Ablation studies on YOLOFruit detection algorithm for fruit harvesting robot using deep learning

O M Lawal*, Z Huamin and Z Fan
Shanxi Agricultural University, Taigu, Jinhong City, Shanxi, 030801, China.

*Email: olarewaju_lawal@sxau.edu.cn

Abstract. Fruit detection algorithm as an integral part of harvesting robot is expected to be robust, accurate, and fast against environmental factors such as occlusion by stem and leaves, uneven illumination, overlapping fruit and many more. For this reason, this paper explored and compared ablation studies on proposed YOLOFruit, YOLOv4, and YOLOv5 detection algorithms. The final selected YOLOFruit algorithm used ResNet43 backbone with Combined activation function for feature extraction, Spatial Pyramid Pooling Network (SPPNet) for detection accuracies, Feature Pyramid Network (FPN) for feature pyramids, Distance Intersection Over Union−Non Maximum Suppression (DIoU−NMS) for detection efficiency and accuracy, and Complete Intersection Over Union (CIoU) loss for faster and better performance. The obtained results showed that the average detection accuracy of YOLOFruit at 86.2% is 1% greater than YOLOv4 at 85.2% and 4.3% higher than YOLOv5 at 81.9%, while the detection time of YOLOFruit at 11.9ms is faster than YOLOv4 at 16.6ms, but not with YOLOv5 at 2.7ms. Hence, the YOLOFruit detection algorithm is highly prospective for better generalization and real–time fruit detection.

1. Introduction
The application of agricultural robots for fruits and vegetables harvesting has increased over the years due to the increasing demand for food and expensive manual labor cost [1]. Meanwhile, a fruit harvesting robot requires fruit detection algorithm as an integral part before a manipulator is guided to pick or harvest the fruits. This fruit detection performance depends on accuracy and speed. In addition, the fruit detection is also influenced by some factors such as occlusions, illumination variation, similar fruit appearance as background and so on. Therefore, it is necessary to develop a robust fruit detection algorithm that is accurate, fast, and can overcome these factors.

Deep learning techniques have made considerable progress in addressing the challenges including factors faced by fruit detection, and have been studied by Koirala et al. (2019) [2], Shi et al. (2020) [3], Liu et al. (2020) [4], Kirk et al. (2020) [5], Li et al. (2020) [6], and many more. Among the two main deep learning object detector studied so far, single–stage such as You Only Look Once (YOLO) and Single Shot Detector (SSD) [7] was reported by Koirala et al. (2019) [8] to be faster than two–stage such as Faster R–CNN [9] with similar accuracy. Although, accuracy is important to fruit detection performance but speed is much required by harvesting robots. For this reason, single–stage deep learning detector is the best candidate for fruit detection. It was proven by Liu et al. (2020) [10] that the application of YOLOv3 [11] algorithm as one of the version of YOLO for fruit detection can solve illumination variation and occlusion factors. The paper reported average precision (AP) of
93.91%, $F_1$ score of 96.4%, and detection time of 54ms on tomato detection. Lawal (2021a) [12] further improved on the tomato detection based on modified YOLOv3 algorithm to achieved AP of 99.3% and detection time of 44ms. Notwithstanding, both papers recorded a slower detection speed using DenseNet backbone. Zheng et al. (2019) [13] also applied YOLOv3 on different type of fruits to achieved an average AP of 91.44% detection accuracy.

However, the detection speed of the tested YOLOv3 algorithm was not more 40frame per second (~25ms) and likely to experience missed fruit detection in highly occluded condition. The quest for YOLOv3 network improvements as suggested by Zheng et al. (2019) [13] lead to YOLOv4 [14] introduction. The detection accuracy and speed of YOLOv3 was respectively improved by 10% and 12% in YOLOv4. For comparison, YOLOv3 uses DarkNet53 backbone with Leaky ReLU [15] activation and feature pyramid network (FPN) as Neck [16], while YOLOv4 is composed of CSPDarknet53 backbone with Mish activation [17], spatial pyramid pooling (SPPNet) (He et al., 2015) [18], path aggregation network (PANet) as Neck [19] and YOLOv3 Head.

Recently, Jocher et al. (2020) [20] proposed YOLOv5 for single−stage object detection. YOLOv5 adopts Cross stage partial networks (Focus−CSPNet) backbone for features extraction, PANet to get feature pyramids, SPPNet to avoid missed object detection, and YOLOv3 Head for final detection. Different configuration files namely YOLOv5s, YOLOv5m, YOLOv5l and YOLOv5x at different depth and width multiple were provided in YOLOv5 repository. YOLOv5 showed significant performance improvement particularly in detection speed compared to YOLOv4 as reported, but still in development stage and no official paper has been published. Consequently, the application and optimization of YOLO architecture through ablation studies for speed and accuracy is a contribution to the development of a robust fruit detection algorithm.

An ablation study is simply removing some features of detection algorithm and seeing the effects on performance. For example, the long overdue residual block arrangement of DarkNet53 and CSPDarkNet53 backbone which is 1,2,8,8,4, and that of Focus−CSPNet backbone which is 1,3,9,9,3 could be replaced with new block arrangement or probably replacing the activation function, backbone, Neck and so on with another in order to bring improvement to the fruit detection algorithm.

Apart from the fact that YOLOv4 and YOLOv5 draws their backbone experience from CSPNet shown in Fig. 1(a) to enhance learning ability of CNN, ResNet framework proposed by He et al. (2016) [21] offers skip or shortcut connection as shown in figure 1(b) to solve drops off from saturated accuracy for deeper neural network and number of parameters reduction without network performance degradation. With the reduction in parameters, the detection speed of applied ResNet architecture becomes faster. According to Zheng et al. (2019) [13], ResNet is an excellent object detection compared to DenseNet, VGGNet, SqueezeNet, and InceptionV4. Lawal (2021b) [22] experimented ResNet43 with YOLOv4 Head for muskmelon detection and reported AP of 89.6%. Nevertheless, the performance of proposed YOLOMuskmelon algorithm cannot be ascertained due to its one−class application. Additionally, the ablation studies on fruit detection algorithm using ResNet backbone still remain limited.

![Figure 1. Structure of (a) YOLOv4 and YOLOv5 backbone draws from CSPNet (b) ResNet [21]](image-url)
This paper explored ablation studies on YOLOFruit detection in quest for a robust algorithm that is accurate and fast to solve the problems experienced by fruit detection. The YOLOFruit algorithm incorporated 2,3,4,3,2 block of ResNet43 backbone with combined activation (Leaky ReLU and HardMish), SPPNet, FPN, complete Intersection over Union (CIoU) loss, and distance Intersection over Union–Non Maximum Suppression (DIOU–NMS) [23] to improve fruit detection performance. The ablation studies on YOLOv4 and YOLOv5 compared with YOLOFruit showed that YOLOFruit can achieve an impressive detection accuracy and real–time detection speed. This ablation studies provides an effective reference for relevant research in terms of strategies for obtaining an optimal accuracy and detection speed.

The remainder of this paper is organized as follows: Section 2 describes the datasets and proposes the fruit detection algorithm. Section 3 explains the algorithm tested results and discussion, and Section 4 draws conclusions with future work.

2. Methodology

2.1. Datasets

The muskmelon and fig images used in this paper were obtained from greenhouse in wanghaizhuang village, Houcheng town, Taigu County, Shanxi Province, China and captured using digital camera with resolution of 3968×2976 pixels under natural daylight conditions, including overlap disturbance, leaf occlusion, stem occlusion, and illumination variation. Some of the captured images from the dataset are shown in Fig. 2. A total of 410 muskmelon and 382 fig images were captured, stored in JPG format and randomly divided into 80% train set, 15% valid set, and 5% test set. Except for the test set, a labelImg graphical image annotation tool was adopted to manually hand label all the ground truth bounding boxes.

The bounding boxes were drawn by the supposed shape as depicted in figure 2 without minding the growing conditions of the images. The annotation files were saved in YOLO format that contains object class, coordinates, height and width. For the muskmelon, a total of 838 bounding boxes were derived from the 330 images in the train set and 198 bounding boxes from 62 images in the valid set, while for fig, a total of 3513 bounding boxes were drew from the 306 images in the train set and 748 bounding boxes from 58 images in the valid set. Finally, the status of the annotated images was thoroughly checked to ensure proper annotation and no missing out class.

![Figure 2. Annotation principle of different image conditions from the dataset](image)

(a) fig with similar image background, (b) muskmelon under highly occluded condition, (c) fig with leave occlusion (d) muskmelon with both leave and stem occlusion.
2.2. YOLOFruit algorithm
The proposed YOLOFruit detection algorithm shown in figure 3 has 43-layers of ResNet43 as backbone. The ResNet43 backbone was arranged as 2,3,4,3,2 residual block in order to improve the detection speed and accuracy, and also to avoid the problem of neuron death due to gradual deepening of CNN training. ResNet framework integration into YOLOFruit algorithm was to solve network degradation problem, allow training of deeper networks, accelerates the training speed, and foster faster network convergence. The 1x1 convolution layer within the ResNet43 backbone was used to reduce complexity and parameters number reduction without performance degradation. Leaky ReLU [15] activation function was applied to the 1x1 convolution layer, because it overcomes the dying ReLU, speeds up training and provide better performance.

At the YOLOFruit detection algorithm, FPN was used as the Neck to get feature pyramids, which enables a well-generalized model on object scaling. The original front detection layers (FDL×3) adopted by YOLOv3 [11] was pruned to FDL×2 in the YOLOFruit algorithm for speed detection enhancement. Furthermore, CIoU loss function [23] for faster convergence and better performance, and DIoU−NMS (non-maximum suppression) for removing redundant detections of multiple bounding boxes to find the best match were applied to the YOLOFruit detection algorithm. According to Zheng et al. (2019) [23], DIoU−NMS considered overlap area and distance between two central points of bounding boxes to select one entity bounding box out of many overlapping bounding boxes.

At the same time, the 3x3 convolution layer within same backbone residual block was activated with HardMish due to its different pattern of approach as shown in figure 4 compared to other activation functions. The used of Leaky ReLU and HardMish within the same residual block of ResNet43 backbone becomes a combined activation function, thus to enhance the expression ability of mapped features. Activation function generally introduces non-linearity into deep neural network to improve performance. The SPPNet [18] shown in figure 3 was introduced after the ResNet43 backbone. This is for the optimization of fruit detection. SPPNet is simply a feature enhancement module that extracts the main information of the feature map and performs stitching. It helps to avoid inaccuracies and loss of target detection, particularly when the extracted feature map is blurred. SPPNet also extract both the multiscale global and local features of the same detection stage.

Figure 3. Overview of YOLOFruit detection algorithm

At the same time, the 3x3 convolution layer within same backbone residual block was activated with HardMish due to its different pattern of approach as shown in figure 4 compared to other activation functions. The used of Leaky ReLU and HardMish within the same residual block of ResNet43 backbone becomes a combined activation function, thus to enhance the expression ability of mapped features. Activation function generally introduces non-linearity into deep neural network to improve performance. The SPPNet [18] shown in figure 3 was introduced after the ResNet43 backbone. This is for the optimization of fruit detection. SPPNet is simply a feature enhancement module that extracts the main information of the feature map and performs stitching. It helps to avoid inaccuracies and loss of target detection, particularly when the extracted feature map is blurred. SPPNet also extract both the multiscale global and local features of the same detection stage.
2.3. Ablation experimental setup
The experimental setup including training and testing were implemented on Intel computer with Core i7-8700 CPU @ 3.20 GHz, 16 GB RAM, NVIDIA GeForce GTX 1080Ti GPU, CUDA v10.2, cuDNN v7.6.5, OpenCV v4.2.0. The ablation studies on different modification of YOLOFruit and YOLOv4 detection algorithms were experimented on DarkNet platform, while the different modification on YOLOv5 algorithm was implemented on PyTorch platform.

Before the different modifications of YOLOFruit and YOLOv4 algorithm training, the generated nine anchor boxes using k−mean clustering were counted from the dataset and assigned to their configuration files according to three scales of detection layer (52×52, 26×26, 13×13 feature). The anchor boxes were assigned in descending order of dimension from the first scale to the third scale in order to improve the fruit detection algorithms. The algorithms receive image inputs of 416×416 pixels, learning rate of 0.001 to reduce training loss, iterations between 0 and 4000, momentum of 0.9, weight decay of 0.0005, and batch and subdivision of 64 and 32 respectively to reduce memory usage. However, the different modification of YOLOv5 used auto−anchor generation. Trained for 300 epochs, initial learning rate of 0.01, momentum of 0.937, weight decay of 0.0005, and also image inputs of 416×416 pixels. Meanwhile, random initialization method was used to initialize the weights for training all the different modification of algorithms in order to maintain consistency. The algorithms were trained and tested according to the details presented in table 1.

The Precision, Recall, F1-score and Average Precision−AP were used to evaluate the trained algorithms. Precision is the ratio of the number of correctly detected fruit to the total number of detected fruits, Recall is the ratio of the number of correctly detected fruit to the total number of fruit in the dataset, F1-score is the trade-off between the Precision and Recall to show the performance of algorithms, AP describes the overall performance of trained algorithms under different confidence thresholds. The calculated parameters can be defined as:

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (1)
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (2)
\]
Recall Precision
\[ F_1 = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \] (3)

\[ \text{AP} = \sum_{n} (r_{n+1} - r_n) \max_{\tilde{r}} p(\tilde{r}) \] (4)

where, TP is True Positive (correct detections), FN is False Negative (missed detections), and FP is False Positive (incorrect detections), \( p(\tilde{r}) \) is the measured Precision at Recall \( \tilde{r} \)

**Table 1. Summary of trained algorithms**

| Algorithm         | Block                  | Backbone  | Activation | SPP | Neck       | Loss |
|-------------------|------------------------|-----------|------------|-----|------------|------|
|YOLOv4             | 1,2,8,8,4              | CSPDarkNet53 | Mish       | √   | PANet      | CIoU |
|YOLOv4FPN          | 1,2,8,8,4              | CSPDarkNet53 | Mish       | √   | FPN        | CIoU |
|YOLOv4RPANet       | 2,3,4,3,2              | CSPDarkNet35 | Mish       | √   | PANet      | CIoU |
|YOLOv4RFPN         | 2,3,4,3,2              | CSPDarkNet35 | Mish       | √   | FPN        | CIoU |
|YOLOv4RCombined    | 2,3,4,3,2              | CSPDarkNet35 | Combined   | √   | FPN        | CIoU |
|YOLOv5             | 1,3,9,9,3              | Focus–CSPNet | Combined   | √   | PANet      | GIoU |
|YOLOv5sR           | 1,3,4,3,2              | Focus–CSPNet | Combined   | √   | PANet      | GIoU |
|YOLOv5sFPN         | 1,3,9,9,3              | Focus–CSPNet | Combined   | √   | FPN        | GIoU |
|YOLOv5sRFPN        | 1,3,4,3,2              | Focus–CSPNet | Combined   | √   | FPN        | GIoU |
|YOLOResNet70Leaky  | 1,2,8,8,4              | ResNet70   | Leaky      | √   | FPN        | CIoU |
|YOLOResNet43Leaky  | 2,3,4,3,2              | ResNet43   | Leaky      | √   | FPN        | CIoU |
|YOLOResNet43Mish   | 2,3,4,3,2              | ResNet43   | Mish       | √   | FPN        | CIoU |
|YOLOResNet43PANet  | 2,3,4,3,2              | ResNet43   | Leaky      | √   | PANet      | CIoU |
|YOLOResNet43ReLU   | 2,3,4,3,2              | ResNet43   | ReLU       | √   | FPN        | CIoU |
|YOLOResNet43Swish  | 2,3,4,3,2              | ResNet43   | Swish      | √   | FPN        | CIoU |
|YOLOResNet43Combined | 2,3,4,3,2            | ResNet43   | Combined   | √   | FPN        | CIoU |

*Combined – Leaky at 1×1 convolution and HardMish at 3×3 convolution layer used for YOLOFruit and YOLOv4
*Combined – Leaky at 1×1 convolution (CSPbottleneck) and HardSwish at 3×3 convolution layer for YOLOv5

3. Results and Discussion

3.1. Ablation on YOLOFruit

The ablation studies on different modification of YOLOFruit is necessary to investigate the features responsible for the detection performance improvement. The ablation average findings on YOLOFruit are presented in table 2. The obtained results showed variation of average \( F_1 \) score between the algorithms. However, the average AP in table 2 is considered more accurate than \( F_1 \) score, due to its global Precision–Recall relationship. The AP of YOLOResNet43Leaky with 85.9% is more than YOLOResNet70Leaky with 84.9%. This indicated that the new residual block arrangement of 2,3,4,3,2 is 1% more accurate than the 1,2,8,8,4. The compared AP of YOLOResNet43Mish at 85.8% is 0.1% less than YOLOResNet43Leaky.

An indication that Leaky activation function is better compatible with ResNet architecture. The incorporation of PANet to YOLOResNet43PANet increased the AP to 86.5%, which is just 0.6% higher than YOLOResNet43Leaky with FPN Neck. Meanwhile, the YOLOResNet43ReLU of AP at
85.3% is 0.6% less than YOLOResNet43Leaky, while the AP of YOLOResNet43Swish at 85.9% shows similar result as YOLOResNet43Leaky. The applied Combined activation function to YOLOResNet43Combined improved the AP performance to 86.2%, which is 0.3% greater than YOLOResNet43Leaky and 0.3% less than YOLOResNet43PANet. In this regard, the level of AP outcome is measure as YOLOResNet43PANet is greater than YOLOResNet43Combined, YOLOResNet43Leaky is equal to YOLOResNet43Swish and greater than YOLOResNet43Mish, YOLOResNet43ReLU, and YOLOResNet70Leaky.

The Precision−Recall (P−R curves) of the tested different modifications on YOLOFruit are displayed in figure 5(a) (under muskmelon) and figure 5(b) (under fig) for justification to the obtained findings in table 2. A better performed P−R curve is expected to have a greater area under curve (AUC). As the rest of algorithms falls below the selected, the P−R curve under muskmelon for YOLOResNet43PANet is 85.8% and YOLOResNet43Combined is 84.3% while under fig, YOLOResNet43PANet is 87.2% and YOLOResNet43Combined is 88.1%. YOLOResNet43Combined tends to show better AUC detecting fig target and YOLOResNet43PANet performed better with detected muskmelon target, which requires future investigation. Furthermore, it was noted that the P−R curves under fig detection for all modified YOLOFruit algorithms is higher than under muskmelon detection.

The obtained model size for different modification of YOLOFruit from table 2 shows that YOLOResNet70Leaky is the highest with 143MB, followed by YOLOResNet43PANet with 136MB, while YOLOResNet43Leaky, YOLOResNet43Mish, YOLOResNet43ReLU, YOLOResNet43Swish, and YOLOResNet43Combined have similar size of 98.2MB. The detection time is a function of algorithm weight size. Table 2 shows that an increased in weight size means an increased in detection time. The fruit detection time of YOLOResNet43Leaky, YOLOResNet43ReLU, and YOLOResNet43Combined is 11.9ms, while YOLOResNet70Leaky is 15.0ms, YOLOResNet43Mish is 12.4ms, YOLOResNet43PANet is 14.5ms, and YOLOResNet43Swish is 12.4ms. Algorithm with the smallest detection time is regarded as the fastest, which can also to be converted to frame per second (~fps = 1/detection time).

Because agricultural harvesting robot requires both detection speed and accuracy, YOLOResNet43Combined having an average accuracy of 86.2% and detection time of 11.9ms outperformed other detection algorithms, and stand selected to represent our YOLOFruit detection algorithm. Meanwhile, the YOLOFruit algorithm detected number of target fruit in the tested images shown in figure 6 without missed detection (SPPNet) [18], and showed robustness under different environment. Furthermore, the compared AP of muskmelon detected under YOLOFruit algorithm at 84.3% is 5.3% less than the proposed YOLOMuskmelon at 89.6% [22]. This is due to the feature complexity from more than one−class detection. Adding more fruit type class to the YOLOFruit detection algorithm is likely to constitute a decreasing accuracy with better performance reliability. This phenomenon requires future investigation.

| YOLOFruit        | Size(MB) | P (%) | R (%) | F₁ (%) | AP (%) | Time (ms) |
|------------------|----------|-------|-------|--------|--------|-----------|
| YOLOResNet70Leaky| 143.0    | 80.0  | 83.0  | 81.0   | 84.9   | 15.0      |
| YOLOResNet43Leaky| 98.2     | 71.0  | 85.0  | 77.0   | 85.9   | 11.9      |
| YOLOResNet43Mish | 98.2     | 79.0  | 84.0  | 82.0   | 85.8   | 12.4      |
| YOLOResNet43PANet| 136.0    | 83.0  | 82.0  | 83.0   | 86.5   | 14.5      |
| YOLOResNet43ReLU| 98.2     | 81.0  | 82.0  | 82.0   | 85.3   | 11.9      |
| YOLOResNet43Swish| 98.2     | 83.0  | 80.0  | 82.0   | 85.9   | 12.4      |
| YOLOResNet43Combined| 98.2 | 76.0  | 85.0  | 80.0   | 86.2   | 11.9      |
3.2. Ablation on YOLOv4

The ablation studies on different modification of YOLOv4 algorithm was based on Neck that is PANet and FPN and activation function of Mish and Combined. Table 3 shows that the average AP of YOLOv4RPANet at 86.7% is 2% more than YOLOv4PANet at 84.7% under the same Mish and PANet incorporation. Meanwhile, under Mish and FPN influence, the AP of YOLOv4RFPN at 85.2% is 1.1% greater than YOLOv4FPN at 84.1%. This further proved that the application of residual block arrangement of 2,3,4,3,2 is more accurate than 1,2,8,8,4. Nevertheless, the AP of YOLOv4Combined at 83.5% is the least among the algorithms. Meaning that the proposed Combined activation function is incompatible with CSPDarkNet backbone for performance improvement. The P–R curve under detected muskmelon in figure 7(a) shows YOLOv4RPANet outperforming other algorithms while under detected fig, YOLOv4Combined becomes the dominant algorithm in figure 7. Just like figure 5, the obtained AP results of all tested algorithms in figure 7(b) is higher than that of figure 7(a) indicating an effectiveness in smaller target detection against bigger target.

Furthermore, PANet addition to the algorithm constituted an increased in weight size compared to FPN. As the weight size of YOLOv4RFPN and YOLOv4RCCombined is at 144MB, the size of
YOLOv4PANet at 244MB is more than YOLOv4FPN at 179MB and YOLOv4RPANet at 209MB. The detection time depends on the weight size. YOLOv4RFPN and YOLOv4RCombined with the least size have detection time of 16.6ms and 16.5ms respectively. With this size effect, the detection time of YOLOv4PANet is 22.3ms, YOLOv4FPN is 18.4ms, and YOLOv4RPANet is 20.6ms. Comparing all the algorithms for the best performance, YOLOv4RFPN algorithm with the detection accuracy of 85.2% and detection time of 16.6ms is outstanding. Therefore, YOLOv4RFPN algorithm represent YOLOv4 detection algorithm in this section. Meanwhile, the visualization outcome of the tested images is similar to the presented results in figure 6, except for the difference in their confidence scores.

| YOLOv4    | Size(MB) | P (%) | R (%) | F1 (%) | AP (%) | Time (ms) |
|-----------|----------|-------|-------|--------|--------|-----------|
| YOLOv4PANet | 244.0    | 74.0  | 83.0  | 78.0   | 84.7   | 22.3      |
| YOLOv4FPN  | 179.0    | 71.0  | 83.0  | 77.0   | 84.1   | 18.4      |
| YOLOv4RPANet | 209.0    | 82.0  | 83.0  | 82.0   | 86.7   | 20.6      |
| YOLOv4RFPN | 144.0    | 79.0  | 83.0  | 81.0   | 85.2   | 16.6      |
| YOLOv4RCombined | 144.0    | 79.0  | 83.0  | 81.0   | 83.5   | 16.5      |

Figure 7. P–R curve of ablation on YOLOv4 under (a) muskmelon (b) fig detection

3.3. Ablation on YOLOv5

It is no doubt that the weight size of YOLOv5s at 14MB is smaller compared to YOLOv5m at 41.3MB, YOLOv5l at 90.8MB, and YOLOv5x at 169MB as reported by Jocher et al. (2020). Nevertheless, the further modification of YOLOv5s to YOLOv5sR reduces the size to 13.2MB and the addition of FPN Neck instead of PANet makes YOLOv5sFPN equal to 12.0MB and YOLOv5sRFPN equal to 11.2MB as indicated in table 4. Like previous explanations, an increase in algorithm size indicates a rise in detection time. The ablation findings on different modification of YOLOv5 shows that YOLOv5sRFPN having detection time of 2.7ms is faster than other algorithms.

The modification of YOLOv5 was only based on backbone and Neck. Among other algorithms, the average detection accuracy of YOLOv5s is 82.7% greater. Contrary to YOLOFruit and YOLOv4 modification, the average AP of YOLOv5s at 82.7% was reduced to 80.9% in YOLOv5sR, and was further decreased to 80.6% in YOLOv5sFPN. Interestingly, the AP gained a rise to 81.9% in YOLOv5sRFPN, which must be investigated in the future. The obtained P–R curve under both detected muskmelon (figure 8(a)) and fig (figure 8(b)) showed a contrary findings compared to
YOLOFruit and YOLOv4 modification, where the AUC experienced in targeted muskmelon is slightly higher than the fig for all the algorithms.

| Table 4. Ablation average results on YOLOv5 detection algorithm |
|---------------------------------------------------------------|
| YOLOv5 | Size(MB) | P (%) | R (%) | F (%) | AP (%) | Time (ms) |
| YOLOv5s | 14.0 | 72.1 | 82.3 | 76.9 | 82.7 | 3.1 |
| YOLOv5m | 41.3 | 70.6 | 79.9 | 74.9 | 80.1 | 6.6 |
| YOLOv5l | 90.8 | 67.2 | 80.9 | 73.4 | 82.2 | 11.3 |
| YOLOv5x | 169.0 | 71.8 | 79.8 | 75.6 | 81.3 | 18.9 |
| YOLOv5sR | 13.2 | 53.6 | 82.1 | 64.9 | 80.9 | 2.9 |
| YOLOv5sFPN | 12.0 | 51.6 | 80.2 | 62.8 | 80.6 | 2.8 |
| YOLOv5sRFPN | 11.2 | 54.2 | 83.1 | 65.6 | 81.9 | 2.7 |

Figure 8. P–R curve of ablation on YOLOv5 under (a) muskmelon (b) fig detection

Figure 9. Tested images under YOLOv5 detection algorithm
Meanwhile, the P–R curves under muskmelon and fig for YOLOv5s is 83.8% and 81.7% respectively greater than other algorithms. However, YOLOv5sRFPN at 82.6% for P–R curve under muskmelon and 81.3% for P–R curve under fig is a strong contender for YOLOv5. Based on the detection time of 2.7ms and average accuracy of 81.9% achieved by YOLOv5sRFPN, YOLOv5sRFPN is 12.9% faster and 0.8% less accurate than YOLOv5s. This indicated that the performance of YOLOv5sRFPN is better than YOLOv5s and outstanding to represent YOLOv5 in this section. The tested image batches on this YOLOv5 detection algorithm are presented in figure 9 for visualization without missed detection [24].

3.4. Detection algorithms comparison
The outstanding results derived from different modification of YOLOFruit in section (3.1), YOLOv4 in section (3.2), and YOLOv5 in section (3.3) were compared as detailed in table 4, figure 13, and figure 14. YOLOFruit, YOLOv4 and YOLOv5 respectively represents YOLOResNet43Combined, YOLOv4RFPN and YOLOv5sRFPN. Table 5 shows that the detection accuracy of YOLOFruit at 86.2% is 1% higher than YOLOv4 at 85.2% and 4.3% greater than YOLOv5 at 81.9%. In addition, the P–R curves under both muskmelons in figure 10(a) and fig in figure 10(b) indicated that the YOLOFruit detection algorithm is 84.3% and 88.1% respectively more than YOLOv4 and YOLOv5. This indicated that the Combined activated ResNet43 backbone of YOLOFruit is more accurate than the Mish activated CSPDarkNet of YOLOv4RFPN and Combined activated Focus–CSPNet backbone of YOLOv5sRFPN.

The compared weight size shows YOLOv5 at 11.2MB with the least, which weighs smaller than YOLOFruit at 98.2MB and YOLOv4 at 144MB. With this, the detection time of YOLOv5 at 2.7ms is faster than YOLOFruit at 11.9ms and YOLOv4 at 16.6ms. This is a case of trade–off between detection accuracy and speed. Therefore, the detection algorithm of YOLOFruit is more accurate than YOLOv4 and YOLOv5, faster than YOLOv4, but slower than YOLOv5. YOLOFruit algorithm can generalize better and perform well for real–time detection, which is applicable for agriculture harvesting or picking robots.

Table 5. Compared average results of detection algorithms

| Algorithm   | Size(MB) | P (%) | R (%) | F1 (%) | AP (%) | Time (ms) |
|-------------|----------|-------|-------|--------|--------|-----------|
| YOLOFruit   | 98.2     | 76.0  | 85.0  | 80.0   | 86.2   | 11.9      |
| YOLOv4      | 144.0    | 79.0  | 83.0  | 81.0   | 85.2   | 16.6      |
| YOLOv5      | 11.2     | 54.2  | 83.1  | 65.6   | 81.9   | 2.7       |

Figure 10. Compared P–R curve of algorithms under (a) muskmelon (b) fig detection
4. Conclusions and future work
The ablation studies on proposed YOLOFruit detection algorithm was explored in this paper in search for accuracy and speed to solve fruit detection challenges. The ablation studies focused on the backbone, Neck and activation function of detection algorithms. The proposed YOLOFruit incorporated Combined activated ResNet43 backbone for feature extraction, SPPNet for detection accuracies, FPN for feature pyramids, DIoU−NMS for detection efficiency and accuracy, and CIoU loss for faster and better performance. The ablation findings showed that the residual block 2,3,4,3,2 arrangements of ResNet43 backbone performed better than 1,2,8,8,4 of ResNet70 backbone. The introduced Combined activation function that is composed of Leaky ReLU and HardMish activated with ResNet43 backbone outperformed Leaky, ReLU, Mish and Swish activated function.

The compared results on different modification of algorithms showed that the average detection accuracy of YOLOFruit at 86.2% is 1% more than YOLOv4 at 85.2% and 4.3% higher than YOLOv5 at 81.9%, while the detection time of YOLOv5 at 2.7ms is faster than YOLOFruit at 11.9ms and YOLOv4 at 16.6ms. Therefore, YOLOFruit algorithm is highly prospective for better generalization and real–time fruit detection, and for picking/harvesting robot application. In the future, other backbones architecture such as MobileNet, DenseNet, EfficientNet and so on would be considered, including other existing algorithm Neck and activation functions, particularly combined technique so as to obtained an improved detection performance.

References
[1] Sa I, Ge Z, Dayoub F, Upcroft B, Perez T and McCool C 2016. Deepfruits: A fruit detection system using deep neural networks. Sensors, 16: 1222.
[2] Koirala A, Walsh K B, Wang Z and McCarthy C 2019. Deep learning – Method overview and review of use for fruit detection and yield estimation. Comput. Electron. Agric., 162: 219–234.
[3] Shi R, Li T and Yamaguchi Y 2020. An attribution-based pruning method for real-time mango detection with YOLO network. Comput. Electron. Agric., 169: 105214.
[4] Liu Z, Wu J, Fu L, Majeed Y, Feng Y, Li R, et al. 2020. Improved kiwifruit detection using pre-trained VGG16 with RGB and NIR information fusion. IEEE Access, 8: 2327–2336.
[5] Kirk R, Cielniak G and Mangan M 2020. Lab fruits: A rapid and robust outdoor fruit detection system combining bio-inspired features with one-stage deep learning networks. Sensors, 20 (1): 275.
[6] Li J, Tang Y, Zou X, Lin G and Wang H 2020. Detection of fruit-bearing branches and localization of litchi clusters for vision-based harvesting robots. IEEE Access, 8: 117746–117758.
[7] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY and Berg AC 2016. Ssd: single shot multi box detector. European conference on computer vision. Springer, pp 21–37.
[8] Koirala A, Walsh KB, Wang Z and McCarthy C 2019. Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of ‘MangoYOLO’. Precision Agriculture 20, 1107–1135.
[9] Ren S, He K, Girshick R and Sun J 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the International Conference on Neural Information Processing Systems 28, Montreal, QC, Canada, pp 91–99.
[10] Liu G, Nouaze JC, Touko PL and Kim JH 2020. Yolo-tomato: a robust algorithm for tomato detection based on yolov3. Sensors, 20: 2145.
[11] Redmon J and Farhadi A 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.
[12] Lawal MO 2021a. Tomato detection based on modified YOLOv3 framework. Sci Rep, 11: 1447.
[13] Zheng YY, Kong JL, Jin XB, Wang XY, Su TL and Zuo M 2019. CropDeep: The crop vision dataset for deep learning based classification and detection in precision agriculture. Sensors, 19: 1058.

[14] Bochkovskiy A, Wang CY and Liao HYM 2020. YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv: 2004.10934v1.

[15] Maas AL, Hannun AY and Ng AY 2013. Rectifier nonlinearities improve neural network acoustic models. International Conference on Machine Learning (ICML), pp 1.

[16] Lin TY, Dollár P, Girshick R, He K, Hariharan B and Belongie S 2017. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, pp 2117–2125.

[17] Misra D 2019. Mish: A self-regularized nonmonotonic neural activation function. arXiv preprint arXiv:1908.08681.

[18] He K, Zhang X, Ren S and Sun J 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 37: 1904–1916.

[19] Liu S, Qi L, Qin H, Shi J and Jia J 2018. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 8759–8768.

[20] Jocher G, Stoken A, Borovec J, et al. 2020. ultralytics/yolov5: v3.0 (Version v3.0). Zenodo.

[21] He K, Zhang X, Ren S and Sun J 2016. Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, pp 770–778.

[22] Lawal MO 2021b. YOLOMuskmelon: Quest for Fruit Detection Speed and Accuracy Using Deep Learning. IEEE Access, 9: 15221–15227.

[23] Zheng Z, Wang P, Liu W, Li J, Ye R and Ren D 2019. Distance-IoU Loss: Faster and better learning for bounding box regression. arXiv preprint, arXiv: 1911.08287v1.

[24] Kusumiyati, Hadiwijaya Y, Putri I E and Munawar A A 2021 Multi-product calibration model for soluble solids and water content quantification in Cucurbitaceae family, using visible/near-infrared spectroscopy Heliyon 7 e07677.