IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 7, 2021, accepted January 18, 2021, date of publication January 20, 2021, date of current version January 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3053167

YOLOMuskmelon: Quest for Fruit Detection Speed and Accuracy Using Deep Learning

OLAREWAJU M. LAWAL, (Member, IEEE)
College of Agricultural Engineering, Shanxi Agricultural University, Jinzhong 030801, China
e-mail: olarewajulawal@yahoo.com

This work was supported in part by the Natural Science Foundation of Shanxi Province, and in part by the Shanxi Agricultural University, China, under Grant 2020BQ34.

ABSTRACT Fruit detection plays a vital role in harvesting robot platforms. However, complicated environment attributes such as illumination variation, occlusion, have made fruit detection a challenging task. A robust YOLOMuskmelon model that is accurate and fast was proposed to solve detection difficulties. The YOLOMuskmelon model incorporated ReLU activated ResNet43 backbone with new 2,3,4,3,2 residual block arrangement, spatial pyramid pooling(SPP), complete Intersection over Union (CIoU) loss, feature pyramid network(FPN), and distance Intersection over Union−Non Maximum Suppression(DIoU−NMS) to improve detection performance. The obtained average precision (AP) results of YOLOMuskmelon at 89.6% is greater than YOLOv3 at 82.3%, YOLOResNet50 at 85.5%, but less than YOLOv4 at 91.6%. However, the detection speed of YOLOMuskmelon at 96.3 frame per second(fps) outperformed YOLOv3 at 56.6fps, YOLOv4 at 54.1fps and YOLOResNet50 at 71.2fps. Meanwhile, the YOLOMuskmelon which is 56.1% faster than YOLOv4 model showed a better generalization and real−time fruit harvesting robots prospect.

INDEX TERMS YOLOMuskmelon, fruit detection, speed and accuracy, complicated environment, harvesting robots.

I. INTRODUCTION

Robotic harvesting offers solution to the expensive cost of manual labor, growing demand for food, increasing fruit quality and so on. With these, there have been growing interest in the application agricultural robots for harvesting fruit and vegetables over the years [1]. Meanwhile, fruit detection plays an important part in harvesting robots, and the detection of fruit for harvest purpose depends on accuracy and speed. However, fruit detection is generally influenced by many factors such as illumination variation, occlusions, including cases when the fruit exhibits a similar visual appearance as its background. Therefore, a robust well-generalized fruit detection model is necessary to overcome these challenges.

Muskmelon as a case study for fruit detection in this paper provides essential nutrients such as vitamin A, vitamin C and several health benefits to man [2]. Meanwhile, muskmelon matures more quickly in moist, warm weather than cool conditions. The appearance, smell, touch, and maturity time frame are the traditional measures used to identified matured muskmelons. The event that happens during maturity stages is that the background color of the fruit turns from green to yellow, the stem breaks (slips) away from the vine easily and the netting gets coarse and rough during maturity stages. These have made muskmelon detection very tasking in addition to its complicated environment conditions mentioned earlier particularly occlusion, which are stems occlusion, leaves occlusion as well as muskmelon overlap. Deep learning methods have made considerable progress to tackle the challenges of fruit detection.

Fruit detection using deep learning have been investigated in numerous studies by Shi *et al.* [3], Li *et al.* [4], Kirk *et al.* [5], Liu *et al.* [6], Lin *et al.* [7], Vasconez *et al.* [8] and so on. Although good performance in their application were achieved, still faces major challenges of tradeoff between detection speed and accuracy i.e. higher accuracy but slower and lower accuracy but faster. Furthermore, Koirala *et al.* [9] summarized the overview of fruit detection. Sa *et al.* [1] reported a better fruit detection results including rock-melon at F_1 of 84.8% experimented on the faster regional-convolutional neural network (Faster R-CNN) [10] detector. Nevertheless, the detection of small fruits is difficult and its speed still requires improvement for real-time harvesting robot. Bargoti and Underwood *et al.* [11] obtained the F_1 of 90% on fruit detection in orchards using Faster R-CNN model, but observed missed fruit detection within the

The associate editor coordinating the review of this manuscript and approving it for publication was Li He.

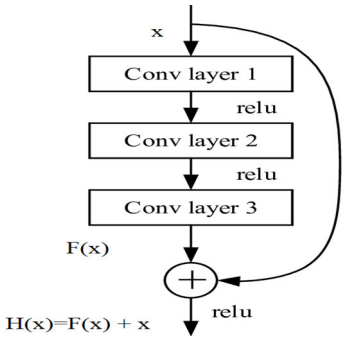

FIGURE 1. ResNet structure [14].

tight clusters of fruits. Zheng *et al.* [12] reported an average precision(AP) of 88.8% at detection speed of 40 frame per second(fps) on muskmelon detection based on YOLOv3 [13] and AP of 87.36% at 13fps based on ResNet50 model [14]. However, the muskmelons were not detected in highly occluded condition, and detection speed need improvement to support harvesting robots. According to Koirala *et al.* [15], a single−stage deep learning detector such as You Only Look Once (YOLO), Single Shot Detector (SSD) [16] is faster than a two−stage detector such as Faster R-CNN with similar accuracy. Therefore, the optimization of a single−stage deep learning architecture for speed and accuracy is a contribution to the development of robust fruit detection, to foster the performance of harvesting robots.

YOLOv3 [13] and YOLOv4 [17] belonging to a single−stage deep learning detector are popular real−time object detectors in computer vision. They directly predict the bounding boxes and their corresponding classes in a single network. In comparison, YOLOv3 applied DarkNet53 backbone with Leaky ReLU [18] activation, feature pyramid network (FPN) [19] as Neck, binary cross-entropy loss for each label and predicts objects in three different scales, while YOLOv4 is composed of CSPDarknet53 backbone with Mish activation, spatial pyramid pooling (SPP) [20], path aggregation network (PANet) as Neck [21] and YOLOv3 Head. In addition, YOLOv3's accuracy and speed were respectively improved by 10% and 12% in YOLOv4. Nevertheless, there are few studies on muskmelon detection using improved YOLOv3 with ResNet framework backbone, and deep learning findings on fruit detection using YOLOv4 are limited.

He *et al.* [14] proposed state-of-the-art ResNet. The skip connection or shortcut connection introduced into Fig. 1 was applied to solve drops off from saturated accuracy for deeper neural network. The 1 × 1 convolution layers are added to the beginning (conv layer1) and end (conv layer 3) of the network, whereby reducing the number of parameters without network performance degradation. ResNet is an excellent object detection compared to VGGNet, SqueezeNet, InceptionV4 and DenseNet according to Zheng *et al.* [12].

This paper proposed a robust YOLOMuskmelon model that is accurate and fast to solve the challenges encountered by fruit detection. The method incorporated ResNet43 backbone [14] with new residual block 2,3,4,3,2 arrangements into modified YOLOv3 [13] for deeper network and rich features extraction. The backbone was activated with rectified linear unit (ReLU) for non-linearity. Meanwhile, SPP, FPN, complete Intersection over Union (CIoU) loss, and distance Intersection over Union−Non Maximum Suppression(DIoU−NMS) [22] were added to the YOLOMuskmelon model to improve detection performance. The exploratory studies on YOLOv3 and YOLOv4 for muskmelon fruit detection was experimented and compared with YOLOMuskmelon. The results showed that YOLOMuskmelon can achieve an impressive detection accuracy and real−time detection speed compared to other state-of-art detection algorithms. The main contributions of this paper is to develop a robust fruit detection algorithm that is accurate and fast to detect muskmelon under different environments, and applicable for harvesting robots.

The remainder of this paper is organized as follows: Section II proposes the muskmelon fruit detection model. Section III describes the tested results and discussion of the proposed method, and Section IV draws conclusions with future work plan.

II. METHODOLOGY

A. DATASET DETAILS

The images used in this paper were captured using digital camera with resolution of 3968 × 2976 pixels and collected from the greenhouse in wanghaizhuang village, Houcheng township, Taigu county, Shanxi Province, China under natural daylight conditions, including leaf occlusion, stem occlusion, illumination variation, and overlap disturbances. Fig. 2 shows some images from the dataset under natural daylight. A total of 410 muskmelon images were taken, stored in JPG format and randomly divided into 80% training set and 20% test set. A graphical image annotation tool called labelImg (https://github.com/tzutalin/labelImg) was applied to hand label all the ground truth bounding boxes, and the annotation files saved in YOLO format to complete dataset construction. The YOLO format annotation contains object class, coordinates, height and width. Meanwhile, the bounding boxes were drawn by the supposed shape depending on what the eyes can see as depicted in Fig. 3 without minding the highly leaves occlusion and other disturbing nature of the images. Fig. 3 displays samples of the annotation principle of the images in the constructed dataset. Finally, the annotated muskmelon images were checked thoroughly to ensure that no unannotated class was missing out. A total of 838 bounding boxes were derived from the 330 images in the training set and 198 bounding boxes from 80 images in the test set.

B. YOLOMUSKMELON MODEL

The overview of the proposed YOLOMuskmelon model is shown in Fig. 4. The 43−layers of ResNet43 is the backbone of YOLOMuskmelon which replaces the DarkNet53 backbone of YOLOv3. The new residual block of ResNet43 backbone was arranged as 2,3,4,3,2 in order to

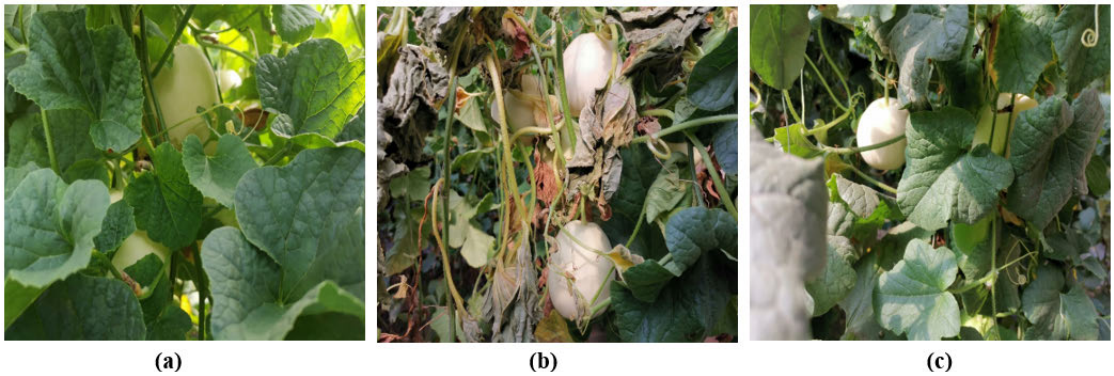

FIGURE 2. Sample of images in datasets (a) leaves occlusion, (b) stems occlusion, and (c) illumination variation.

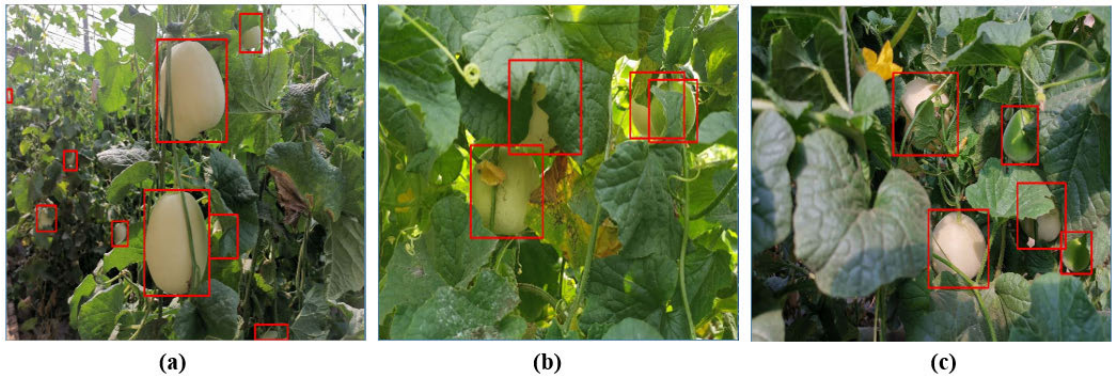

FIGURE 3. Annotation principle of images in dataset (a) small, medium and large targets, (b) targets under occlusion, (c) targets under occlusion and illumination variation.

improve the detection speed and promote deeper network towards detection accuracy without vanishing gradient. The main reason for the integration of ResNet43 backbone into YOLOMuskmelon was to allow training of much deeper networks, solve network degradation problem, avoid overfitting, accelerates the training speed, and promotes faster network convergence. Meanwhile, shortcut connection of ResNet43 is non−linear function which capture patterns in complicated data before passing to the next layer for feature mapping, while DarkNet53' shortcut is linear activated. The designed 1×1 convolution bottleneck within the ResNet43 backbone is used to reduce complexity and number of parameters without performance degradation. ReLU [18] activation function was applied to each layer of the ResNet43, because it is computationally efficient, overcomes vanishing gradient problem, promote fast training and better performance. Generally, activation function plays a vital role in the performance of every deep neural network by introducing non-linearity.

The non-maximum suppression (NMS) is a widely used algorithm to select one entity (i.e. bounding boxes) out of many overlapping entities by removing redundant detections of multiple bounding boxes in order to find the best match. The DIoU−NMS was applied to the YOLOMuskmelon, because it considered overlap area and distance between two central points of bounding boxes according to Zheng *et al.* [22]. The original front detection layers (FDL×3) of YOLOv3 [13] was pruned to FDL×2 in the YOLOMuskmelon for speed detection enhancement. The FPN was adopted as the Neck to get feature pyramids in YOLOMuskmelon, which enables models to generalized very well on object scaling. Furthermore, YOLOMuskmelon used CIoU loss function [22] to address bounding boxes regression loss for a faster convergence and better performance.

The purpose of adding the SPP network [20] shown in Fig. 4 to YOLOMuskmelon is for the optimization of muskmelon fruit detection. SPP network is basically a feature enhancement module, which extracts the main information of the feature map and performs stitching. A situation of extracted feature map being blurred, leading to inaccuracies or loss of detection. The SPP network helps to avoid missed target fruit detection and inaccuracies. SPP extract both the multiscale global and local features of the same detection stage.

C. EXPERIMENT SETUP AND EVALUATION

The models training and testing were implemented on DarkNet platform, computer with the following specifications: Intel-Core i7-8700 CPU @ 64-bit 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 YOLOMuskmelon was compared

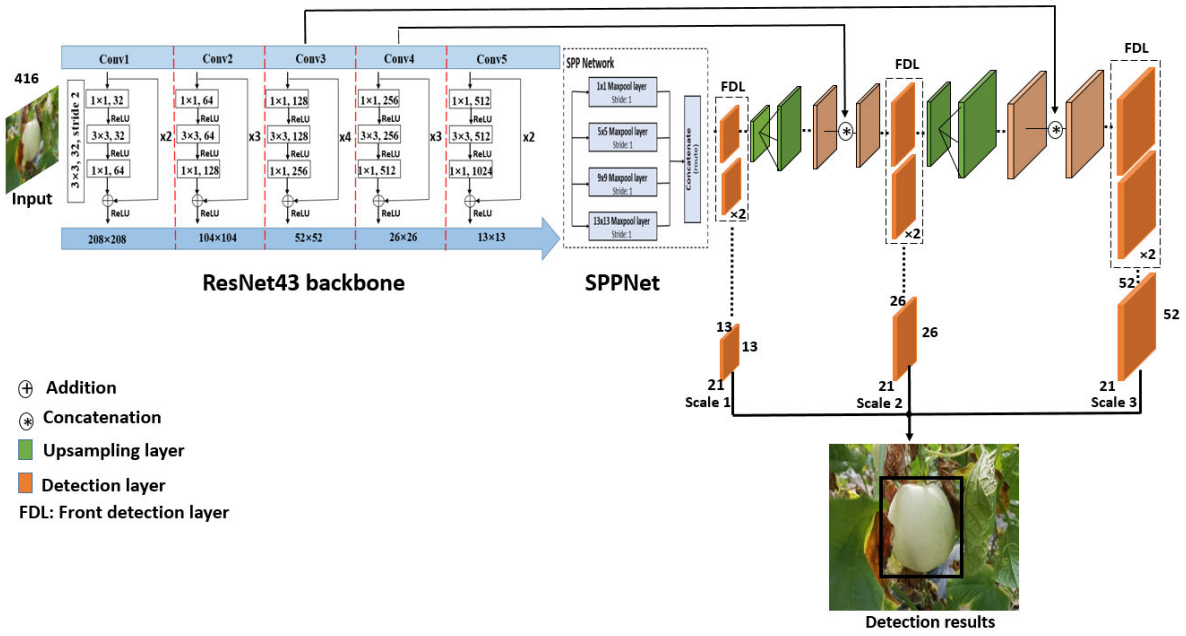

FIGURE 4. Overview of YOLOMuskmelon detection model.

TABLE 1. Details of Trained Model.

Model	Backbone	Activation	SPP	Neck	Loss
YOLOv3[7]	Darknet53	Leaky		FPN	MSE
YOLOv4[11]	CSPDarknet53	Mish	√	PANet	CIoU
YOLOResNet50	ResNet50	ReLU	√	FPN	CIoU
YOLOMuskmelon					
YOLOLeaky	ResNet43	Leaky	√	FPN	CIoU
YOLOMish	ResNet43	Mish	√	FPN	CIoU
YOLOSwish	ResNet43	Swish	√	FPN	CIoU
YOLOReLU	ResNet43	ReLU	√	FPN	CIoU

TABLE 2. Ablation studies' results of YOLOMuskmelon.

Model	IoU(%)	P(%)	R(%)	F_1 (%)	AP (%)	fps
YOLOLeaky	71.8	87.0	87.0	83.0	86.3	96.6
YOLOMish	66.8	83.0	85.0	84.0	88.3	96.3
YOLOSwish	71.4	85.0	82.0	84.0	88.5	96.1
YOLOReLU	70.9	85.0	82.0	84.0	89.6	96.3

with YOLOResNet50, YOLOv3, and YOLOv4. The different modifications of YOLOMuskmelon were represented by model YOLOLeaky, YOLOMish, YOLOSwish, and YOLOReLU, whose ResNet43 backbone were respectively activated with Leaky, Mish, Swish, and ReLU. And all the modified YOLOMuskmelon models incorporated the same SPP network, FPN Neck and CIoU loss function. The models were trained and tested according to details shown in Table 1.

The size of the anchor boxes was counted from the annotated dataset using k-means clustering algorithm before models training. The generated nine anchor boxes were embedded into the models according to the three scales of detection layer (52 × 52, 26 × 26 and 13 × 13 feature). The anchors were assigned to the model configuration files individually

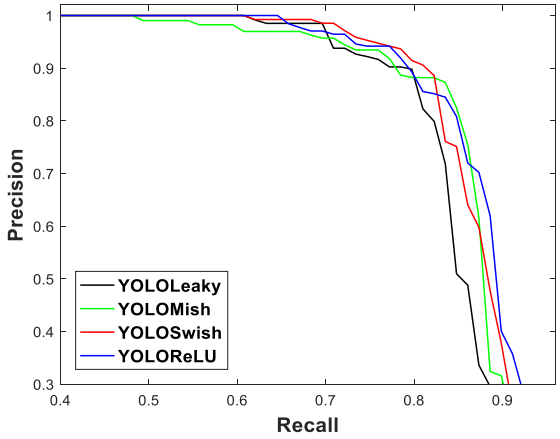

FIGURE 5. P–R curves for different modification of YOLOMuskmelon.

in descending order of dimension, from the first scale to the third scale in order to improve the muskmelon fruit detection model. All the models receive an inputs images of 416 × 416 pixels, learning rate of 0.001 to reduce training loss with iterations between 0 and 4000, and batch and subdivision of 64 and 32 respectively to reduce the memory usage. The momentum and weight decay were set to 0.9 and 0.0005 respectively. Meanwhile, random initialization method was used to initialize the weights for training YOLOMuskmelon and YOLOResNet50, while the official pre-trained weights was used for training YOLOv3 and YOLOv4 model.

The models were evaluated using Precision, Recall, F_1-score and AP. Precision is the ratio of the number of correctly detected muskmelon to the total number of detected muskmelon, Recall is the ratio of the number of correctly detected muskmelon to the total number of muskmelon in the dataset, and F_1-score is simply the trade-off between the Precision and Recall to show the performance of the

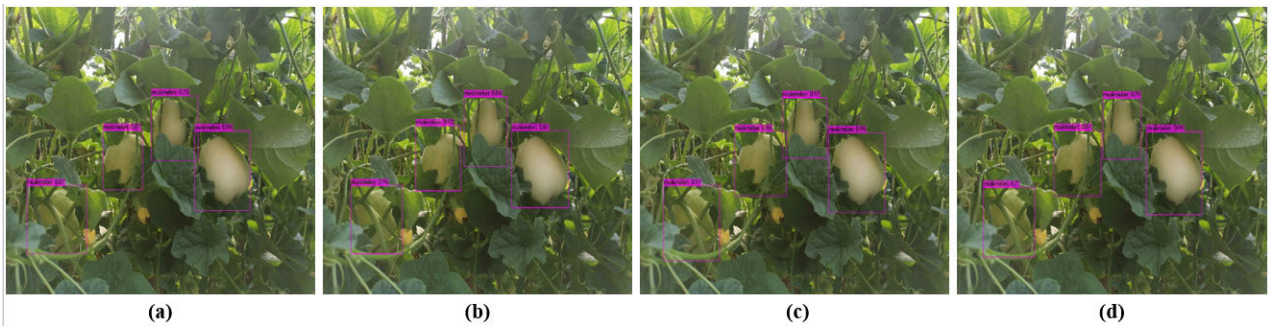

FIGURE 6. Tested image result for different modification of YOLOMuskmelon (a) YOLOLeaky, (b) YOLOMish, (c) YOLOSwish, and (d) YOLOReLU.

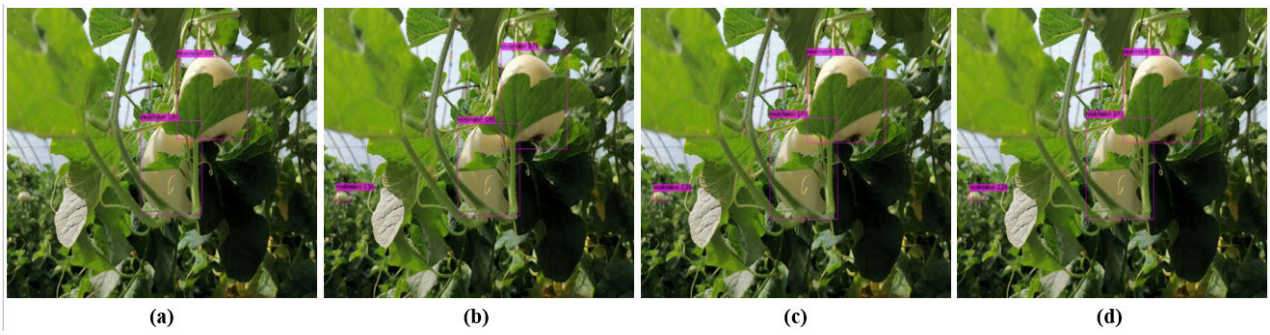

FIGURE 7. SPP network added impact on (a) YOLOv3, (b) YOLOv4, (c) YOLOResNet50, and (d) YOLOMuskmelon.

trained models. The calculation parameters are shown in Eq. (1)- (3).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{F}_1 = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \tag{3}$$

where, TP is the True Positive (correct detections), FN is False Negative (missed detections), and FP is the False Positive (incorrect detections). AP is the Average Precision that describes the overall performance of the model under different confidence thresholds, defined in Eq. (4)

$$\text{AP} = \sum_{\text{n}} (\text{r}_{\text{n}+1} - \text{r}_{\text{n}}) \max_{\tilde{r}:\tilde{r}^3 r_{\text{n}+1}} \text{p}(\tilde{r}) \tag{4}$$

where $p(\tilde{r})$ is the measured precision at recall$\tilde{r}$

III. RESULTS AND DISCUSSION

A. ABLATION STUDIES ON YOLOMUSKMELON

The obtained model weight size of different modification of YOLOMuskmelon is 98.1MB. The ablation study is necessary to determine the effect of different activation functions on YOLOMuskmelon model. From the results of ablation studies shown in Table 2, the F_1 score of YOLOLeaky activated with Leaky is 83%, YOLOMish activated with Mish is 84%, YOLOSwish activated with Swish is 84%, and YOLOReLU activated with ReLU is 84%. With the F_1 score showing no significant difference between the models, the AP of YOLOReLU at 89.6% is greater than YOLOSwish at 88.5%, YOLOMish at 88.3% and YOLOLeaky at 86.3%. The AP is considered more accurate compared to the F_1 score, because it displays the Precision−Recall (P−R) relationship globally. The P−R curves set at 50% IoU threshold for different activations of YOLOMuskmelon is shown in Fig. 5. Meanwhile, P−R curve with a better performance is expected to have a greater area under curve (AUC). YOLOReLU model shows a remarkable performance than other models. The level of AP performance among the models is measure as YOLOReLU>YOLOSwish>YOLOMish>YOLOLeaky. This is an indication that ReLU activation function in YOLOReLU model is best compatible with the ResNet43 backbone, which requires further investigation. The visualization of the image tested under different modification of YOLOMuskmelon model is shown in Fig. 6 for justification. The models were able to detect the same number of muskmelon target in the image, except for some little variation observed with their confidence percentages. Nevertheless, YOLOMuskmelon showed robustness after being tested on different environments.

The obtained detection speed in fps tested on different modification of YOLOMuskmelon model shows no significant difference, as the speed of YOLOLeaky is 96.6fps, YOLOMish is 96.3fps, YOLOSwish is 96.1fps and YOLOReLU is 96.3fps. The obtained detection speed results are due to no changes in their models' weight size. Having achieved an average IoU greater than 50% on all the tested models shows that their detection performance is good, but model YOLOReLU stands selected to represent the YOLOMuskmelon due to its AP result.

B. YOLOMUSKMELON AGAINST OTHER MODELS

Fig. 7 shows that the undetected target muskmelon in Fig. 7(a) was found detected in Fig. 7(b), Fig. 7(c), and Fig. 7(d). This is as a result of SPP network added to YOLOv4, YOLOResNet50 and YOLOMuskmelon,

TABLE 3. Compared performance difference between models.

Model	IoU(%)	P(%)	R(%)	F_1(%)	AP (%)	fps
YOLOv3[13]	73.4	90.0	74.0	82.0	82.3	56.6
YOLOv4[17]	63.7	85.0	85.0	85.0	91.6	54.1
YOLOResNet50	74.5	92.0	76.0	83.0	85.5	71.2
YOLOMuskmelon	70.9	85.0	82.0	84.0	89.6	96.3

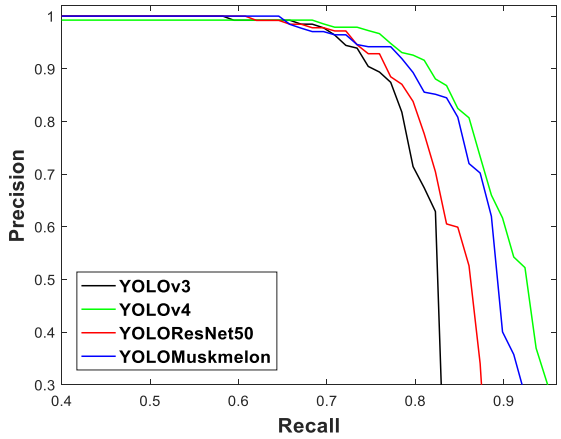

FIGURE 8. P−R curves difference between models.

unlike YOLOv3. SPP enhancement support separate out the most important features from the model backbone [14]. The compared findings shown in Table 3 between models indicated that YOLOMuskmelon hold the least weight size of 98.1MB compared to YOLOv3 at 234MB, YOLOv4 at 244MB, YOLOResNet50 at 158MB. For this reason, YOLOMuskmelon detection speed of 96.3fps is faster than YOLOv3 at 56.6fps, YOLOv4 at 54.1fps and YOLOResNet50 at 71.2fps. This is an excellent achievement. The obtained F_1 score in Table 3 shows 1% difference between the models as the measured value of YOLOv4>YOLOMuskmelon>YOLOResNet50>YOLOv3.

Furthermore, the tested AP in Table 3 supported by P−R curves shown in Fig. 8 indicated that YOLOv4 at 91.6% is greater than YOLOMuskmelon at 89.6%, YOLOResNet50 at 85.5%, and YOLOv3 at 82.3%. With these results, YOLOMuskmelon outperformed YOLOv3 and YOLOResNet50, but has AP of 2% less than YOLOv4 model. This is a case of tradeoff between detection accuracy and speed. However, YOLOMuskmelon detection speed is 56.1% faster than YOLOv4 through the calculated percentage difference. The application of higher grade GPU or reduced image resolution to YOLOMuskmelon would further improve the detection speed.

In comparison against other reported work, our YOLOMuskmelon detection model outperformed YOLOv3−DarkNet53 at AP of 88.93%, detection speed of 40fps and RetNet−ResNet50 at AP of 87.36%, detection speed of 13fps [12], and rock−melon detection [1], in spite of the model being tested under complicated environment. Therefore, YOLOMuskmelon detection model could better generalize and perform excellently well for real−time detection, which is applicable for harvesting or picking robots.

C. FEATURE MAP VISUALIZATION

Understanding the mechanism of the deep neural network clearly tends to be very difficult. Nevertheless, some features visual clues captured were presented in this section. To study the effectiveness of YOLOMuskmelon detection model, some of the feature maps obtained from different last convolutional layers of 208 × 208, 104 × 104, 52 × 52, 26 × 26 and 13 × 13 are shown in Fig. 9(a)–(e) randomly, and compared with YOLOResNet50 model. These layers also correspond to different detection scales provided if all is considered. The FPN embedded YOLOMuskmelon model received the three scales detections from 52 × 52, 26 × 26 and 13 × 13 feature vector. The first feature map shows that only the regions corresponding to the headmost muskmelon was activated. As the network goes deeper from one activated feature map to another, the previous unseen regions are present in the following activated feature map. Combining the

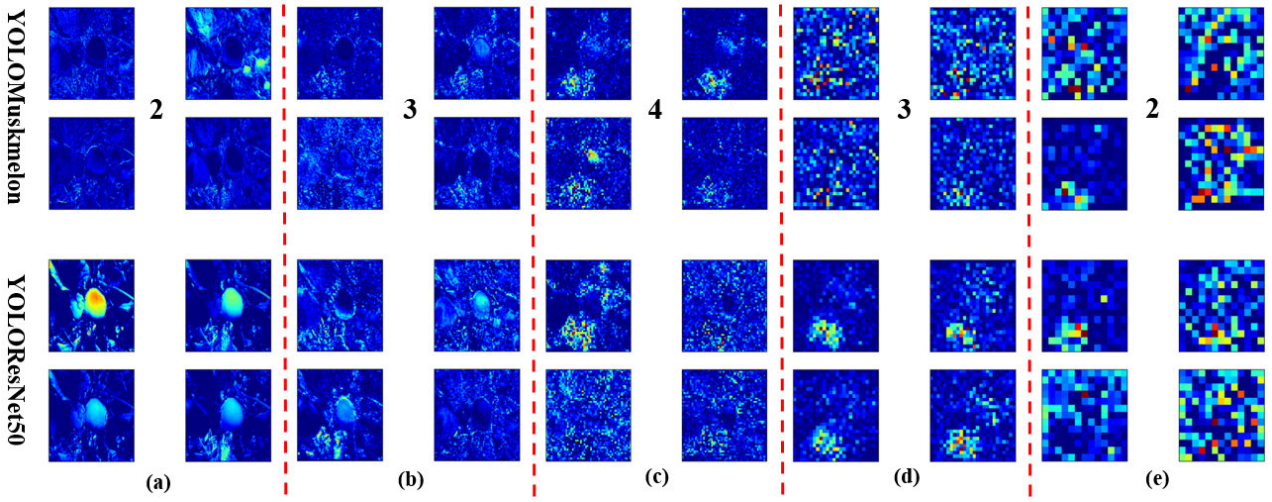

FIGURE 9. Feature map of convolution layers at scale (a) 208 × 208, (b) 104 × 104, (c) 52 × 52, (d) 26 × 26, (e) 13 × 13.

results from different scales as used in this paper, the targeted muskmelons were detected by the model. Fig. 9 shows that the YOLOMuskmelon model's filters learned some information of different directions, which is responsible for its outstanding performance. It can better be explain using the presented results in Table 2, Table 3, Fig. 5 and Fig. 8.

IV. CONCLUSION AND FUTURE WORK

The accurate and speed detection of muskmelon fruit is of great significance to the harvesting and picking robots. A robust YOLOMuskmelon model was proposed in this paper to solve fruit detection challenges. The method incorporated 2,3,4,3,2 block arrangements of ResNet43 backbone with ReLU activation for deeper network, in addition to spatial pyramid pooling(SPP) network for detection accuracies, feature pyramid network(FPN) for feature pyramids extraction, distance Intersection over Union−Non Maximum Suppression(DIoU−NMS) for detection efficiency and accuracy, and complete Intersection over Union (CIoU) loss for better performance and faster convergence. The ablation studies on different modification of YOLOMuskmelon showed that ReLU activated function performed better than Leaky, Mish and Swish with respect to average precision (AP). This is an indication that ReLU activation is better compatible with ResNet framework, call for future studies. The compared AP findings showed that YOLOMuskmelon at 89.6% is greater than YOLOv3 at 82.3%, YOLOResNet50 at 85.5%, but not with YOLOv4 at 91.6%. However, the detection speed of YOLOMuskmelon at 96.3 frame per second(fps) outperformed YOLOv3 at 56.6fps, YOLOv4 at 54.1fps and YOLOResNet50 at 71.2fps. From all indications, our YOLOMuskmelon is 56.1% faster than YOLOv4, which is highly prospective for better generalization and real−time fruit detection. This is applicable for the picking and harvesting robots. Furthermore, our YOLOMuskmelon model can be used for other fruits detection, as it also outperformed other existing state of the art model. Nevertheless, further studies are necessary to improve the accuracy performance of our YOLOMuskmelon model.

In the future, backbones such as MobileNet, DenseNet EfficientNet and so on would be experimented to study reasons why the applied ReLU activation function performed better than other activation functions in our YOLOMuskmelon, and to also improve the fruit detection performance.

REFERENCES

[1] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, "DeepFruits: A fruit detection system using deep neural networks," *Sensors*, vol. 16, no. 8, p. 1222, Aug. 2016.

[2] H. Thakur, S. Sharma, and M. Thakur, "Recent trends in muskmelon (*CucumismeloL.*) research: An overview," *J. Horticultural Sci. Biotechnol.*, vol. 94, no. 4, pp. 533–547, Jan. 2019.

[3] R. Shi, T. Li, and Y. Yamaguchi, "An attribution-based pruning method for real-time mango detection with YOLO network," *Comput. Electron. Agricult.*, vol. 169, Feb. 2020, Art. no. 105214.

[4] J. Li, Y. Tang, X. Zou, G. Lin, and H. Wang, "Detection of fruit-bearing branches and localization of litchi clusters for vision-based harvesting robots," *IEEE Access*, vol. 8, pp. 117746–117758, Jul. 2020.

[5] R. Kirk, G. Cielniak, and M. Mangan, "L*a*b*fruits: A rapid and robust outdoor fruit detection system combining bio-inspired features with one-stage deep learning networks," *Sensors*, vol. 20, no. 1, p. 275, Jan. 2020.

[6] Z. Liu, J. Wu, L. Fu, Y. Majeed, Y. Feng, R. Li, and Y. Cui, "Improved kiwifruit detection using pre-trained VGG16 with RGB and NIR information fusion," *IEEE Access*, vol. 8, pp. 2327–2336, 2020.

[7] G. Lin, Y. Tang, X. Zou, J. Xiong, and J. Li, "Guava detection and pose estimation using a low-cost RGB-D sensor in the field," *Sensors*, vol. 19, no. 2, p. 428, Jan. 2019.

[8] J. P. Vasconez, J. Delpiano, S. Vougioukas, and F. A. Cheein, "Comparison of convolutional neural networks in fruit detection and counting: A comprehensive evaluation," *Comput. Electron. Agricult.*, vol. 173, Jun. 2020, Art. no. 105348.

[9] A. Koirala, K. B. Walsh, Z. Wang, and C. McCarthy, "Deep learning–method overview and review of use for fruit detection and yield estimation," *Comput. Electron. Agricult.*, vol. 162, pp. 219–234, Jul. 2019.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2015, pp. 91–99.

[11] S. Bargoti and J. Underwood, "Deep fruit detection in orchards," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Singapore, May 2017, pp. 3626–3633.

[12] Y.-Y. Zheng, J.-L. Kong, X.-B. Jin, X.-Y. Wang, and M. Zuo, "CropDeep: The crop vision dataset for deep-learning-based classification and detection in precision agriculture," *Sensors*, vol. 19, no. 5, p. 1058, Mar. 2019.

[13] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[15] A. Koirala, K. B. Walsh, Z. Wang, and C. McCarthy, "Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of 'MangoYOLO,"' *Precis. Agricult.*, vol. 20, no. 6, pp. 1107–1135, Feb. 2019.

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multi box detector," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[17] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*. [Online]. Available: http://arxiv.org/abs/2004.10934

[18] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2013, p. 1.

[19] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jul. 2017, pp. 2117–2125.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[21] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 8759–8768.

[22] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," 2019, *arXiv:1911.08287*. [Online]. Available: http://arxiv.org/abs/1911.08287

OLAREWAJU M. LAWAL (Member, IEEE) received M.Eng. degree in agricultural and environmental engineering (crop processing and storage option) from the Federal University of Technology, Akure, in 2015, and the Ph.D. degree in nuclear science and technology from Xi'an Jiaotong University, Shaanxi, China, in 2019.

From 2001 to 2017, he has acquired experiences, taught in various schools, and worked at different research institutions. He is currently a Lecturer with the College of Agricultural Engineering, Shanxi Agricultural University, China. His research interests include machine learning, artificial intelligence, robotics, food irradiation, crop storage and processing, agro-processing machine design and fabrication, and development of processing and storage equipment.