COMPUTER VISION SYSTEM FOR DETECTING ORCHARD TREES FROM UAV IMAGES

Hela Jemaa[1*], Wassim Bouachir[2], Brigitte Leblon[2,3], and Nizar Bouguila[1]

[1]Concordia University, Montreal, QC, H3G 1M8 - hela.jemaa@mail.concordia.ca, nizar.bouguila@concordia.ca
[2]Université TELUQ, Montreal, QC, H2S 3L4 - wassim.bouachir@teluq.ca, Brigitte.Leblon@teluq.ca
[3]Faculty of Forestry and Environmental Management, University of New Brunswick,
Fredericton, NB, E3B 5A3 - bleblon@unb.ca

KEY WORDS: Orchards, Unmanned Aerial Vehicles (UAV), Tree Detection, YOLO, DeepForest, CNN.

ABSTRACT:

Orchard tree inventory plays an important role in acquiring up-to-date information on planted trees for effective treatments and crop insurance purposes. Determining tree damage could help assess orchards' health faster and cheaper. Having accurate information on the tree's status could also help managers to plan necessary fieldwork and predict productivity. Traditional orchard inventory is often performed manually, and thus is time-consuming, costly, and subject to error. An alternative is computer vision algorithms that could automatically detect orchard trees based on UAV imagery. The objective of this study is to develop a method using advanced computer vision algorithms to automatically detect apple trees on UAV multispectral images. This task is challenging since apple trees are overlapping over the UAV images, and hence distinguishing different crowns could be difficult. Motivated by the latest advances in UAV imagery and deep-learning models, addressed the tree detection problem by exploring the two CNN models YOLO (You Only Look Once) and DeepForest for detecting apple trees on UAV images. We first constructed a labelled dataset by dividing the study area into equally sized patches. Then we manually annotated all apple trees seen in RGB images. The annotated dataset was then randomly divided into three subsets (training, validation, and testing), for training and testing machine learning models. The performed experiments demonstrate the efficiency and validity of the proposed approach for orchard tree inventory. In particular, the proposed framework achieved a precision of 91% and an F1-score of 87% by adopting the DeepForest model for tree detection.

1. INTRODUCTION

Orchard tree inventories are critical for obtaining current information on planted trees for successful treatments and crop insurance. Therefore, surveying orchard trees, including counting their numbers and determining their locations, pattern, and distribution is important for predicting production volumes and for the purpose of plantation management. The apple orchard tree is considered one of the most popular fruit trees in North America. It is an important fruit crop and a key category of agricultural production. To improve apple orchard production, developing methods to survey and monitor apple tree evolution and production quality is an essential step for farmers. Existing approaches rely on human expertise to extract quantitative orchard tree parameters (e.g. orchard density, crown widths, tree height, leaf area index, and tree position (Belcore et al., 2020). However, traditional orchard inventory is often performed manually, which is labor-intensive and time-consuming. In addition, such traditional methods are costly and subject to errors. Recent advances in remote sensing provided new tools that offer an alternative to traditional methods, such as satellites, airplanes, and unmanned aerial vehicles (UAVs). UAVs are revolutionizing all kinds of industries: These aircraft are becoming more popular due to their cost and time effectiveness when compared to traditional field surveys. Another reason for their popularity is that they can handle a variety of payloads, including optical and hyperspectral cameras, light detection and ranging systems (LiDAR), synthetic aperture radars (SAR), inertial measurement units (IMU), and global positioning systems (GPS) The high spatial resolution UAV images combined with computer vision algorithms are making tremendous advances in domains such as forestry (Grenzdörffer et al., 2008), self-driving cars (Yang and Coughlin, 2014), and bird counting (Zaman et al., 2011).

The aim of this work is to develop a method that takes advantage of advanced computer vision algorithms combined with UAV imagery to automatically detect orchard apple trees in the imageries. This paper is structured as follows. Section 2 introduces background concepts and related works on orchard trees detection with UAV images. Section 3 provides a detailed description of the proposed framework. The experimental results are presented and discussed in section 4. Finally, section 5 concludes the paper and suggests directions for future work.

2. RELATED WORKS

This section presents related works, including traditional and modern tree detection methods, with a focus on UAV images.

2.1 Classical Machine Learning Methods

Classical machine learning-based object detection methods comprise three main stages: image pre-processing, feature extraction, and classification. The methods include local maxima filtering, template matching, valley following, watershed region growing, circular structures fitting, and support vector machines (SVM) with Histogram of Oriented Gradients (HOG). Maillard and Gomes (2016) adapted the "template matching" image processing approach on Very High Resolution (VHR) Google Earth images acquired over a variety of orchard trees. The template is based on a "geometrical optical" model created from a series of parameters, such as illumination angles, maximum and ambient radiance, and tree size specifications. The overall

*Corresponding author: hela.jemaa@mail.concordia.ca

accuracy was above 90% with walnut, mango, and orange trees, but fell under 75% with apple trees. It appears that the openness of the apple tree crown is most probably responsible for this poorer result. Malek et al. (2014) detect palm trees on UAV RGB images by extracting a set of key points using the Scale-Invariant Feature Transform (SIFT). The key points are then analyzed with an extreme learning machine (ELM) classifier which is a priori trained on a set of palm and no-palm tree keypoints.

Canopy Height Models (CHMs) were used by Mohan et al. (2017) to evaluate the applicability of a structure-from-motion (SfM) local-maxima based algorithm for automatic individual tree detection (ITD) on UAV images acquired with low consumer-grade cameras. Based on local maxima and the outcome of a classification process that can distinguish between trees, soil, and shadows, Random Forest regression was used to estimate the number of trees (Fassnacht et al., 2017). These features are fed into a Support Vector Machine (SVM) with a Radial Basis Function to classify the tree species (RBF). Similarly, Wang et al. (2019) used an SVM to classify images into vegetation and non-vegetation features. These features were used to train an SVM to recognize palm trees once HOG was extracted. This approach appeared to be confined to recognizing palm trees and fails when palm trees were mixed in with other tree species. Based on Digital Surface Models, García-Murillo et al. (2020) proposed a system for detecting individual citrus trees using a segmentation method based on Extended Maxima Transforms and a controlled-marker watershed for single tree segmentation. Haddadi et al. (2020) proposed a method that detects apple orchard trees using multispectral UAV images based on Normalized Difference Vegetation Index (NDVI) entropy and variance features. Their method achieved a 93% accuracy.

2.2 Deep-Learning Approaches

Lately, deep learning (DL) methods have flourished as they surpassed classical Machine Learning (ML) algorithms in a large spectrum of computer vision applications. Deep learning methods have demonstrated their ability to extract robust semantic information from massive datasets, allowing them to manage many tough conditions such as scale change, rotation, and appearance variation, thanks to their deep and sophisticated architectures. Convolutional Neural Networks (CNNs) are the most popular deep learning models that have the capability to extract millions of high-level features of objects that can be used effectively for object detection. Several studies explored various Deep Learning algorithms to detect trees over UAV RGB imagery.

Ferreira et al. (2020) detected Amazonian palm trees and their related species using a morphological operations-based approach performed in the score maps of palm species derived from a fully CNN. Safonova et al. (2019) developed a new CNN architecture that predicts damage stages of fir trees in candidate regions chosen by a detection strategy. Jintasuttisak et al. (2022) applied YOLO-V5, the state-of-the-art CNN, for detecting date palm trees on UAV images. Csillik et al. (2018) detected citrus and other crop trees from UAV images using a simple CNN algorithm, followed by a classification refinement using super-pixels derived from a Simple Linear Iterative Clustering (SLIC) algorithm. Li et al. (2017) proposed a deep convolutional neural network (DCNN)-based framework for large-scale oil palm tree detection using high spatial resolution UAV images. To detect and categorize trees in aerial images, Santos et al. (2019) employed a deep learning-based technique. They trained and tested three different detection algorithms: Faster R-CNN (Ren et al., 2015), YOLOv3 (Redmon and Farhadi, 2018), and RetinaNet (Lin et al., 2017b). Similarly, using images collected from UAV along seismic lines, Fromm et al. (2019) trained Faster R-CNN, Single Shot Multi-Box Detector (SSD) (Liu et al., 2016), and R-FCN CNN architectures (Dai et al., 2016) to detect seedlings. This analysis of the various methodologies reveals that CNN-based techniques are becoming more prevalent in tree detection.

3 PROPOSED METHOD

The goal of our research is to develop a model that accurately and automatically detects apple orchard trees on UAV multispectral images. In this paper, we explored the use of two models: YOLO-V5, the latest version of YOLO (Bochkovskiy et al., 2020) and DeepForest (Weinstein et al., 2020). The models were applied to orthomosaics representing an apple orchard that constructed from high-resolution UAV images (Figure 1). The resulting detected tree locations were compared with our ground-truth annotations.

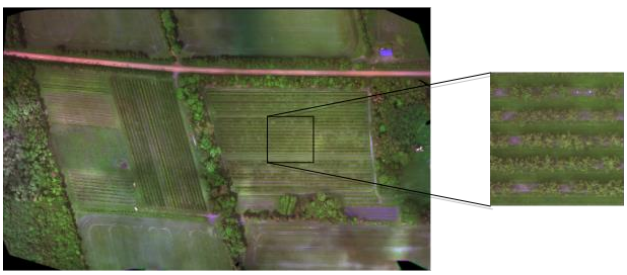

Figure 1. Example of model's input image.

As illustrated in Fig. 2, visual recognition of trees could be a complex task even for a human. This is mainly due to several challenges, such as the similarity in appearance between apple trees and the other trees present in the orchard, appearance variation between apple trees (intra-patch variability), and appearance variability between different patches (inter-patch variability). The intra-patch variability could be explained by the variety of apple species present in the surveyed orchard which has 18 different species (e.g., Cortland, Gala, Sun- rise', 'Cortland', 'Virginia Gold', 'Gala', 'Honey Gold', 'mix', 'Jona Gold', 'Russet', 'Spygold'). The inter-patch variability could be explained by the difference in tree ages. There are patches where only very young trees exist.

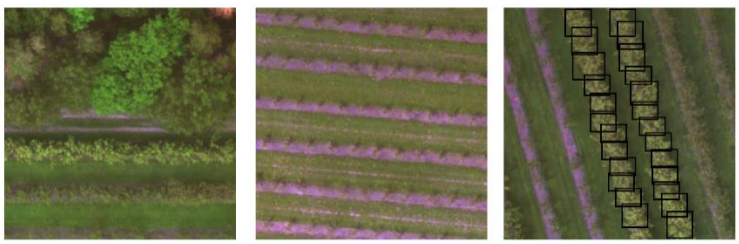

Figure 2. Examples of challenges in orchard orthomosaic: (a) the presence of other trees apart from apple trees, (b) the presence of very young trees, (c) the overlapping tree crowns indicated in the two middle rows.

The workflow of the proposed apple-tree detection system is presented in Fig. 3. It consists of two main phases: the training phase (comprising training and validation) and the testing phase. Prior to those two steps, the captured dataset should be prepared. The details of each phase, that is, data preparation, training, validation, and testing are presented in the following sub-sections.

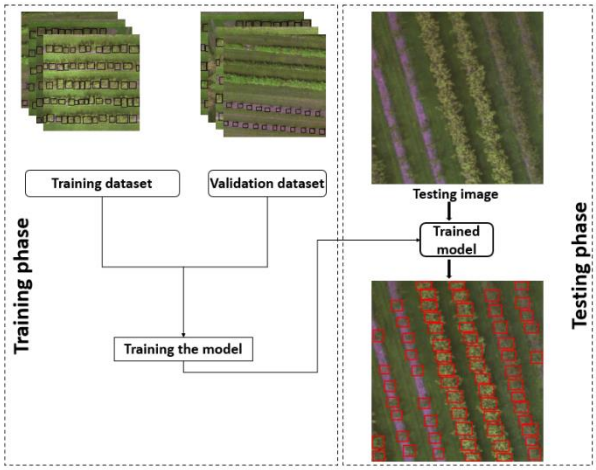

Figure 3. Flowchart of the methodology used in the study.

3.1 Data Aquisition

Images were captured during the summer 2016 over two apple orchards in Souris, PEI, Canada (Lat. 46.44633N, Long. 62.08151W). Our data consists of UAV images taken using a MicaSense RedEdge narrowband camera (MicaSense Inc., Seattle, U.S.A.) mounted on a DJI Matrice 100 quadcopter (Dajiang Innovations Dajiang Baiwang Technology Co., Ltd. Shenzhen, China). It has five sensors, blue, green, red, red edge, and near-infrared.

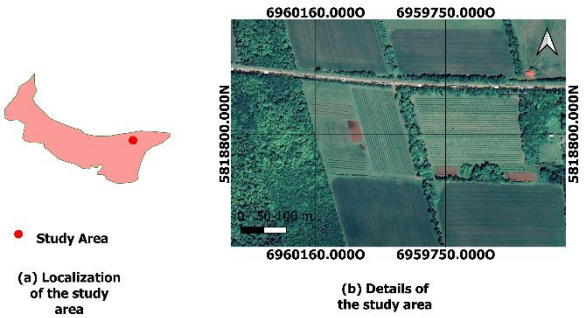

Figure 4. The study area used in this research.

In general, UAV images have high data volume given that they can cover dozens of hectares. Due to the huge volume, complexity, and computational cost, we split the orthomosaic into small patches of 512*512 pixels using a regular grid (Figure 5). The patch size is chosen to fit the models' input. Then, we performed manual annotation which consists in localizing our objects of interest (apple trees) manually using bounding boxes. Through the data annotation step we generate text files encoding: Object id, bounding box localization (center coordinates: x-center, y-center,width, height) of our object of interest (Figure 6)

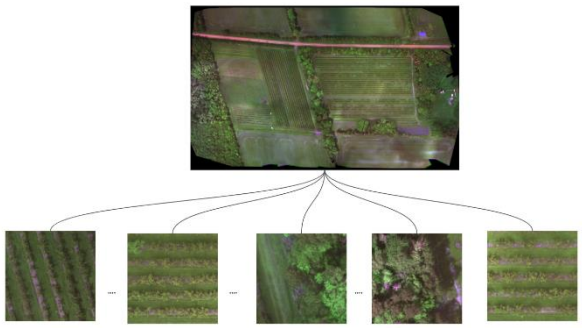

Figure 5. Splitting the UAV orthomosaic into 512*512 patches.

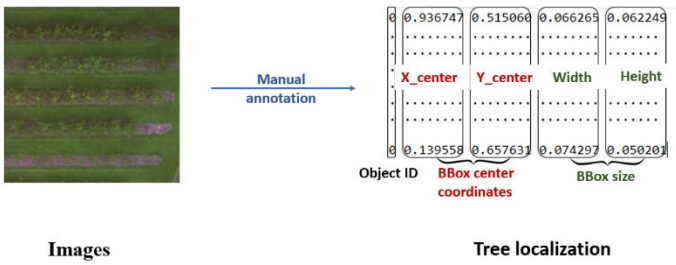

Figure 6. Data annotation step.

Finally, the apple tree images were divided into three datasets: Training, validation, and testing. The labelled data of the training dataset were used to train the models. The validation dataset was used during the training process to assess how well the network was performing during training. The testing dataset was used to quantify the performance of our methods.

3.2 DeepForest

DeepForest is a deep learning model developed to detect individual trees on high-resolution RGB imagery using deep learning (Weinstein et al., 2020). It supports the application of the model to new data, fine-tuning the model to new datasets with user-labeled crowns, training new models, and evaluating model predictions. This simplifies the process of using and retraining deep learning models for a range of forests, sensors, and spatial resolution. DeepForest is mainly based on the Retinanet model, a one-stage object detector that allows the focal loss function to tackle the excessive foreground-background class imbalance between RetinaNet and state-of-the-art two-stage detectors like Faster R-CNN with FPN while running at faster rates. As shown in Fig. 7(a), RetinaNet is a network architecture based on ResNet as a backbone (He et al., 2016). It generates a rich, multiscale convolutional Feature Pyramid Network (FPN) (Lin et al., 2017a) illustrated in Fig.7 (b) that is connected to two subnetworks: one for classifying anchor boxes (Fig.7 (c)) and another one for regressing object boxes (Fig.7 (d)).

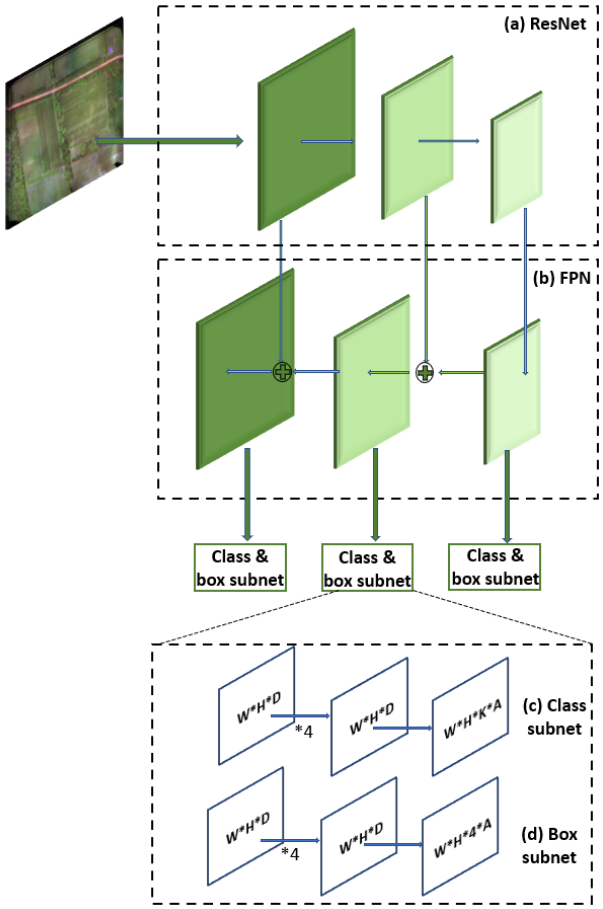

Figure 7. RetinaNet architecture: (a) ResNet backbone, (b) Feature Pyramid Network, (c) Classification subnet, and (d) Box regression subnet.

DeepForest includes one prebuild model that was trained on data from the National Ecological Observatory Network (NEON) using a semi-supervised approach. The model was pretrained on data from 22 NEON sites using an unsupervised LiDAR-based algorithm (Silva et al., 2016) to generate millions of moderate quality annotations for model pretraining. The pretrained model was then retrained based on over 10.000 hand annotations of airborne RGB imagery from six sites. The full workflow of DeepForest prebuilt model is shown in Figure 8.

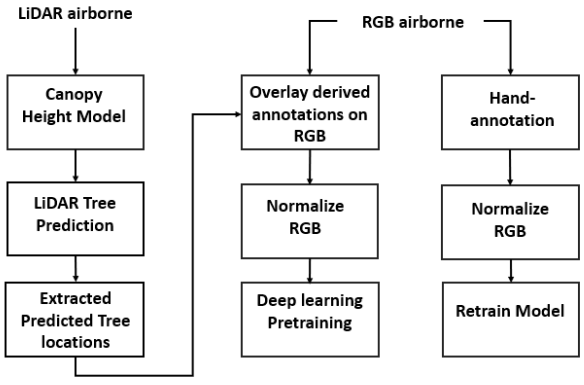

Figure 8. Prebuilt model training workflow. Redrawn from Weinstein et al. (2020).

3.3 YOLO-V5

You Only Look Once (YOLO) is a one-stage object detector (Wang et al., 2020). Its network architecture is made up of three primary parts: the backbone shown in Fig. 9 (a), the neck illustrated in Fig. 9 (b), and the head in Fig.9 (c). The YOLO-V5 model backbone is based on the Cross Stage Partial Network (CSPNet). It aims to extract high-level features while maintaining high accuracy and shortening model processing time. This is accomplished by splitting the base layer's feature map into two sections and then merging them using a suggested cross-stage hierarchy. The fundamental idea is to separate the gradient flow in order to make it propagate over several network paths. Furthermore, CSPNet can significantly minimize the amount of computation required and increase both the speed and accuracy of inference. It deals with three important problems: Strengthening the learning ability of a CNN, removing computational bottlenecks, and reducing memory costs.

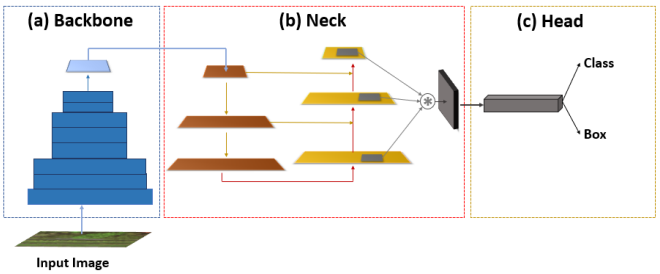

Figure 9. The network architecture of YOLO-V5s.

The model neck is used to collect feature maps from various stages to generate feature pyramids. In this level, the Path Aggregation Network (PANet) (Liu et al., 2018) and Spatial Pooling Pyramid (SPP) (He et al., 2015) are adopted for parameter aggregation from different backbone levels for different detector levels, instead of FPN used in YOLO-v3 (Redmon and Farhadi, 2018). The Spatial pyramid pooling can maintain spatial information by pooling in local spatial bins. These spatial bins have sizes proportional to the image size, so the number of bins is fixed regardless of the image size. This contrasts with the sliding window pooling of the previous deep networks, where the number of sliding windows depends on the input size. The Path Aggregation Network is conducted for improving performance. Its ability to preserve spatial information accurately helps in the proper localization of pixels for mask formation. The property that makes PANet so accurate is that it takes an additional bottom-up path to the top-down path taken by FPN (Lin et al., 2017a). This helps in shortening that path by using clean lateral connections from the lower layers to the top ones. It uses features from all the layers and lets the network decide which ones are useful. It performs a Region of Interest (ROI) Align operation on each feature map to extract the features for the object. This is followed by an element-wise max fusion operation to enable the network to adapt to new features.

PANet uses information from both fully convolutional layers and fully connected layers to provide a more accurate mask prediction. Finally, for the head, the YOLO-v3 anchor-based head architecture is adopted for the used YOLO version. Within each portion of the network described above, YOLO-V5 has numerous key components, including Focus, CBL (Convolution, Batch Normalization, and Leaky- ReLU), CSP (Cross-Stage Partial Connections), and SPP (Spatial Pyramid Pooling). The Focus module divides the input image into four parallel slices, which are then utilized to construct feature maps with the CBL module. The CBL module is a basic feature extraction module that employs a convolution operation, batch normalization, and a

leaky-ReLU activation function. The CSP module is a CSPNet-based module that is used to improve the model's learning capability. The SPP module is a module that allows the mixing and pool of spatial elements (He et al., 2015). It concatenates to its initial features after down sampling the input features through three parallel max-pooling layers.

YOLO-v5 implies some new data augmentation techniques such as Mosaic and SAT (Self-Adversarial Training). The Mosaic technique is illustrated in Figure 10. It mixes 4 training images (contexts) to allow the detection of objects outside their normal context. SAT is basically altering the original image to create the deception that there is no desired object on the image and force the Neural Network to detect an object on this modified image in the normal way. The model depths of each version of the YOLO-V5 network are different, but they are all based on the same network structure, which is made up of three primary parts: the backbone, the neck, and the head. The letters s, m, l, and n in the names of the various YOLO-V5 sub-versions represent the increasing depth of the network architecture used.

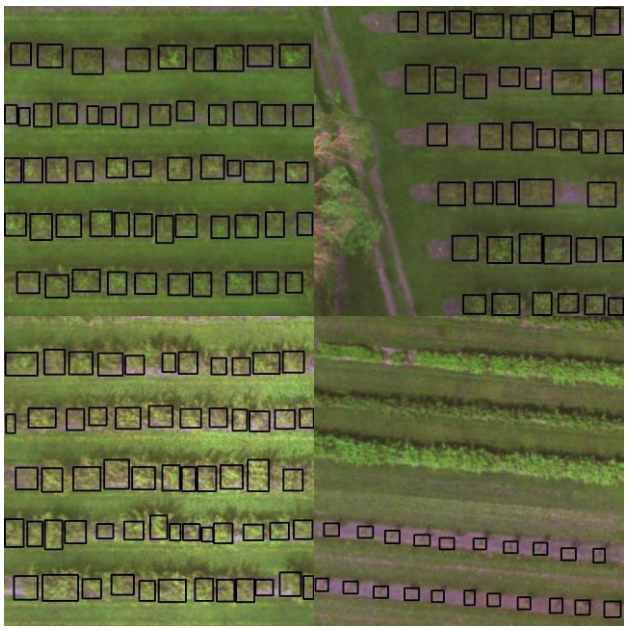

Figure 10. Mosaic data augmentation technique.

4 EXPERIMENTAL RESULTS

In this section, we compare the performance of the YOLO-V5 based approach (of all sub-versions) with the performance of DeepForest method. Each method was trained with the image training dataset and the same set of hyperparameters. (Epoch=500, batch size=8). The performance was compared both quantitatively and qualitatively. All experiments were conducted on a PC with Intel Core i7-7700 CPU, NVIDIA GeForce GTX-1080 GPU, and 64 GB of RAM. The operating system used by the PC was Windows 10.

4.1 Data

The apple tree images were divided into three datasets: training (66%), validation (19%), and testing (15%). Therefore, the number of images in the training, validation, and testing datasets were 73, 21, and 16 respectively. Table 1 provides the number of patches and labeled apple trees in each dataset with the corresponding percentage.

Dataset	Number of patches	Number of labeled trees	Percentage (%)
Training	73	3303	66
Validation	20	620	19
Testing	16	853	15

Table 1. Number of patches and labelled apple trees in each dataset with the corresponding percentages.

4.2 Quantitative performance

The predictions and the ground-truth data were used to compute the following performance metrics:

1. Precision P (Equation 1) is the percentage of correct detections among all the detected trees.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

2. Recall R (Equation 2) is the percentage of correctly detected trees over the total number of trees in the ground truth.

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

3. F1-score (Equation 3) is the harmonic average of precision and recall.

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (3)$$

4. Average Precision, AP (Equation 4), combines precision and recall into a single metric measuring the area under the precision-recall curve resulting in a score ranging from 0 to 1.

$$APk = \sum_{i=1}^{m-1} [recalls(i) - recalls(i+1)] * Precisions(i) \quad (4)$$

5. Mean Average Precision, MAP (Equation 5), is the average of AP.

$$mAP = \sum_{k=1}^{n} APk \quad (5)$$

Where

6. TP = true positives = number of correctly detected trees
7. FP = false positives = number of objects incorrectly detected trees
8. FN = false negatives = number of missed trees
9. n = number of thresholds.

On a test image, a detection is considered as correct if the overlap between the detected tree and the tree in the ground truth was greater than 50%. The overlap between the detection and the ground truth is computed using the Intersection Over Union (IOU) metric. The IOU can be calculated by using the intersection of area between the predicted bounding box and the ground truth and dividing it by the area of the union, as shown in Equation 6.

$$IOU = \frac{Area(B1 \cap B2)}{Area(B1 \cup B2)} \quad (6)$$

Where

1. B1 = area of the ground truth bounding box
2. B2 = area of the predicted bounding box

Table 2 summarizes the quantitative results of the two explored models.

Model	Precision	Recall	F1-score	mAP
DeepForest	**0.908**	**0.830**	**0.868**	**0.812**
YOLO-v5s	0.742	0.743	0.742	0.742
YOLO-v5m	0.668	0.706	0.686	0.69
YOLO-v5l	0.719	0.548	0.626	0.642

Table 2. Metrics associated with each model. Values in bold font correspond to the best achieved performance.

It can be seen that the mAP of YOLO-V5 sub-versions are around 70%, and that the best mAP is 75% when using the YOLO-V5 based model. However, the mAP value of DeepForest model is 82%. In summary, DeepForest outperformed all the tested models with a recall of 83%, a precision of 90.8%, an F1-score of 86.8%, and an MAP of 81.2%. The good result of DeepForest can be explained by the fact that the DeepForest model was pre-trained on over 30 million algorithmically generated crowns from 22 forests and fine-tuned using 10,000 hand-labeled crowns.

4.3 Qualitative performance assessment

Qualitative performance assessment consists of the visual assessment by plotting together the model-detected and the manually delineated trees to analyze how well the model detects trees. Visual assessment of predictions across orchard images reveals a good overall correspondence between predicted and observed bounding boxes, with most errors resulting from the insufficient overlap between observed and predicted tree crowns rather than the model missing a tree entirely. Fig. 11 shows the DeepForest detections for two testing images compared with ground truth. We notice from Fig. 11 (1) that the model incorrectly detects the other present trees as apple ones. This problem could be resolved by applying a mask to filter surrounding trees and to focus only on the apple orchard trees.

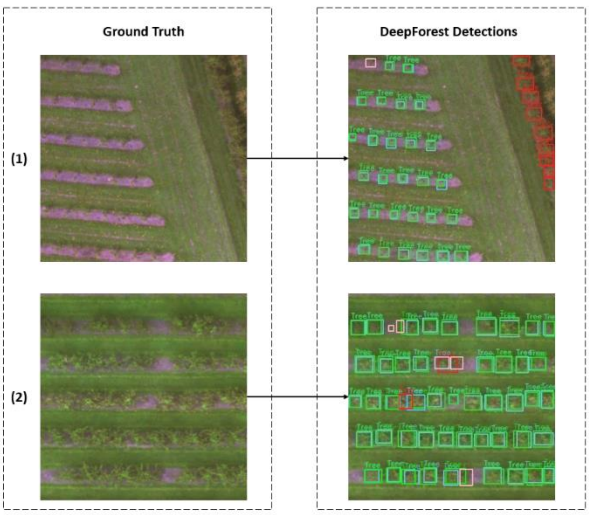

Figure 11. Comparison between the manually detected and DeepForest -based trees on the RGB patches. Red rectangles correspond to false detections (FP), purple rectangles correspond to DeepForest missed detections (FN), blue rectangles correspond to correctly detected trees (TP).

Fig. 12 shows the YOLO-V5 detections for two testing images compared with ground truth.

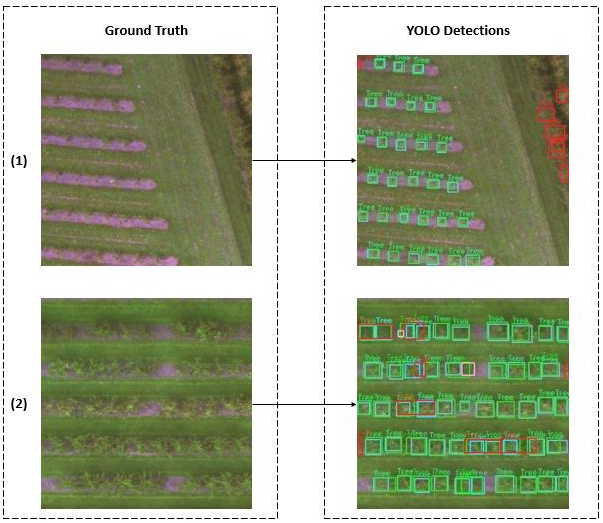

Figure 12. Comparison between the manually detected and YOLO-based trees on the RGB patches. Red rectangles correspond to false detections (FP), purple rectangles correspond to YOLO missed detections (FN), blue rectangles correspond to correctly detected trees (TP).

5 CONCLUSION

In this paper, we explored two state-of-the-art tree detection methods, the latest version of YOLO and the DeepForest model, to detect orchard apple trees from UAV RGB imagery. We started by creating an annotated dataset, then divided the available data into training, validation, and testing datasets. The two models are trained using the same training set. The validation dataset was used to evaluate model performance during training, while the testing dataset was utilised to quantify the performance of the two models.

Through qualitative and quantitative performance assessments, it has been shown that the tested models produce satisfactory results. The DeepForest model had superior performance when compared with the latest version of YOLO.

For future work will investigate the use of multispectral images, including the red-edge and near-infrared images. Also, deep learning algorithms commonly require vast amounts of labelled data. However, the manual labelling of images tends to be costly, challenging, and error prone. Generating synthetic data sets through data augmentation and exploring semi-supervised approaches are interesting techniques to be investigated in future work.

REFERENCES

Belcore, E., Wawrzaszek, A., Wozniak, E., Grasso, N., Piras, M., 2020. Individual Tree Detection from UAV Imagery Using hölder Exponent. Remote. Sens., 12, 2407.

Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y. M., 2020. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

Csillik, O., Cherbini, J., Johnson, R., Lyons, A., Kelly, M., 2018. Identification of citrus trees from unmanned aerial vehicle imagery using convolutional neural networks. Drones, 2(4), 39.

Dai, J., Li, Y., He, K., Sun, J., 2016. R-fcn: Object detection via region-based fully convolutional networks. Advances in neural information processing systems, 29.

Fan, Z., Lu, J., Gong, M., Xie, H., Goodman, E. D., 2018. Automatic tobacco plant detection in UAV images via deep neural networks. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 11(3), 876–887.

Fassnacht, F. E., Mangold, D., Schafer, J., Immitzer, M., Kattenborn, T., Koch, B., Latifi, H., 2017. Estimating stand density, biomass, and tree species from very high-resolution stereo-imagery–towards an all-in-one sensor for forestry applications? Forestry: An International Journal of Forest Research, 90(5), 613–631.

Ferreira, M. P., de Almeida, D. R. A., de Almeida Papa, D., Minervino, J. B. S., Veras, H. F. P., Formighieri, A., Santos, C. A. N., Ferreira, M. A. D., Figueiredo, E. O., Ferreira, E. J. L., 2020. Individual tree detection and species classification of Amazonian palms using UAV images and deep learning. Forest Ecology and Management, 475, 118397.

Fromm, M., Schubert, M., Castilla, G., Linke, J., McDermid, G., 2019. Automated detection of conifer seedlings in drone imagery using convolutional neural networks. Remote Sensing, 11(21), 2585.

García-Murillo, D. G., Caicedo-Acosta, J., Castellanos-Dominguez, G., 2020. Individual detection of citrus and avocado trees using extended maxima transform summation on digital surface models. Remote Sensing, 12(10), 1633.

Grenzdörffer, G., Engel, A., Teichert, B., 2008. The photogrammetric potential of low-cost UAVs in forestry and agriculture. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 31(B3), 1207– 1214.

Haddadi, A., Leblon, B., Patterson, G., 2020. Detecting and counting orchard trees on unmanned aerial vehicle (UAV)-based images using entropy and NDVI features. ISPRS – International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 1211-1215.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 37(9), 1904–1916.

Jintasuttisak, T., Edirisinghe, E., Elbattay, A., 2022. Deep neural network-based date palm tree detection in drone imagery. Computers and Electronics in Agriculture, 192, 106560.

Li, W., Fu, H., Yu, L., 2017. Deep convolutional neural network based large-scale oil palm tree detection for high-resolution remote sensing images. 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), IEEE, 846–849.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017a. Feature pyramid networks for object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 2117–2125.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017b. Focal loss for dense object detection. Proceedings of the IEEE international conference on computer vision, 2980–2988.

Liu, S., Qi, L., Qin, H., Shi, J., Jia, J., 2018. Path aggregation network for instance segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition, 8759–8768.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A. C., 2016. SSD: Single shot multibox detector. Proceedings of the European conference on computer vision, Springer, 21–37.

Maillard, P., Gomes, M. F., 2016. Detection and counting of orchard trees from VHR images using a geometrical-optical model and marked template matching. ISPRS Annals of Photogrammetry, Remote Sensing, Spatial Information Sciences, 3(7), 75-82.

Malek, S., Bazi, Y., Alajlan, N., AlHichri, H., Melgani, F., 2014. Efficient framework for palm tree detection in UAV im- ages. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 7(12), 4692–4703.

Mohan, M., Silva, C. A., Klauberg, C., Jat, P., Catts, G., Cardil, A., Hudak, A. T., Dia, M., 2017. Individual tree detection from unmanned aerial vehicle (UAV) derived canopy height model in an open canopy mixed conifer forest. Forests, 8(9), 340.

Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.

Safonova, A., Tabik, S., Alcaraz-Segura, D., Rubtsov, A., Maglinets, Y., Herrera, F., 2019. Detection of fir trees (Abies sibirica) damaged by the bark beetle in unmanned aerial vehicle images with deep learning. Remote sensing, 11(6), 643.

Santos, A. A. d., Marcato Junior, J., Ara´ujo, M. S., Di Mar- tini, D. R., Tetila, E. C., Siqueira, H. L., Aoki, C., Eltner, A., Matsubara, E. T., Pistori, H. 2019. Assessment of CNN- based methods for individual tree detection on images captured by RGB cameras attached to UAVs. Sensors, 19(16), 3595.

Wang, C.-Y., Liao, H.-Y. M., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., Yeh, I.-H., 2020. Cspnet: A new backbone that can enhance learning capability of CNN. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops,390–391.

Wang, Y., Zhu, X., Wu, B., 2019. Automatic detection of individual oil palm trees from UAV images using HOG features and an SVM classifier. International Journal of Remote Sensing, 40(19), 7356–7370.

Weinstein, B. G., Marconi, S., Aubry-Kientz, M., Vincent, G., Senyondo, H., White, E. P., 2020. DeepForest: A Python package for RGB deep learning tree crown delineation. Methods in Ecology and Evolution, 11(12), 1743–1751.

Yang, J., Coughlin, J. F., 2014. In-vehicle technology for self-driving cars: Advantages and challenges for aging drivers. International Journal of Automotive Technology, 15(2), 333–340.

Zaman, B., Jensen, A. M., McKee, M., 2011. Use of high-resolution multispectral imagery acquired with an autonomous unmanned aerial vehicle to quantify the spread of an invasive wetlands' species. Proceedings of the 2011 IEEE International Geoscience and Remote Sensing Symposium, IEEE, 803–806.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778.