology, spectral characteristics, visual textures, and spatial contexts [53]. It may be related to similarities between crops and weeds, dense environments, plant configuration, high-definition canopy mapping, and conflicts between shade and lighting [54]. However, uniform coloration of leaves and some crops’ growth patterns improve the recognition accuracy for these objects [55].

In the case of coffee plants, the optimal recognition point was at six months after planting. The analysis showed an inability to recognize plants at 12 months and low accuracy for plants at 3 months. The plants differed from the soil and invasive plants at this growth stage. In addition, they still had separate crowns, contributing significantly to the good performance.

4.3. Counting Prototype Performance

The tests carried out with commercial cultivation were satisfactory in terms of identification and counting. They demonstrated the high potential of RPA RGB images for automatic plant counting. The counting prototype’s best results were observed six months

after planting, which was mainly influenced by plant uniformity. YOLO-based algorithms behave more assertively when applied to objects with well-defined formats. This feature was also found in the studies by Sozzi et al. [56] demonstrating that YOLO models effectively counted white grape clusters, highlighting a potential application in the robotic platforms used and under development in viticulture.

The results presented in Figure 10 show that some plants were not recognized. Despite the images' high spatial resolution, some errors could still be found. Evaluating weed detection from RGB images, Hasan et al. [51] explained that the use of emerging technologies improves the accuracy and speed of automatic detection systems. As an example, application of spectral indices can improve performance.

Improvements in the quality of plant identification from RGB images can be obtained without applying spectral treatments through rigorous standardization of attributes, such as luminosity, capture height, camera tilt angle, and crop type. Ahmad et al. [57] showed that, before improving the image processing algorithm, the lighting effect should be alleviated and the image quality at the time of acquisition enhanced. According to Gu et al. [58], the proper distance and shooting angle are essential. This can affect the recognition to a certain extent, demonstrating the importance of correct distances from the target. In commercial plantations, identification failure is caused by unequal plant characteristics, such as tipping over at planting time, retarded growth, and attacks by pests.

Even with the characteristics faced in the survey carried out for commercial cultivation, obtaining RGB images is considered a low-cost activity. Therefore, use of these images without complex treatment procedures provides technicians and producers with a new option for coffee tree monitoring.

5. Conclusions

An algorithm based on machine learning was developed for automatic counting of coffee plants from remotely piloted aircraft RGB images. It presented 96.8% accuracy with images without spectral treatment.

The analysis showed the best stage of development to carry out the detection was at six months after transplantation. This was attributed to the amount of leaf mass and the well-defined shape of the plants at this stage. At this age, the plant crowns do not yet mix with other plants, contributing to the algorithm's good performance. Furthermore, there is less confusion between coffee plants and weeds at this age. Plants at 12 months are not recommended for automatic coffee plant detection, as mixing of the coffee plant canopies influences the identification of individual plants from RGB images.

The results presented can contribute to software development for automatic plant counting and automatic location of coordinates in fault regions in coffee plantations.

Author Contributions: Conceptualization, L.S.S. and G.A.e.S.F.; methodology, G.H.R.d.S. and N.L.B.; software, G.H.R.d.S.; validation, G.H.R.d.S., L.S.S. and G.A.e.S.F. formal analysis, N.L.B. and R.d.O.F.; investigation, L.S.S.; resources, R.d.O.F.; data curation, L.S.S. and N.L.B.; writing—original draft preparation, L.S.S. and G.H.R.d.S.; writing—review and editing, L.S.S. and G.A.e.S.F.; visualization, R.d.O.F.; supervision, G.A.e.S.F.; project administration, G.A.e.S.F.; funding acquisition, G.A.e.S.F. and R.d.O.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Council for Scientific and Technological Development (CNPq) (project 305953/2020-6) and Embrapa Café—Coffee Research Consortium (project 10.18.20.041.00.00).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors acknowledge the Embrapa Café—Consórcio Pesquisa Café, the National Council for Scientific and Technological Development (CNPq), the Coordination for the

Improvement of Higher Education Personnel (CAPES), the Federal University of Lavras (UFLA), and Bom Jardim Farm.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Lowenberg-Deboer, J.; Erickson, B. Setting the record straight on precision agriculture adoption. *Agron. J.* **2019**, *111*, 1552–1569. [\[CrossRef\]](#)
- Ren, G.; Lin, T.; Ying, Y.; Chowdhary, G.; Ting, K.C. Agricultural robotics research applicable to poultry production: A review. *Comput. Electron. Agric.* **2020**, *169*, 105216. [\[CrossRef\]](#)
- Marin, D.B.; Ferraz, G.A.e.S.; Santana, L.S.; Barbosa, B.D.S.; Barata, R.A.P.; Osco, L.P.; Ramos, A.P.M.; Guimarães, P.H.S. Detecting coffee leaf rust with UAV-based vegetation indices and decision tree machine learning models. *Comput. Electron. Agric.* **2021**, *190*, 106476. [\[CrossRef\]](#)
- Zhang, J.; Huang, Y.; Pu, R.; Gonzalez-Moreno, P.; Yuan, L.; Wu, K.; Huang, W. Monitoring plant diseases and pests through remote sensing technology: A review. *Comput. Electron. Agric.* **2019**, *165*, 104943. [\[CrossRef\]](#)
- Nicol, L.A.; Nicol, C.J. Adoption of Precision Agriculture in Alberta Irrigation Districts with Implications for Sustainability. *J. Rural Community Dev.* **2021**, *16*, 152–174.
- Zhou, Z.; Majeed, Y.; Diverres Naranjo, G.; Gambacorta, E.M.T. Assessment for crop water stress with infrared thermal imagery in precision agriculture: A review and future prospects for deep learning applications. *Comput. Electron. Agric.* **2021**, *182*, 106019. [\[CrossRef\]](#)
- Bhatnagar, V.; Poonia, R.C.; Sunda, S. State of the art and gap analysis of precision agriculture: A case study of Indian Farmers. *Int. J. Agric. Environ. Inf. Syst.* **2019**, *10*, 72–92. [\[CrossRef\]](#)
- Kayad, A.; Sozzi, M.; Gatto, S.; Whelan, B.; Sartori, L.; Marinello, F. Ten years of corn yield dynamics at field scale under digital agriculture solutions: A case study from North Italy. *Comput. Electron. Agric.* **2021**, *185*, 106126. [\[CrossRef\]](#)
- Jiménez-Brenes, F.M.; López-Granados, F.; Torres-Sánchez, J.; Peña, J.M.; Ramírez, P.; Castillejo-González, I.L.; de Castro, A.I. Automatic UAV-based detection of *Cynodon dactylon* for site-specific vineyard management. *PLoS ONE* **2019**, *14*, e0218132. [\[CrossRef\]](#)
- Gomes, J.F.S.; Leta, F.R. Applications of computer vision techniques in the agriculture and food industry: A review. *Eur. Food Res. Technol.* **2012**, *235*, 989–1000. [\[CrossRef\]](#)
- Vibhute, A.; Bodhe, S.K. Applications of Image Processing in Agriculture: A Survey. *Int. J. Comput. Appl.* **2012**, *52*, 34–40. [\[CrossRef\]](#)
- Rico-Fernández, M.P.; Rios-Cabrera, R.; Castelán, M.; Guerrero-Reyes, H.I.; Juarez-Maldonado, A. A contextualized approach for segmentation of foliage in different crop species. *Comput. Electron. Agric.* **2019**, *156*, 378–386. [\[CrossRef\]](#)
- Karpathy, A.; Leung, T. Karpathy_Large-scale_Video_Classification_2014_CVPR_paper. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 23–28 June 2014; pp. 10–20.
- Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [\[CrossRef\]](#) [\[PubMed\]](#)
- Otter, D.W.; Medina, J.R.; Kalita, J.K. A Survey of the Usages of Deep Learning for Natural Language Processing. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 604–624. [\[CrossRef\]](#)
- Xie, B.; Zhang, H.K.; Xue, J. Deep Convolutional Neural Network for Mapping Smallholder Agriculture Using High Spatial Resolution Satellite Image. *Sensors* **2019**, *19*, 2398. [\[CrossRef\]](#)
- van Klompenburg, T.; Kassahun, A.; Catal, C. Crop yield prediction using machine learning: A systematic literature review. *Comput. Electron. Agric.* **2020**, *177*, 105709. [\[CrossRef\]](#)
- Osco, L.P.; dos de Arruda, M.S.; Marcato Junior, J.; da Silva, N.B.; Ramos, A.P.M.; Moryia, É.A.S.; Imai, N.N.; Pereira, D.R.; Crete, J.E.; Matsubara, E.T.; et al. A convolutional neural network approach for counting and geolocating citrus-trees in UAV multispectral imagery. *ISPRS J. Photogramm. Remote Sens.* **2020**, *160*, 97–106. [\[CrossRef\]](#)
- Lewis, K.P.; Espineli, J.D. Classification and detection of nutritional deficiencies in coffee plants using image processing and convolutional neural network (Cnn). *Int. J. Sci. Technol. Res.* **2020**, *9*, 2076–2081.
- Kerkech, M.; Hafiane, A.; Canals, R. Deep leaning approach with colorimetric spaces and vegetation indices for vine diseases detection in UAV images. *Comput. Electron. Agric.* **2018**, *155*, 237–243. [\[CrossRef\]](#)
- Cui, C.; Fearn, T. Modern practical convolutional neural networks for multivariate regression: Applications to NIR calibration. *Chemom. Intell. Lab. Syst.* **2018**, *182*, 9–20. [\[CrossRef\]](#)
- Chen, Y.T.; Chen, S.F. Localizing plucking points of tea leaves using deep convolutional neural networks. *Comput. Electron. Agric.* **2020**, *171*, 105298. [\[CrossRef\]](#)
- Huang, Y.; Chen, Z.; Yu, T.; Huang, X.; Gu, X. Agricultural remote sensing big data: Management and applications. *J. Integr. Agric.* **2018**, *17*, 1915–1931. [\[CrossRef\]](#)
- Kattenborn, T.; Leitloff, J.; Schiefer, F.; Hinz, S. Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 24–49. [\[CrossRef\]](#)
- Martinelli, F.; Scalenghe, R.; Davino, S.; Panno, S.; Scuderi, G.; Ruisi, P.; Villa, P.; Stroppiana, D.; Boschetti, M.; Goulart, L.R.; et al. Advanced methods of plant disease detection. A review. *Agron. Sustain. Dev.* **2015**, *35*, 1–25. [\[CrossRef\]](#)

26. Bento, N.L.; Ferraz, G.A.E.S.; Barata, R.A.P.; Soares, D.V.; dos Santos, L.M.; Santana, L.S.; Ferraz, P.F.P.; Conti, L.; Palchetti, E. Characterization of Recently Planted Coffee Cultivars from Vegetation Indices Obtained by a Remotely Piloted Aircraft System. *Sustainability* **2022**, *14*, 1446. [\[CrossRef\]](#)
27. Bazame, H.C.; Molin, J.P.; Althoff, D.; Martello, M. Detection, classification, and mapping of coffee fruits during harvest with computer vision. *Comput. Electron. Agric.* **2021**, *183*, 106066. [\[CrossRef\]](#)
28. Santana, L.S.; Ferraz, G.A.E.S.; Cunha, J.P.B.; Santana, M.S.; de Faria, R.O.; Marin, D.B.; Rossi, G.; Conti, L.; Vieri, M.; Sarri, D. Monitoring Errors of Semi-Mechanized Coffee Planting by Remotely Piloted Aircraft. *Agronomy* **2021**, *11*, 1224. [\[CrossRef\]](#)
29. Yiannis, A.; Partel, V. UAV-Based High Throughput Phenotyping in Citrus Utilizing Multispectral Imaging and Artificial Intelligence. *Remote Sens.* **2019**, *11*, 410. [\[CrossRef\]](#)
30. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
31. Uzket, B.; Yeh, C.; Ermon, S. Efficient Object Detection in Large Images Using Deep Reinforcement Learning. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Village, CO, USA, 1–5 March 2020; pp. 1824–1833.
32. Dreossi, T.; Ghosh, S.; Yue, X.; Keutzer, K.; Sangiovanni-Vincentelli, A.; Seshia, S.A. Counterexample-guided data augmentation. In Proceedings of the IJCAI International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 2071–2078. [\[CrossRef\]](#)
33. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [\[CrossRef\]](#)
34. Perez, L.; Wang, J. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *Comput. Vis. Pattern Recognit.* **2017**, *11*, 1–8.
35. Liu, L.; Pan, Z.; Lei, B. Learning a Rotation Invariant Detector with Rotatable Bounding Box. *Comput. Vis. Pattern Recognit.* **2017**. [\[CrossRef\]](#)
36. Werfel, J.; Xie, X.; Seung, H.S. Learning curves for stochastic gradient descent in linear feedforward networks. *Neural Comput.* **2005**, *17*, 2699–2718. [\[CrossRef\]](#) [\[PubMed\]](#)
37. Richards, B.A.; Lillicrap, T.P.; Beaudoin, P.; Bengio, Y.; Sacramento, J.; Saxe, A.; Scellier, B.; Schapiro, A.C.; Senn, W. A deep learning framework for neuroscience. *Nat. Neurosci.* **2019**, *22*, 1761–1770. [\[CrossRef\]](#) [\[PubMed\]](#)
38. Miyashita, D.; Kousai, S.; Suzuki, T.; Deguchi, J. A Neuromorphic Chip Optimized for Deep Learning and CMOS Technology with Time-Domain Analog and Digital Mixed-Signal Processing. *IEEE J. Solid-State Circuits* **2017**, *52*, 2679–2689. [\[CrossRef\]](#)
39. Nandakumar, S.R.; Le Gallo, M.; Piveteau, C.; Joshi, V.; Mariani, G.; Boybat, I.; Karunaratne, G.; Khaddam-Aljameh, R.; Egger, U.; Petropoulos, A.; et al. Mixed-Precision Deep Learning Based on Computational Memory. *Front. Neurosci.* **2020**, *14*, 406. [\[CrossRef\]](#)
40. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; p. 10.
41. Redmon, J. YOLOv3: An Incremental Improvement. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
42. 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. *Comput. Electron. Agric.* **2019**, *157*, 417–426. [\[CrossRef\]](#)
43. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J.; Berkeley, U.C.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; Volume 1, p. 5000. [\[CrossRef\]](#)
44. Huang, R.; Pedoeem, J. YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers. In Proceedings of the IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; p. 7.
45. Sokolova, M.; Japkowicz, N.; Szpakowicz, S. Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation. In *AI 2006: Advances in Artificial Intelligence*; AAAI Work Technical Report; Springer: Berlin/Heidelberg, Germany, 2006; Volume WS-06-06, pp. 24–29. [\[CrossRef\]](#)
46. Xie, G.; Lu, W. Image Edge Detection Based on Opencv. *Int. J. Electron. Electr. Eng.* **2013**, *1*, 104–106. [\[CrossRef\]](#)
47. Ruder, S. An overview of gradient descent optimization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1–14.
48. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [\[CrossRef\]](#)
49. Vogl, T.P.; Mangis, J.K.; Rigler, A.K.; Zink, W.T.; Alkon, D.L. Biological Cybernetics Accelerating the Convergence of the Back-Propagation Method. *Biol. Cybern.* **1988**, *59*, 257–263. [\[CrossRef\]](#)
50. Dias, J.S. *Sensibilidade Paramétrica como Guia para o Treinamento Híbrido de Redes Neurais*; Universidade Federal de Santa Catarina: Florianópolis, Brazil, 1998.
51. Hasan, A.S.M.M.; Sohel, F.; Diepeveen, D.; Laga, H.; Jones, M.G.K. A survey of deep learning techniques for weed detection from images. *Comput. Electron. Agric.* **2021**, *184*, 106067. [\[CrossRef\]](#)
52. Kamilaris, A.; Prenafeta-Boldú, F.X. A review of the use of convolutional neural networks in agriculture. *J. Agric. Sci.* **2018**, *156*, 312–322. [\[CrossRef\]](#)
53. Wang, A.; Zhang, W.; Wei, X. A review on weed detection using ground-based machine vision and image processing techniques. *Comput. Electron. Agric.* **2019**, *158*, 226–240. [\[CrossRef\]](#)

54. Osco, L.P.; Marcato Junior, J.; Marques Ramos, A.P.; de Castro Jorge, L.A.; Fatholahi, S.N.; de Andrade Silva, J.; Matsubara, E.T.; Pistori, H.; Gonçalves, W.N.; Li, J. A review on deep learning in UAV remote sensing. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *102*, 102456. [\[CrossRef\]](#)
55. Xu, W.; Zhao, L.; Li, J.; Shang, S.; Ding, X.; Wang, T. Detection and classification of tea buds based on deep learning. *Comput. Electron. Agric.* **2022**, *192*, 106547. [\[CrossRef\]](#)
56. Sozzi, M.; Cantalamessa, S.; Cogato, A.; Kayad, A.; Marinello, F. Automatic Bunch Detection in White Grape Varieties Using YOLOv3, YOLOv4, and YOLOv5 Deep Learning Algorithms. *Agron. J.* **2022**, *12*, 319. [\[CrossRef\]](#)
57. Ahmad, J.; Muhammad, K.; Ahmad, I.; Ahmad, W.; Smith, M.L.; Smith, L.N.; Jain, D.K.; Wang, H.; Mehmood, I. Visual features based boosted classification of weeds for real-time selective herbicide sprayer systems. *Comput. Ind.* **2018**, *98*, 23–33. [\[CrossRef\]](#)
58. Gu, C.; Wang, D.; Zhang, H.; Zhang, J.; Zhang, D.; Liang, D. Fusion of Deep Convolution and Shallow Features to Recognize the Severity of Wheat Fusarium Head Blight. *Front. Plant Sci.* **2021**, *11*, 599886. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.