IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 26, Issue 1, Ser. 4 (Jan. – Feb. 2024), PP 37-47
www.iosrjournals.org

Fast And Accurate Detection Of Palm Fruits InComplex Background Orchards

Kalombo Nyembwe Serge
Master In Computer Science And TechnologyIn School Of Information Science And TechnologyAt The Zhejiang Sci-Tech University

Abstract
Identifying palm fruits has been one of the most essential portion of intelligent palm plantation management. Consequently, it is crucial to develop an identification algorithm that is effective, precise, simple to enlarge combined numerical approaches. The current study proposes a strategy based on the most recent deep learning algorithm to identify palm fruit rapidly and precisely in a complicated orchard condition. Here, a monocular camera was used to extract the deep characteristics of palm fruits using the YOLOv5 neural network algorithm, leading to precise identification of various palm dimensions. The detection algorithm possessed approximately a recognition rate of 99.29%, a mean execution time of 0.171s, a shortest execution time of a 0.135s, and an AP of 0.9995, respectively. Furthermore, the identification findings were further discussed with the YOLOv3 and machine learning algorithms. Deep learning algorithms outperformed machine learning algorithms in terms of precise identification and recognition time. Compared to YOLOv3, YOLOv5 yielded a greater detection confidence and precision rate. The outcomes reveal that the proposed approach could rapidly detect various species and maturity levels within palm plantations subjected to various light and shading environments, thereby offering useful insights for palm harvest, maturity and yield prediction.
Keywords*: Palm identification; Orchard habitat; Deep learning, YOLOv5.*

Date of Submission: 14-02-2024 Date of Acceptance: 24-02-2024

I. Introduction

Given the shortage of agricultural labor and the rapid development of artificial intelligence, robots have attracted a lot of attention in the field of agriculture. Agricultural robots automate laborious farm tasks, allowing farmers to focus more on farm management. One of the most popular agricultural robots is the Harvesting robots. Harvesting robots have improved significantly in rapidity and accurateness in recent years, and there has been growing interest in agricultural robots for harvesting fruits and vegetables. The key to automatic picking is the vision system, and accurate identification is the basis for subsequent operations and picking in fruit and vegetable picking. However, developing a robust and efficient fruit detection algorithm is a major challenge due to the similarity or occlusion of fruits and branches, as well as other background issues and uncertainties in the orchard environment.

Palm fruits contribute significantly to global economic growth [1]. Palm fruits are used to make oil and a variety of other products. As a result, increasing their harvesting output is critical. As a result, a system that can detect these in complex orchards is critical. However, detecting fruit from images is a difficult task for a variety of reasons, including appearance variability caused by lighting conditions (natural and artificial light) that prevent the detection of fruits based on color [2]. Fruit from a distance Fruit clustering that prevents direct vision; camera viewpoint, increase data and segmentation as a result of these factors, difficult data labeling, the supervised learning process, and high-resolution imaging impose hardware / algorithm constraints, and accurate detection becomes the main mission of palm collecting cyborg in natural environment. This paper presents a deep learning-based technique for palm tree detection in orchard. We use a standard RGB color camera to capture images of palm trees in a plantation and detect palm tree fruits under different lighting and occlusion conditions.

II. Related work

The remainder of this paper is organized as follows. The second section presents related work. Section III describes the structure and implementation of the palm tree detection algorithm in the plantation. Sections IV and V present the experimental results and comparative analysis. A summary and future work plans are included in Section VI.

Development of convolutional neural networks

In deep learning, researchers have classified convolutional neural network algorithms into three categories based on their research objectives: classification networks, detection networks, and segmentation networks. LeNet [3] was one of the pioneering convolutional neural networks utilized for classification purposes. Later in 2012, Alex et al. [4] deepened the LeNet-based network structure and learned higher dimensional image features, thereby advancing the field.In 2014, Karen and Andrew introduced VGGNet, a convolutional neural network composed of 16-19 layers, establishing that network depth could boost overall performance. The following year, He et al. achieved a 3.57% top5 error rate with Residual Unit, a 152-layer deep neural network that used fewer parameters than VGGNet. Google's Inception-v1 to v4, spanning 2014 to 2016, brought significant advancements. Inception-v1 adopted a sparse connection, unlike VGGNet's full connection. Batch Normalization arrived with Inception-v2. Inception-v3 amplified network depth and nonlinearity, and even altered the network input from 224 to 299. Inception-v4 fused Inception with ResNet to improve the network's capabilities. In 2016, Chollet [5] introduced Xception, which utilized depthwise separable convolution inspired by Inception v3. This improved the model significantly without increasing network complexity. There are two types of detection networks: those based on candidate regions (known as two-stage detectors) and those based on regression methods (one-stage detectors). Among candidate region-based networks, R-CNN series is highly popular. In 2013, Girshick et al. [6] proposed R-CNN, which employed CNN to extract feature vectors from each region proposal, and used linear SVM for classification. Fast R-CNN, introduced by Girshick [7] in April 2015, enhanced object detection accuracy by utilizing a selective search method. In 2015, Ren et al. [8] introduced Faster R-CNN, a more efficient object detection model which employed RPN (Region Proposal Network) instead of selective search to produce proposal windows. A year later, Lin et al. [9] enhanced the Faster R-CNN by proposing FPN, amplifying coarse outputs, and fine-tuning convolution feature map outputs to produce better results. Meanwhile, Redmon et al. [10] presented YOLO (You Only Look Once), another regression- based model that predicted the bounding boxes, confidence, and probability of all object categories in SS grids in a single step. In 2016, Liu et al. [17] introduced SSD (Single Shot MultiBox Detector) which aimed at optimizing the strengths of YOLOv1 and Faster R-CNN. The next year, Redmon and Farhadi [11] combined the YOLOv1 and SSD technologies, resulting in YOLOv2 which leveraged anchors and improved overall performance.The year 2018 saw the introduction of YOLOv3 [12], which surpassed previous versions by utilizing multi-scale feature detection and logistic instead of softmax for classification, resulting in both accuracy and speed. Demonstrating improved performance on small and occluded objects, YOLOv4 was presented by Bochkovskiy et al. in 2020 [13], combining nearly all detection tactics. This model was used in the current study. Segmentation networks are classified into two categories: semantic segmentation and instance segmentation networks. Long et al. proposed FCN in 2014 [14], which classifies images pixel-wise. In 2015, Badrinarayanan et al. suggested Segnet, which utilized deconvolution and upper pooling [15]. In 2014, Chen et al. introduced DeepLab-v1 based on VGG16 [16]In 2016, the upgraded DeepLab-v2 [17] replaced VGG16 with ResNet to accomplish superior segmentation outcomes across multiple scales. A year later, the more universal DeepLab-v3 [18] emerged, featuring a replicated ResNet block that employed the BN layer in ASPP. DeepLab-v3+ [19] arrived in 2018 and harnessed a modified Xception backbone as well as the decode module. In 2019, Auto-DeepLab [20] by Liu et al. was introduced, paving the way foreffective exploration of a two-level hierarchical architecture.

Investigation into identifying fruits and vegetables

One kind of object detection involves identifying fruits and vegetables. Vision-based object detection has a wide range of applications in engineering [21]. Traditional methods of locating regions of interest in an image involve using hand-crafted features such as color, shape, texture, strength, or fusion features and appropriate classifiers (Support vector machine, Adboost, etc.). These methods often lack universality and robustness. However, with the development of deep learning technology, researchers are now focusing on the use of deep convolutional neural networks for fruit and vegetable detection. Deep learning has the ability to extract deep features and has a higher learning capacity, making these algorithms more effective at detecting fruits and vegetables in uncontrolled environments. Researchers have been actively studying fruit detection and branch segmentation, with a particular emphasis on apple fruit detection [22]. Meanwhile, the development of a specialized neural network for mango detection remains ongoing [23]. Similarly, numerous neural networks have been developed for detecting other fruits, such as litchi [24], grape [25], and strawberry [26], all of which have yielded positive results. The detection of fruits such as pomelo, kiwi, waxberry, and guava is also receiving increasing attention. Lately, the development of deep learning has enabled progress in the detection of fruit flowers, which was previously challenging for traditional algorithms [27]. Regarding vegetable detection, the tomato detection network is focused on improving bounding box and detection rate [28]. Furthermore, using deep neural networks, significant advancements have been made in cucumber fruit length estimation [29], sweet pepper detection [30], and date fruit variety and maturity judgment [31].The

application of deep convolutional neural networks in banana plantations has gained prominence in recent years. Neupane et al. [32] utilized RGB aerial images obtained from UAVs with fast-RCNN to count and identify banana plants on a farm. Clark and McKechnie [33], on the other hand, used aerial images to detect banana plantations but focused on using U-NET neural network to create maps rather than conduct research on fruit detection. Meanwhile, Chen et al. [34] managed to get excellent results using the Deeplab V3+ network with two binocular cameras to detect banana central stocks.

The current status of our topic's research progress

Our initial research [35] highlighted the effectiveness of the SVM classifier, coupled with color and texture features, for palm detection. While early studies focused on identifying one particular type of orchard palm fruit for CPU processing purposes, the palm detection model developed can serve as a guide for new varieties that may emerge. The use of GPU processing technology supplies faster, more efficient detection capabilities, and our approach is particularly cost-effective as it employs a standard RGB camera, avoiding the need for complex sensors in image collection. Our research paper introduces the YOLOv5 detection algorithm, the latest and most potent tool designed for palm fruit detection. Designed to emulate human vision, it swiftly detects key features of palm images and accurately locates the fruit within plantations.

III. Materials and methods

Acquiring images

In order to develop and test the suggested algorithm, we collected 1164 palm images from the Congolese Academy of Agronomy on three separate occasions: January 9th and March 19th of 2022, as well as May 27th of 2019. Out of the collected images, 388, 178, and 134 were deemed valid. The Canon sx610hs digital color camera, which boasts a 20481536-pixel resolution, was utilized. Positioned at a height of 150 cm and a shooting distance of approximately 80-120 cm at a horizontal angle, the camera also took ten extra photos from elevation angles of 45 to 60 degrees for comparison purposes. The dataset of 1164 images was split up into three categories: a training set of 835 images, a validation set of 209 images, and a test set of 120 images.The image tagging process was assisted by Python, specifically PyCharm Community Edition 2019.3.1 x64 version, on a powerful Intel(R) Core(TM) i7-9750H @2.6GHz 2.59GHz, 16.0GB RAM laptop and NVIDIA GeForce RTX 2070 with Max -Q on design. Colabeler is a solid open source labeling tool for this task. After successful labeling, a corresponding Extensible Markup Language (XML) file is produced containing the fruit label data and the coordinates of their respective bounding boxes.

Description of algorithm

This paper utilizes the latest detection algorithm, such as YOLOv5. Many scholars prefer the YOLO series due to its flexibility and rapidity in detection. The YOLOv5's rapid accuracy is achieved through a series of algorithmic optimizations, combining a plethora of tricks to increase speed and detection ability. In this paper, we delved into the network's internal structure in great detail and outline its applicability to palm detection in orchards through principle and structural design. Figure 1 presents the flow chart for palm detection with the YOLOv5 algorithm, and the detection procedures are outlined below:

- Step 1: Palm images are transmitted to the network.
- Step 2: Extracting critical data from the image is done through employment of the Mish activation function,utilizing a CSPDarknet53 module as the structural foundation.
- Step 3: To optimize the features extracted from the backbone, a combination of FPN (Feature Pyramid Networks)

+ PAN (Path Aggregation Network) modules and SPP (Spatial Pyramid Pooling) modules form the neck section.

- Step 4: After extracting features, the prediction head crunches the data and produces detection results. Moving forward, let's take a closer look at the module's individual components. The CSPDarknet53 structure forms the backbone, with 5 CSP modules (blue blocks) and 11 CBM modules (yellow blocks) that consist of convolutional, batch normalization, and Mish operations.

Incorporating Batch Normalization and Mish activation functions, the CBM module executes a convolution operation, and is a vital element of the CSP module, which will be elaborated on shortly. Additionally, the CBL module (green block) implements a convolution operation that utilizes Batch Normalization and Leaky Reluactivation functions, similar to the CBM module.

One important aspect to consider is that scholars necessitate an activation function to create nonlinear mappings between inputs and outputs. This enables them to utilize deeper representations and access a more extensive hypothesis space. Although the Leaky Relu function is commonly used in deep learning, the Mish function has been shown to perform better on average.

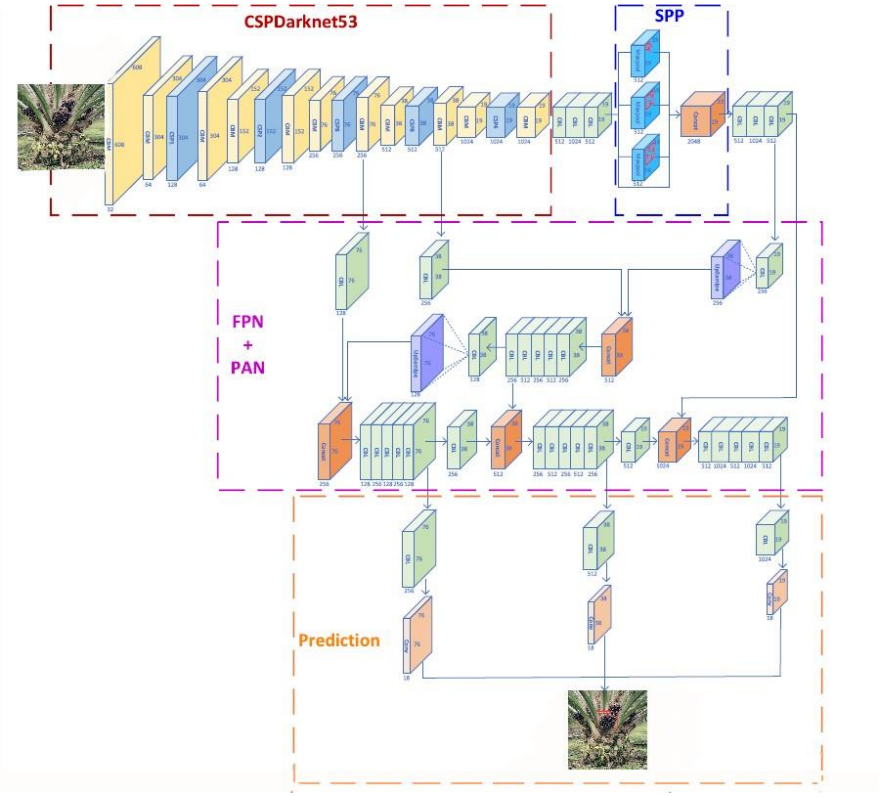


Fig. 1. Flow chart outlining how YOLOv5 detects palms.

Utilizing the Mish activation function is a notable innovation that heightens detection accuracy in the network. While the Leaky Relu activation function is employed in other parts of the network, the Mish activation function is utilized in the backbone.

Comparing the graphs of the Leaky Relu function and the Mish function (as illustrated in Figure 2), it's clear that both have distinct functionalities. One of the network innovations utilized by the CSP module is the incorporation ofCSPn, which represents n Res units.

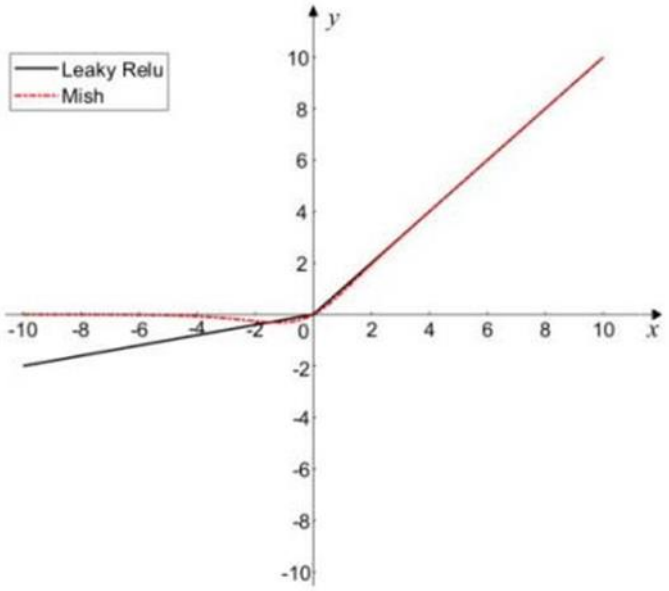


Fig. 2. Mish function and Leaky Relu function.

The structure is shown in Figure 3, where the add operation adds tensors without expanding dimensions, andthe concat operation adds tensors and dimensions.

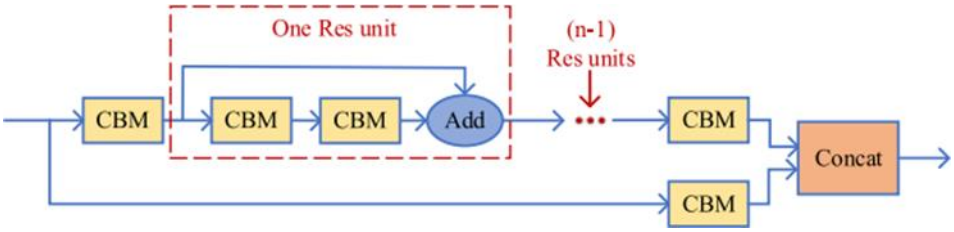


Fig. 3. The structure of CSPn

In the image processing model, CSP1 means one Res unit, while CSP8 represents eight Res units. The CSP module, used in five successive steps, progressively decreases the image's input size from 608 to 19 via down sampling. Upper features are split up and processed individually, then combined to encourage precision and economize memory. Integrating Res Units enhances the network's depth and amplification of feature extraction. The plantation boasts palms bearing green, black, and red fruit, with the black and green hue blending seamlessly with the tree's stem, branches, and leaves. The fruits themselves have an unconventional shape. The detection of palm fruits is highly impacted by the background hence the significance of extracting their deep characteristics. Deep learning modules like SPP and FPN+PAN are used in the network's neck structure (delineated in a purple dotted line area). To keep the size after pooling, the SPP module uses max-pooling of 1 1, 5 5, 9 9, 13 13, padding of 2, and stride of

1. Unlike the traditional approach of max-pooling, the SPP module broadens the acceptance range of backbone network features, leading to improved precision with less calculation cost. YOLOv5 improves this method by encompassing the PAN module, another structural innovation based on the FPN module in YOLOv3. The FPN up- samples the feature map to fuse tensor and dimension with the feature map after CSP operation in the backbone network and convey object semantic information, as reported by the arrow direction in the figure.To enhance small object detection, the FPN+PAN method repeatedly fuses different trunk and detection layers while using several scales to extract more detailed semantic and positioning information. The PAN structure is incorporated with the corresponding FPN scale's feature map, which is subsequently downsampled and convolved to extract positioning features. Palm detection is particularly challenging because of the differing fruit sizes between various palm species, making the detection algorithm's generalization ability important. The network's head structure is the prediction section (exhibited by the orange dotted line area), which outputs three-layer scale feature maps (76 76, 38 38, 19 19) using the CBL module and convolution operation.Every scale predicts three anchor boxes, with each anchor containing six values: four box coordinates, one object confidence, and one class confidence. Consequently, each layer has a total of 18 outputs. By assessing the output information, the network could obtain the bounding box and confidence of the detected palm. Any bounding box whose confidence score is below a set threshold gets eliminated from the list. The remaining boxes undergo an evaluation by the DIOU nms algorithm, which is a novel structural innovation in the network. The Distance-IOU (DIOU) formula is utilized for this purpose.

In measuring the distance between two boxes, "where" signifies the gap separating the midpoint of each, while the diagonal distance between their closed areas represents their "n". Unlike conventional NMS, DIOU nms do not remove boxes with distant center points, acknowledging that they could belong to different objects. Once DIOU nms filtering is complete, the detection results are output, signifying the completion of the task at hand.

IV. Results

The palm detection findings during the training and detection phases are detailed in this section. In addition, the assessment metrics, training variables and detection outcomes across different environments are discussed in detail.

Training model evaluation

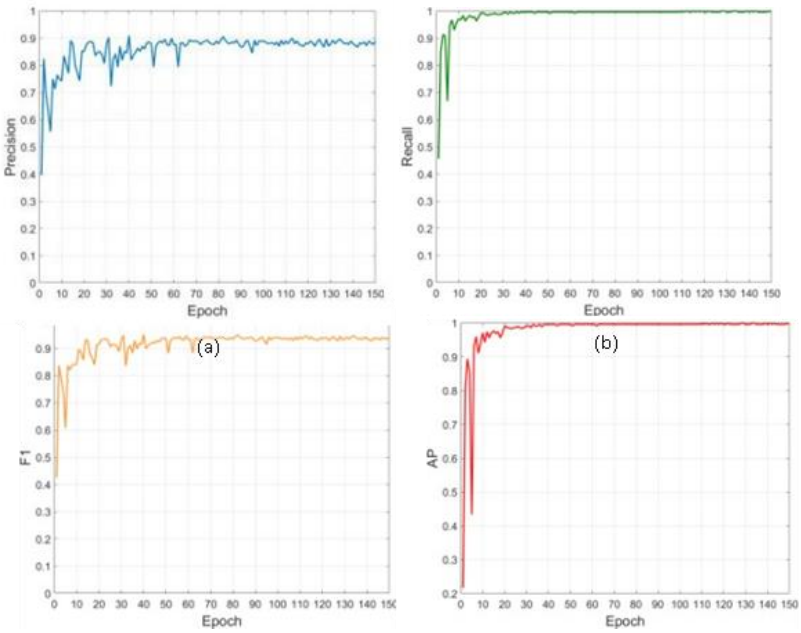
To appraise the model's generalization ability and refine it over time, precision, recall, and the F1 score wereemployed as assessment measures throughout the training phase.

During training, each iteration involved processing a batch of 4 images, resulting in a total of 835 images trained. To complete one epoch, it took 209 iterations. The validation set was utilized to assess the weight results for each epoch, allowing the model's precision and recall group to be determined based on a specified threshold. By modifying the precision and recall thresholds, various groups can be generated, which can be utilized to construct a P-R curve. The area under the P-R curve represents the AP (Average Precision) of the model. To determine the most effective training method, three groups were created, each with different maximum epochs: 150, 200, and 350. The weights corresponding to maximum AP were analyzed for each group, revealing precision and recall at a threshold of 0.5 (see Table 1). As expected, AP increased with both

epoch and iteration count and reached its highest value of 0.9996 after 27 hours in the 300-epoch group. However, a high AP of 0.9995 could be achieved in just 12.5 hours with a 150-epoch group, making it a more practical choice. Further investigation was conducted on evaluation indexes during the 150-epoch training process.

TABLE 1. The effectiveness of three distinct training.

No.	1	2	3
Epoch	150	200	350
Image size	608 x 608	608 x 608	608 x 608
Time(h)	8	14	29.3
Precision	0.8793	0.8796	0.9893
Recall	0.9965	1.0005	1.0009
F1	0.9347	0.9859	0.9714
AP	0.9967	0.9999	0.9996



(c) (d)

Fig. 4. The graph displaying the evaluation index curve after 200 epochs of training showcases (a) Precision, (b) Recall, (c) F1, and (d) AP.

In Figure 4. , the precision, recall, AP, and F1 numerical curves from the second training are depicted. Notably, the recall and AP curves exhibit swift convergence towards unity. The precision and F1 curves stabilize post the 150th epoch. Thus, the YOLOv5 algorithm-based Palm detection model is chosen as the optimum weight model from the second training.

Fig. 5. Test results under direct sunlight.

Detection results

Different lighting conditions were utilized to verify the effectiveness of the trained palm detection model. Three examples were selected for each condition, and the detection outcomes are shown in Figure 5.

Fig. 6. The results of detection perform under bright sunny backlight conditions.

Fig. 7. The outcomes of the investigation conducted in overcast weather conditions.

Accurate detection of Palms was attained under differing levels of illumination, as seen in Fig. 6. In both fully and partially illuminated conditions, and with varying light strengths, detection rates remained consistent. The results even held up in cloudy weather, with Fig. 7 showcasing the detection of both single and double Palms. Though rare, images containing up to three Palm hands were also clear and easily detected. Interestingly, these detection rates were higher compared with those found in previous studies [36], potentially because of the the higher resistance to environmental factors provided by deep learning algorithms.

(b) (c)

Fig. 8. The palm detection outcomes evaluated under varying levels of occlusion: (a) slight occlusion of the region; (b) an increase in occlusion area; (c) loss of over half of the left palm information and more than half of the left banana information.

Previous investigations on palm detection have been inconsistent because of the important size of palm branches and leaves, resulting in various levels of occlusion. Furthermore, the issue is aggravated by the fact that incomplete palms could be identified in a single image because of the different capture angles, which could also be categorized as an occlusion environment. Fig. 8 depicts a number of occlusion tests performed on the detection model. In (a), a small region of occlusion had no effect on the detection outcome. In (b), as the occlusion area increased, detection remained precise. Even with half of the left Palm missing in (c), the model still identified it confidently with a rating of 1. However, when the left and some parts of the palm were lost, detection accuracy fell to 0.61, showing insufficient information. Such occlusions are common in continuous detection, making accurate detection of all palms in successive frames important in resolving the repeated detection issue.

Fig. 9. Results obtained from detecting various types of palm.

The palm detection model developed in this study is expected to accept different species of palms, such as various sizes and types within the same scene. Fig. 9 exhibits the detection results for different palm varieties. For instance,

(a) exhibits the MUSA AA palm detection result, despite slow growth, and plant fiber and branches coverage, fruit was shown in strong light. On the other hand, (b) indicates the detection result of MUSA ABBB palm with short and dense fingers and no obvious separation in the fruit. Lastly, (c) reveals the detection result of MUSA ABB palm.

Fig. 10. Detection results of palm at different maturity.

Finally, palm fruits at different steps of development were found, as revealed in Fig.10. Single or multiple immature palm hands were precisely identified, even when different maturity palms were present in the same image. It is important to note that palm confidence is very high, with many confidence being 1, which would be compared with other models in the discussion section.

V. Discussion

To ensure the accuracy of the palm detection model in plantation, this section compares with other algorithms. The performance of the model is analyzed and compared against traditional machine learning and deep learning-based palm detection algorithms.

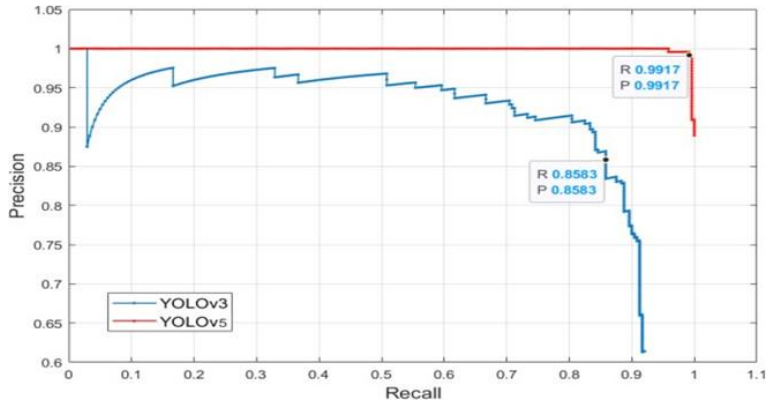


Fig. 11. The curve illustrating Precision-Recall for distinct methods of detection.

Yolov5 vs. yolov3 in palm detection

Using the YOLOv3 neural network, the dataset in this study underwent training and detection. A 300-epoch was employed, and the validation of the best training model with an AP of 0.8697 was chosen. The P-

R curves of the two methods for the validation set are depicted in Fig. 11. The YOLOv3 algorithm's P-R curve encompasses the YOLOv5 algorithm's P-R curve. The break-even point for YOLOv3 is (0.8583,0.8583), while that of YOLOv5 is (0.9917,0.9917). To improve network detection of small objects, YOLOv5 uses an FPN+PAN structure to repeatedly extract features from both the trunk and detection layer via multi-scales. Conversely, YOLOv3 employs an FPN structure in the neck area. The inclusion of multiple palm varieties in the dataset, coupled with the considerable distance between palm plants, resulted in significant variability in the size of different varieties or distances of palms within the same image. This condition leads to incomplete detection of small palm fruit by YOLOv3, resulting in low AP values. Apparently, the two detection methods reveal varying levels of small object detection, as evidenced by the detection results. Fig. 12 provides a side-by-side comparison of the detection results of different palm hands with (a) and (c) being the detection result of YOLOv3, while (b) and (d) are the detection results of YOLOv5. The latter method identifies the small palm fruit that is obscured by surrounding palms or leaveswhere YOLOv3 fails.

Fig. 12. Observed palm detection results: (a) YOLOv3 detected palm fruit occlusions; (b) palm fruit occlusions were detected using YOLOv5; (c) YOLOv3 detected leaf occlusions on the palm; and (d) leaf occlusions on the palm were detected using YOLOv5.

In palm plantations, small objects like fruits can be a challenging detection task for computer systems like YOLOv3. This algorithm failed to distinguish between a palm hand and a fruit, whereas YOLOv5's innovative structure and strategic approaches succeeded. Palm management extremely depends on precise fruit detection, as fruit sizes vary across palm varieties. Accurate small object detection can provide valuable insights for consistent production management.

Our team took on the challenge of identifying palm trees at high altitudes. Despite the ease of capturing most palms horizontally, some trees bear fruit far out of reach from the ground. Our experiments involved capturing palm images at varying elevation angles to determine if detection was possible. Fig.14 displays the detection results of YOLOv3 and YOLOv5, showing that YOLOv5 outperformed YOLOv3 by accurately detecting palm fruits. Its superior generalization abilities were evident in the comparison of the two methods, which differ in confidence. We compared the confidence levels of the palm detection algorithms in 120 images. The outcomes are illustrated in the accompanying FIGURE. YOLOv3's detection confidence ranged from 0.5 to 1.0, whereas YOLOv5's confidence level was nearly 1.0. In the event that the palm covered a more extensive region, the confidence level would be lower.

(a) (b)

Fig. 13. A contrast of detection findings captured at varying elevation angles: (a) YOLOv3, and (b) YOLOv4.

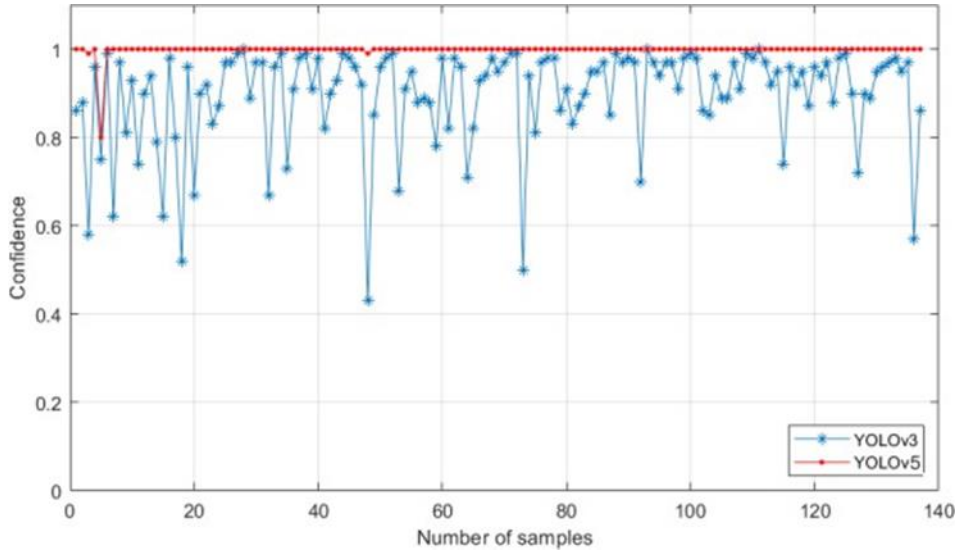


Fig. 14. Assessing the level of confidence in each detection algorithm.

After observing the comparison above, it is obvious that there was a disparity in the confidence levels of the two methods. Consequently, we proceeded to evaluate the confidence levels of palm detection in 120 images by both algorithms. The outcome is depicted in the accompanying figure. YOLOv3 had a detection confidence score ranging from 0.5 to 1.0, whereas YOLOv5 had an almost perfect score of 1.0. When the banana covered a larger area, the confidence level reduced.

A comparative analysis of deep learning and machine learning algorithms.

We carried out a comparative analysis of the palm detection outcomes of three distinct algorithms: YOLOv5, YOLOv3, and HOG+LBP+SVM. Earlier research [60] provided a detailed account of the different conditions in which the algorithms produced detection results, and the issues that machine learning algorithms encountered are explained below. Notably, in literature [60], the palm was erroneously identified as two palm hands when the key parts were obscured. YOLOv3 and YOLOv5 were subjected to experiments.

Three different algorithms were tested for MUSA AA palm detection. Despite strong illumination, the palm's few underdeveloped fingers and low contrast between fruit and leaves posed challenges. One machine learning algorithm's sliding window scale proved insufficient, leading to detection failure. YOLOv3 successfully detected a palm hand but misplaced the upper right hand, while YOLOv5 detected both hands. Our analysis included a comparison of traditional machine learning algorithms and deep learning algorithms for palm detection.When it comes to detecting palms in varying conditions, all three algorithms perform well. Table 2 compares key metrics for each algorithm in the same test set. The machine learning algorithm, compared to its deep learning counterparts, boasts a lower running cost, shorter training time, and smaller file size, and can run on a CPU without requiring a GPU. However, it takes longer to detect and has a lower detection rate. YOLOv3 and YOLOv5, on the other hand, require a GPU to run. YOLOv3's optimal model required 300 epochs of training, while YOLOv5 required only 150. Although YOLOv3's training time was less than YOLOv5's, its weight file was larger.When it comes to image detection, YOLOv3 stands out with its speedy processing time and high detection rate. However, it falls short when compared to YOLOv5, which boasts an impressive detection rate of 99.29%. YOLOv5 also has a shorter average detection time of 0.171s and a minimum detection time of 0.135s. This deeper network does lead to longer detection times than YOLOv3, but the results are worth it. Ultimately, YOLOv5 proves to be the optimal choice for achieving the best weight model.

TABLE 2. The indices of detection for the trio of algorithms.

Algorithm	**HOG+LBP+SVM**	**YOLOv3**	**YOLOv5**
Training time	2.35 h	6.32h	12.5 h
Weight file size	15.5 MB	469 MB	244 MB
Hardware platform	Intel (R) Core (TM) i7 – 5500U @2.4 GHz, 16.0 GB RAM, NVDIAGeForce 940M	Intel (R) Core (TM) I7 – 9750H @2.6 Ghz, 2.59 Ghz 16 RAM, NVDIA Geforce RTX2070 With Max Q Design	Intel (R) Core (TM) I7 – 9750H @2.6 Ghz, 2.59 Ghz 16 RAM, NVDIA Geforce RTX2070 With Max Q Design
Test set	150 images	150 images	150 images

The average detectiontime	1.725s	0.078s	0.771s
The average detection time	0.643s	0.080s	0.935s
Detection rate	89.93%	90.98%	99.99

Its exceptional confidence and detection rate rendered it superior to both the conventional machine learningalgorithm and YOLOv3 algorithm during detection, requiring fewer iterations in the training phase.

VI. Conclusions

Identifying palm fruits has been one of the most important portion of intelligent palm plantation management. Here, an approach of detecting palm trees in the natural condition was proposed using the YOLOv5 network from the most recent release. Additionally, we evaluated the ability of the conventional machine learning algorithm and another neural network algorithm in the detection of palms. The following conclusions can be derived from the experimental findings: (1) in the plantation, a deep learning algorithm for palm detection was found to be suitable for use in the deep ocean. The physical properties of the YOLOv5 neural network, as well as the primary issues with detecting palms, were investigated. The CSPDarknet53 structure in the network increases the network's complexity, this would allow for a deeper extraction of palm features and would lessen the effect of the green background on the irregular fruits; the SPP structure enhances the network's receptivity to deeper features with a lower cost of computation. Additionally, the FPN+PAN structure combines multiple-scale features to extract more profound palm semantic information and positioning information, this would allow for a greater detection of precise palm fruits. When the sizes of the palm fruits in the same image are vastly different, accurate detection can still be achieved; the DIOU nms algorithm increases the confidence of the results of palm detection. (2) The palm detection algorithm in the plantation utilizing YOLOv5 is capable of producing accurate detection in different lighting and occlusion conditions for various varieties and maturity, which offering exact information for the palm plantation intelligent management and fruit picking. (3) To identify palm trees in the plantation, a deep learning algorithm is more effective than a machine learning algorithm. The average detection time of the three algorithms was 1.325 seconds, 0.038 seconds, and 0.171 seconds, respectively, when compared to HOG+LBP+SVM, YOLOv3, and YOLOv5. The detection rates of palm were 89.63%, 90.78%, and 99.29%, respectively. YOLOv5 achieved the best weight model with fewer iterations during the training process. YOLOv5 has a high confidence and detection rate, it was also more effective than traditional machine learning algorithms like YOLOv3 in the detection stage. Ultimately, the proposed method is appropriate for detecting palm trees in a plantation.

Further research should continue to focus on the obtainment of the actual coordinate of a palm fruit in the real world as well as achievement of the palm fruit localization, and calculation of the pick point's location.

Acknowledgments

We thank friends, farmers, and other local partners for their support, contribution and assistance during the implementation of research and capacity-building activities.

References

[1] Zhang P, Guo Z, Ullah S, Melagraki G, Afantitis A, Lynch I. Nanotechnology And Artificial Intelligence To Enable Sustainable And Precision Agriculture. Nat. Plants. 2021;7(7):864–876.
[2] Dharmaraj V, Vijayanand C. Artificial Intelligence (Ai) In Agriculture. Int. J. Curr. Microbiol. Appl. Sci. 2018;7(12):2122–2128.
[3] Javaid, M.; Haleem, A.; Khan, I.H.; Suman, R. Understanding The Potential Applications Of Artificial Intelligence In Agriculture Sector. Adv. Agrochem.
[4] K. Alex, S. Ilya, And E. H. Geoffrey, ‘‘Imagenet Classification With Deep Convolutional Neural Networks,’’ In Proc. 25th Int. Conf. Neural Inf. Process. Syst., 2012, Pp. 1097–1105.
[5] K. Simonyan And A. Zisserman, ‘‘Very Deep Convolutional Networks For Large-Scale Image Recognition,’’ 2014, Arxiv:1409.1556. [Online]. Available: Http://Arxiv.Org/Abs/1409.1556
[6] K. He, X. Zhang, S. Ren, And J. Sun, ‘‘Deep Residual Learning For Image Recognition,’’ 2015, Arxiv: 1512.3385. [Online]. Available: Https://Arxiv.Org/Abs/1512.3385
[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, And A. Rabinovich, ‘‘Going Deeper With Convolutions,’’ 2014, Arxiv:1409.4842. [Online]. Available: Http://Arxiv.Org/Abs/1409.4842
[8] S. Ioffe And C. Szegedy, ‘‘Batch Normalization: Accelerating Deep Network Training By Reducing Internal Covariate Shift,’’ 2015, Arxiv: 1502.3167. [Online]. Available: Https://Arxiv.Org/Abs/1502.3167
[9] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, And Z. Wojna, ‘‘Rethinking The Inception Architecture For Computer Vision,’’ 2015, Arxiv: 1512.1567. [Online]. Available: Https://Arxiv.Org/Abs/1512.1567
[10] C. Szegedy, S. Ioffe, V. Vanhoucke, And A. Alemi, ‘‘Inception-V4, Inception-Resnet And The Impact Of Residual Connections On Learning,’’ 2016, Arxiv: 1602.7261. [Online]. Available: Https://Arxiv.Org/Abs/ 1602.7261
[11] F. Chollet, ‘‘Xception: Deep Learning With Depthwise Separable Convolutions,’’ 2016, Arxiv: 1610.2357. [Online]. Available: Https://Arxiv.Org/ Abs/1610.2357
[12] R. Girshick, J. Donahue, T. Darrell, And J. Malik, ‘‘Rich Feature Hierarchies For Accurate Object Detection And Semantic Segmentation,’’ 2013, Arxiv:1311.2524. [Online]. Available: Http://Arxiv.Org/Abs/1311.2524
[13] R. Girshick, ‘‘Fast R-Cnn,’’ 2015, Arxiv: 1504.8083. [Online]. Available: Https://Arxiv.Org/Abs/1504.8083