Auto Tree Counting for Oil Palm Plantation Based on Uav/Drone Orthophoto

Mohd Khairul Bazli Bin Mohd Aziz
Centre for Mathematical Sciences, Universiti Malaysia Pahang Al-Sultan Abdullah

Eiffel Syafhamka Nadimen A/L Syamsul
Centre for Mathematical Sciences, Universiti Malaysia Pahang Al-Sultan Abdullah

Wan Mohd Rozaimi Bin Wan Mustafa
MySpatial Sdn. Bhd.

https://doi.org/10.5109/7323301

出版情報：Proceedings of International Exchange and Innovation Conference on Engineering & Sciences (IEICES). 10, pp.463-468, 2024-10-17. International Exchange and Innovation Conference on Engineering & Sciences
バージョン：

Auto Tree Counting for Oil Palm Plantation Based on Uav/Drone Orthophoto

Mohd Khairul Bazli Bin Mohd Aziz[1], Eiffel Syafhamka Nadimen A/L Syamsul[1], Wan Mohd Rozaimi Bin Wan Mustafa[2]
[1]Centre for Mathematical Sciences, Universiti Malaysia Pahang Al-Sultan Abdullah [2]MySpatial Sdn. Bhd.
Corresponding author email: khairulbazli@umpsa.edu.my

Abstract: *Accurate tree counting in plantations is essential for effective management and resource allocation, yet traditional manual digitization methods are labor-intensive and time-consuming. This study presents an innovative approach to automate tree counting in oil palm plantations using UAV/drone orthophoto imagery and the Faster R-CNN algorithm implemented in Python. High-resolution images were captured using UAVs and pre-processed for clarity. The Faster R-CNN model was trained on annotated images with manually labeled bounding boxes. Leveraging a two-stage pipeline, the model generates region proposals and classifies objects within them. Performance evaluation using precision, recall, and Intersection over Union (IoU) metrics demonstrated the model's high accuracy, achieving an average precision (AP) of 75% and an average recall (AR) of 44%. The automated process significantly reduces labor costs and time compared to traditional methods. This study highlights the effectiveness of integrating UAV technology with machine learning for agricultural applications, providing a scalable solution adaptable to various plantations and tree species.*

Keywords: tree counting, oil palm plantation, Faster R-CNN, machine learning

1. INTRODUCTION

Accurate tree counting is a critical task in plantation management, essential for effective resource allocation and operational planning. Traditionally, this task has relied heavily on manual digitization methods, where human labor is used to count and document each tree. While effective in terms of accuracy, these methods are labor-intensive, time-consuming, and prone to human error, making them impractical for large-scale plantations. The manual process can significantly delay decision-making and increase operational costs, posing a substantial challenge for efficient plantation management.

The integration of advanced technologies in agriculture is becoming increasingly important to enhance productivity and reduce manual labor. One such technological advancement is the use of unmanned aerial vehicles (UAVs) or drones, which can capture high-resolution orthophoto images of plantations from above. Coupled with sophisticated machine learning algorithms, such as the Faster Region-based Convolutional Neural Network (Faster R-CNN), these technologies offer a promising solution to automate the tree counting process.

UAVs provide a bird's-eye view of large plantation areas, allowing for comprehensive coverage and detailed imagery that can be used for various analytical purposes. The Faster R-CNN algorithm, known for its accuracy and efficiency in object detection, can process these images to identify and count individual trees. This combination of UAV imagery and machine learning can significantly reduce the time and labor involved in tree counting, providing a scalable and accurate alternative to traditional methods [1], [2].

Despite the potential benefits, there are challenges to implementing these technologies in tree counting. Variations in tree size, density, and overlapping can affect the accuracy of the detection algorithms. Moreover, the quality of the UAV images can be influenced by factors such as lighting conditions and the angle of capture, which must be accounted for in the preprocessing stage [3].

Previous studies have demonstrated the effectiveness of using UAVs and deep learning for precision agriculture and tree counting. For instance, Yamaguchi et al. [6] used high-resolution object detection to improve precision agriculture, while Kato et al. [7] implemented automated tree counting using convolutional neural networks and aerial imagery.

The current reliance on manual digitization for tree counting in oil palm plantations is inefficient and unsustainable for large-scale operations. Manual methods, while accurate, require substantial labor and time, leading to delays in data processing and higher operational costs. This inefficiency hinders timely decision-making and resource management, critical aspects for maintaining plantation productivity and profitability.

Given these limitations, there is a pressing need for a more efficient method that can handle large-scale data with minimal human intervention while maintaining or exceeding the accuracy of manual counting. This study aims to address this need by exploring the use of UAV orthophoto imagery combined with the Faster R-CNN algorithm to automate the tree counting process. By leveraging these technologies, the study seeks to develop a scalable and efficient solution that can improve the accuracy and speed of tree counting in oil palm plantations, ultimately enhancing overall plantation management [4].

The proposed approach not only aims to reduce labor and time costs but also to provide a framework that can be adapted to various types of plantations and tree species. This flexibility is crucial for broader applicability and for addressing the diverse needs of different agricultural sectors.

2. METHODOLOGY

The data for this study consists of high-resolution orthophoto images captured using unmanned aerial vehicles (UAVs) over oil palm plantations in Peninsular Malaysia.

Fig. 1. Palm Oil Plantation Raster.

From Fig. 1, the whole image shows a plantation that is on a hilly plane, thus the trees positioning follows the plane topography. This leads to the tree edges overlapping with one another and not be in the conventional uniform order of the standard plantation.

2.1 Data Preparation

Shown in Fig. 2 is the research design for this study. The research design contains the steps and the process flow of the methodology.

Pre-processing

RGB Conversion: All images were converted from their original format to the RGB color space. This conversion is necessary for the subsequent processing stages as most machine learning algorithms, including Faster R-CNN, operate on RGB images.

Normalization

Each pixel value in the images was normalized by dividing by 255, bringing all pixel values to a range between 0 and 1. This step is essential to ensure that the model can process the images effectively and consistently.

Annotation

Bounding boxes were manually drawn around each oil palm tree in the images. These bounding boxes serve as the ground truth for training the model. The coordinates of these bounding boxes were saved in XML format, including information such as the top-left and bottom-right coordinates, the size of the image in pixels, and the object class (oil palm tree).

Data Augmentation

To improve the model's robustness, data augmentation techniques such as horizontal flipping, rotation, and scaling were applied. These augmentations help the model generalize better by exposing it to various transformations of the input data.

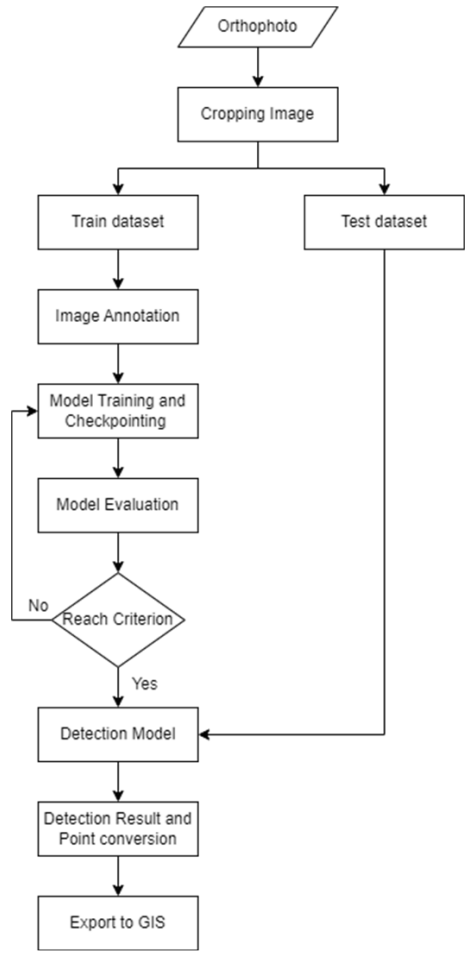


Fig.2. Research design.

2.2 Modeling

The Faster Region-based Convolutional Neural Network (Faster R-CNN) was used for this study due to its high accuracy and efficiency in object detection tasks. The Faster R-CNN model consists of several components, including a feature extraction network, a region proposal network (RPN), and a detection network.

Feature Extraction

The feature extraction network, typically a convolutional neural network (CNN) like ResNet-50, processes the input image to extract high-level features. These features are then used by the RPN to generate region proposals.

Region Proposal Network (RPN)

The RPN generates region proposals by sliding a small network over the feature map output by the feature extraction network. It outputs a set of rectangular object proposals, each with an associated abjectness score. The RPN uses anchor boxes of different scales and aspect ratios to detect objects of various sizes.

Region of Interest (RoI) Pooling
The proposed regions are then passed through an RoI pooling layer, which converts the different-sized proposals into fixed-size feature maps. This step ensures that all proposed regions can be processed uniformly by the subsequent layers.

Detection Network
The detection network classifies the objects within the proposed regions and refines their bounding boxes. It outputs the final object class and the coordinates of the bounding box for each detected object.

2.3 Evaluation

To assess the performance of the model, a thorough evaluation using standard metrics is essential. The primary evaluation tool used is the confusion matrix, which falls under classification evaluation. Table 3.1 presents the confusion matrix:

Table 1. Confusion Matrix

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

From these components, common metrics such as Accuracy, Precision, Recall, and F1 Score can be calculated:

Recall: The ability to identify relevant cases within a dataset, indicating how well the model detects true positives.

$$Recall = \frac{TP}{TP + FN} \quad (1)$$

Precision: The ability to identify only the relevant data points, reflecting the quality of the positive predictions.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

F1 Score: The harmonic mean of Precision and Recall, providing a balance between the two.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

The model's performance in object detection is also measured using the Intersection over Union (IoU) metric. IoU measures the overlap between the predicted bounding box and the ground truth bounding box. It is calculated as:

$$IoU = \frac{TP}{TP + FP + FN} \quad (4)$$

Average Precision (AP) and Average Recall (AR)
To provide a detailed breakdown of the model's performance, different IoU thresholds (0.50, 0.75, and 0.50 to 0.95) and object sizes (small, medium, and large) are used.

Average Precision (AP): Calculated by computing the precision and recall for different IoU thresholds and taking the area under the precision-recall curve. AP measures the model's precision at various thresholds.

$$AP = \sum (p(r) \times \Delta r) \quad (5)$$

Average Recall (AR): Calculated by computing the recall at different IoU thresholds and object sizes, measuring the proportion of true positives detected by the model.

$$AR = \frac{1}{R} \sum_{r=1}^{R} p(r) \quad (6)$$

Higher values of AP or AR indicate better performance of the object detection model.

3. RESULTS AND ANALYSIS

All the images will be converted into Red, Green, and Blue, RGB channel that can be viewed on computer display and all the images will be divided with 255 to normalize the dataset so that the pixel value of images in the dataset will lie between 0 and 1 for a faster training process.

The box coordinates of the annotated image will be extracted from its each individual xml file and will be stored into a list of boxes and its labels. The dataset will also be converted into Tensor so that it can be processed by the deep learning model, in this case the faster R-CNN model. The dataset is split into training and testing data to be used for model evaluation.

3.1 Model Evaluation

From the iteration of training, the IoU metric will show the AP and AR of the overall model, this will indicate whether the model performance is good or not. Each epoch will show a slight improvement of AP and AR evaluation value, this indicates the model underwent training using the custom dataset that was being used. Shown in figure 4.1.5 (a) is the value for metric.

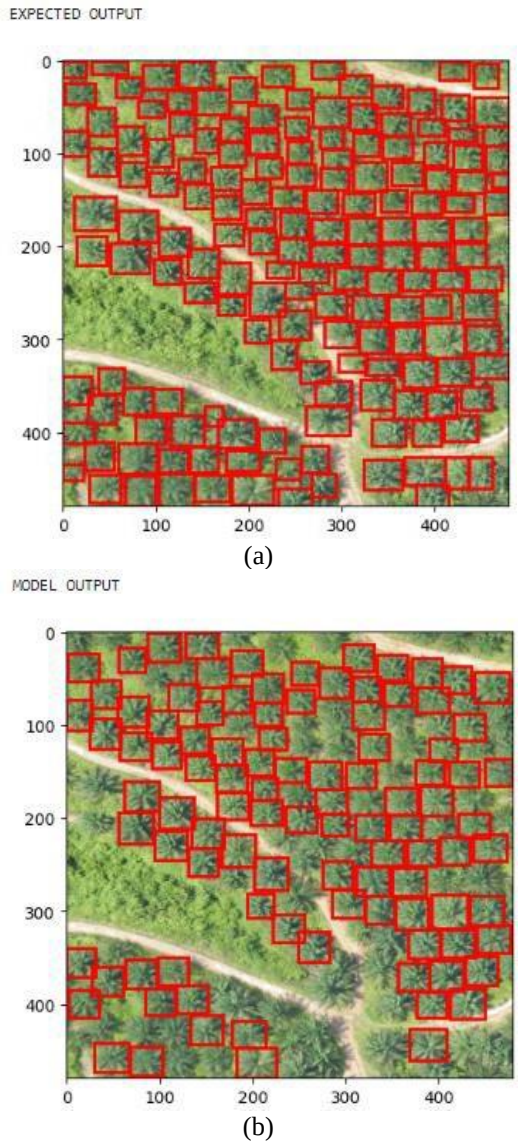
```
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.389
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.750
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.345
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.309
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.460
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.575
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.009
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.078
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.440
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.351
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.524
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.599
```

Fig. 2. Model evaluation result.

The result shows that the AP and AR of the model is acceptable at 75% and 44% respectively, meaning that it can detect the tree correctly but there is still some error when recognizing the correct trees. Shown in Fig. 3 is an example of expected output for the tree detection and the model output for the detected palm oil tree.

From the image in Fig. 3, the model can detect the palm oil tree correctly, but the trees that are so clump together are harder to be detect by the model thus causing the trees to not be box. The model bound each of the boxes in the middle of the tree. Fig. 3(a) shows the expected output, which serves as the ground truth with correctly

placed red boxes marking target detections. Fig. 3(b) presents the model's actual output, with red boxes showing where the model has identified targets. By comparing these two images, it shows the model's accuracy, identifying any discrepancies such as missed detections or false positives.



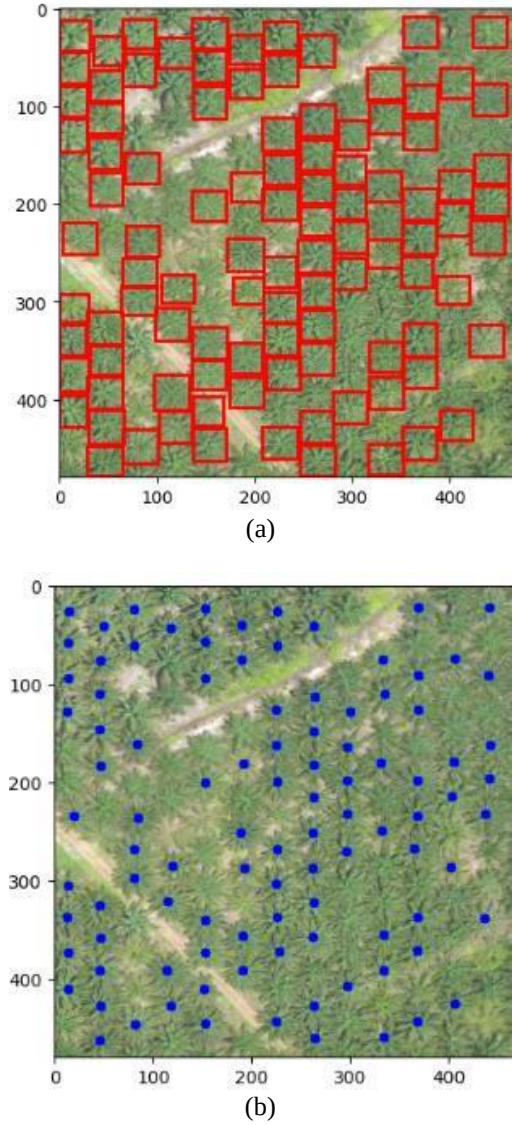
(a)

(b)

Fig. 3. Expected and model output comparison.

3.2 Orthophoto Analysis and Tree Point Generation

With bounding box being able to successfully detect the tree and bound it in the middle of the box, it can be converted to point by finding the midpoint of the predicted boxes. The analysis of the test dataset's orthophoto images demonstrated the model's ability to detect trees effectively. Shown in Fig. 4 is the conversion of the predicted bounding boxes to point plot. Fig. 4(a) shows the predicted output with red bounding boxes indicating the detected objects or regions of interest. In contrast, Fig. 4(b) converts these bounding boxes into a point plot, where each blue dot represents the center of a detected bounding box from Fig. 4(a). The conversion of predicted bounding box in (a) coordinates to the tree crown's midpoint in (b) showed that the model accurately placed the bounding boxes around the trees, with the points plotted on the crown. Although the bounding boxes varied in size based on the tree's size and density, the midpoint calculation consistently placed the point on the tree crown, ensuring accurate tree detection even in areas where trees were densely packed.



(a)

(b)

Fig. 4. Conversion from predicted bounding boxes to point plot.

3.3 Model Accuracy

The model's performance was evaluated using key metrics such as Average Precision (AP) and Average Recall (AR) at different Intersection over Union (IoU) thresholds and object areas. Shown in Fig. 5 the result of the model performance. The model achieved an AP of 38.9% across IoU thresholds of 0.50 to 0.95, indicating moderate overall precision. At an IoU threshold of 0.5, the AP was 75%, suggesting high precision at this threshold. However, the AP dropped to 34.5% at an IoU threshold of 0.75, indicating lower precision at higher thresholds. The AP for small, medium, and large objects was 30.9%, 46%, and 57.5%, respectively, showing better precision for larger objects. The AR can be summarized as in Table 2.

```
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.389
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.750
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.345
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.309
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.460
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.575
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.009
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.078
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.440
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.351
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.524
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.599
```

Fig. 5. Model performance result.

Table 2. AP Summary

Average Precision (AP)	IoU thresholds	Comment
38.9% (all)	0.50 to 0.95	moderate overall precision
75%	0.50	relatively high precision
34.5%	0.75	lower precision
30.9% (small)	0.50 to 0.95	moderate precision
46% (medium)	0.50 to 0.95	higher precision
57.5% (large)	0.50 to 0.95	higher precision

The AR across IoU thresholds and object sizes also varied. For one detection per image, the AR was 0.9%, increasing to 7.8% for 10 detections per image and 44% for 100 detections per image. The AR for small, medium, and large objects was 35.1%, 52.4%, and 59.9%, respectively, indicating better recall for larger objects. The AR can be summarized in Table 3 as such. These metrics suggest that the model can accurately detect oil palm trees, with some limitations in densely packed areas.

Table 3. AR Summary

Average Recall (AR)	IoU thresholds	Comment
0.9% (1 detection)	0.50 to 0.95	low recall
7.8% (10 detection)	0.50 to 0.95	higher recall
44% (100 detection)	0.50 to 0.95	moderate overall recall
35.1% (small with 100 detection)	0.50 to 0.95	moderate precision
52.4% (medium with 100 detection)	0.50 to 0.95	relatively higher recall
59.9% (large with 100 detection)	0.50 to 0.95	relatively higher recall

The Faster R-CNN method was compared to traditional manual digitization across three aspects: accuracy, efficiency, and flexibility.

Accuracy: The Faster R-CNN model achieved an average precision of 75%, while the traditional manual method typically offers higher precision. However, the model's performance can be improved by increasing the training dataset size, reducing human error inherent in manual methods.

Efficiency: The automatic method processed a test dataset of six images (1,298 x 1,430 pixels) in 25 seconds, including bounding box detection, midpoint conversion, georeferencing, and CSV file generation. This efficiency is significantly superior to the manual method, which is time-consuming and labor-intensive.

Flexibility: The Faster R-CNN model demonstrated adaptability to different environments and tree sizes due to the diverse training dataset, making it less biased and more versatile compared to the manual method.

4. CONCLUSION

The deployment of the Faster R-CNN model for palm tree counting has proven to be successful, demonstrating the model's capability to accurately detect and count palm trees in orthophoto imagery. This method offers a significantly faster and more efficient alternative to traditional manual digitization, reducing the time and effort required for tree counting.

However, integrating the model into ArcGIS software presents some challenges, particularly in exporting the model's output into a compatible format. This integration process requires further exploration and development to streamline and enhance compatibility between the model and ArcGIS.

Recent advances in object detection models, particularly for aerial imagery, present intriguing opportunities to improve the accuracy of our Faster R-CNN-based model. For small item recognition in aerial photos, for example, the Efficient-Lightweight YOLO (EL-YOLOv5) model improves detection performance by fine-tuning the architecture to balance low-level and deep feature maps. The approach improves precision without appreciably raising computational complexity by incorporating more precise low-level features [8]. In the same way, the Efficient YOLOv7-Drone model tackles the difficulties of aerial imagery collected by drones, which frequently contains closely spaced small objects. By eliminating ineffective detection heads and using adaptive multi-layer masking to concentrate processing power on pertinent foreground regions, this technique maximizes efficiency and accuracy in detection [9]. A notable advantage of the Faster R-CNN model is its adaptability to other types of farms or tree species. By expanding and modifying the training dataset, the model can be fine-tuned for accurate tree counting in various agricultural settings, making it a versatile tool for different agricultural projects.

In summary, the Faster R-CNN model offers an efficient and accurate solution for palm tree counting. Despite the challenges in integrating with ArcGIS, its speed and flexibility make it a promising tool for tree counting across various agricultural applications. Continued research and development will further enhance its usability and expand its potential in precision agriculture.

5. ACKNOWLEDGEMENT

This research was supported by Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA) under the

UMPSA Research Grant Scheme (RDU230308). We would like to express our gratitude to UMPSA for providing the necessary funding and resources to carry out this project.

6. REFERENCES

[1] Ammar, A., Koubaa, A., & Benjdira, B. (2021). Deep-learning-based automated palm tree counting and geolocation in large farms from aerial geotagged images. Agronomy, 11(8). https://doi.org/10.3390/agronomy11081458

[2] Bayraktar, E., Basarkan, M. E., & Celebi, N. (2020). A low-cost UAV framework towards ornamental plant detection and counting in the wild. ISPRS Journal of Photogrammetry and Remote Sensing, 167, 1–11. https://doi.org/10.1016/j.isprsjprs.2020.06.012

[3] Cheong Lee, C., Yee Tan, S., Sze Lim, T., & Chet Koo, V. (2019). Oil Palm Tree Detection from High Resolution Drone Image Using Convolutional Neural Network. Journal of Engineering Technology and Applied Physics, 1(2), 6–9. https://doi.org/10.33093/jetap

[4] Kalantar, B., Idrees, M. O., Mansor, S., & Halin, A. A. (2017). Smart Counting-Oil Palm tree inventory with UAV Geo-computational Methods in Flood Hazard and Risk Management View project Workshop on Geomatics Development and Point clouds processing of LiDAR and UAV for 3D mapping View project. https://www.researchgate.net/publication/317230939

[5] Li, K., Wan, G., Cheng, G., Meng, L., & Han, J. (2020). Object detection in optical remote sensing images: A survey and a new benchmark. ISPRS Journal of Photogrammetry and Remote Sensing, 159, 296–307. https://doi.org/10.1016/j.isprsjprs.2019.11.023

[6] Yamaguchi, H., Nakashima, Y., & Ishikawa, H. (2019). High-Resolution Object Detection Using Deep Learning and UAV Imagery for Precision Agriculture. IEICES Transactions on Information and Systems, E102-D(10), 2029-2036.

[7] Kato, M., Takahashi, S., & Ito, K. (2020). Automated Tree Counting Using Convolutional Neural Networks and Aerial Imagery. IEICES Transactions on Communications, E103-B(8), 882-891.

[8] Hu, M., Li, Z., Yu, J., Wan, X., Tan, H., & Lin, Z. (2023). Efficient-lightweight yolo: Improving small object detection in yolo for aerial images. Sensors, 23(14), 6423.

[9] Fu, X., Wei, G., Yuan, X., Liang, Y., & Bo, Y. (2023). Efficient YOLOv7-drone: an enhanced object detection approach for drone aerial imagery. Drones, 7(10), 616.