YOLOFig detection model development using deep learning

Olarewaju Mubashiru Lawal | Huamin Zhao

College of Agricultural Engineering, Shanxi Agricultural University, Jinzhong, China

Correspondence
Olarewaju Mubashiru Lawal, College of Agricultural Engineering, Shanxi Agricultural University, Jinzhong City, 030801, China.
Email: olarewajulawal@yahoo.com

Funding information
Natural Science Foundation of Shanxi Province, China, Grant/Award Number: 2020BQ34

Abstract
The detection of fruit, including accuracy and speed is of great significance for robotic harvesting. Nevertheless, attributes such as illumination variation, occlusion, and so on have made fruit detection a challenging task. A robust YOLOFig detection model was proposed to solve detection challenges and to improve detection accuracy and speed. The YOLOFig detection model incorporated Leaky activated ResNet43 backbone with a new 2,3,4,3,2 residual block arrangement, spatial pyramid pooling network (SPPNet), feature pyramid network (FPN), complete (CIoU) loss, and distance DIoU–NMS to improve the fruit detection performance. The obtained average precision (AP) and speed (frames per second or fps) respectively under 2,3,4,3,2 residual block arranged backbone for YOLOv3b is 78.6% and 69.8 fps, YOLOv4b is 87.6% and 57.1 fps, and YOLOFig is 89.3% and 96.8 fps; under 1,2,8,8,4 residual block arranged backbone for YOLOv3 is 77.1% and 56.3 fps, YOLOv4 is 87.1% and 52.5 fps, and YOLOResNet70 is 87.3% and 79 fps; and under 3,4,6,3 residual block arranged backbone for YOLOResNet50 is 85.4% and 77.1 fps. An indication that the new residual block arranged backbone of 2,3,4,3,2 outperformed 1,2,8,8,4 on an average AP of 1.33% and detection speed of 15.2%. Finally, the compared results showed that the YOLOFig detection model performed better than other models at the same level of residual block arrangement. It can better generalize and is highly suitable for real-time harvesting robots.

1 | INTRODUCTION

In the past few years, the popularity of figs has increased due to their sweetness and high nutritional value. Fig fruit and its leaves according to Lianju et al. [1], contain plenty of amino acids and inorganic elements, medical components such as flavone, rutin and quercetin, which implies that fig has a significant role in the human body health. Meanwhile, fig fruits are transformed into several processing products, such as dried figs, preserved fruits, jam, juice, wine, powder, and so on. For these reasons, over one million tonnes of figs are produced yearly in the world, according to the Food and Agriculture Organization of the United Nations [2], where Turkey is the largest fig producer with 305,450 tonnes, followed by Egypt with 167,622 tonnes, and China ranked 14 with 14,324 tonnes of production volume per year. Nevertheless, figs harvesting mainly relies on manual labour, and the labour cost accounts for a high percentage of the total production cost. Therefore, it is necessary to replace the manual labour with harvesting robots, because robotic harvesting offers a solution to labour cost reduction, enabling selective harvesting, optimizing harvest scheduling and maximizing operational efficiency and profits [3]. The potential applications of vision-based fruit picking/harvesting robots to agriculture have been reviewed, summarized, and future perspective reported by Tang et al. [4]. Robotic fruit harvesting is an exciting technology with challenges, because it requires the integration of multiple subsystems such as fruit detection, motion, and guided manipulation. Fruit detection is the most important and challenging of these multiple subsystems. Uneven illumination in an unstructured environment, fruit occlusion by stem and leaves, overlapping fruit and other unpredictable factors are problems faced by fruit detection, which in other word makes it extremely difficult to develop a vision system as intelligent as humans for detection. In addition, the different colour types, similar visual appearance as background, e.g. leaves, and ripeness variation from green to dark brown are contributory factors to fig fruit detection.
Therefore, a fig fruit detection model that is environmentally adaptable to the harvesting robots is crucial to overcome these challenges.

Deep learning techniques have made considerable efforts in addressing the challenges in fruit detection. Koirala et al. [5] summarized fruit detection and yield estimation in deep learning. Presently, two-stage object detectors such as faster region-based network (Faster R-CNN) [6] and single-stage object detectors, e.g., You Only Look Once (YOLO), single shot detector (SSD) [7] are mainly the divided stages of deep learning detection models. Sa et al. [8] explored multi-modal (RGB and NIR) information of deepfruits based on Faster R-CNN to achieve an improved F1 score from 80.7% to 83.8% for the fruit detection. However, the detection of small fruits is difficult with a slower detection speed for real-time in-field operation of harvesting robots. Missing fruits detection was noted in the tight clusters of fruit tested on the model proposed by Bargoti et al. [9] in spite of the F1 score of more than 90% reported. The proposed mangoYOLO by Koirala et al. [10] through modified YOLOv3 obtained an accuracy of 98.3% on detected mango, but with a slower inference time of 15 ms. Zheng et al. [11] reported 92% detection accuracy on different crops and suggested that YOLOv3 [12] framework has a good potential application for agricultural detection tasks. An indication that a single-stage is faster than a two-stage fruit detector. Lawal [13] modified the YOLOv3 framework using the DenseNet backbone for tomato detection, and achieved an accuracy of 99.3% but with a slower detection time of 44 ms. Due to the significant role of both detection accuracy and speed in agricultural harvesting robots, it is necessary to develop a robust detection algorithm for their tasks.

YOLOv3 [12] and YOLOv4 [14], being a single-stage deep learning detector is popular for real-time object detector in computer vision. It directly predicts the bounding boxes and corresponding classes in a single-stage network at three different scales. YOLOv3 and YOLOv4 respectively applied the DarkNet53 backbone activated with Leaky rectified linear unit (ReLU) [15] and the CSPDarkNet53 backbone activated with Mish [16]. Meanwhile, the backbone of YOLOv3 and YOLOv4 which is composed of 1,2,8,8,4 residual block arrangement is long overdue for improvement. Furthermore, YOLOv3 uses a feature pyramid network (FPN) [17] as a model Neck, binary cross-entropy loss, while YOLOv4 applies spatial pyramid pooling network (SPPNet) [18], path aggregation network (PANet) [19] as a model Neck and YOLOv3 head. YOLOv4 is built on YOLOv3 with 10% detection accuracy and 12% detection speed improvement as reported. Recently, Lawal [20] reported a mix of DarkNet and DenseNet as a backbone with FPN to improve the YOLOv4 algorithm, achieving an accuracy of 98.4% greater than YOLOv4 at 97.6% tested on tomatoes in a natural scene. Nevertheless, fig fruit detection studies using deep learning single-stage detector is limited and exploratory findings on fruit detection using YOLOv4 are rarely available. In addition, there are a few studies on fig fruit detection using YOLOv4 integrated with the ResNet architecture backbone.

The state-of-the-art building block of ResNet shown in Figure 1 was proposed by He et al. [21]. This residual block or identity block is when the activation from the previous layer is fast-forwarded to a deeper layer in the neural network. The skip connection (shortcut connection) introduced into ResNet was explored to solve drops off from saturated accuracy for a deeper neural network. The 1 × 1 convolution layers are applied at the beginning (conv layer 1) and end (conv layer 3), while 3 × 3 convolution layer is placed in the middle (conv layer 2) of the network in order to reduce the number of parameters without network performance degradation. According to Zheng et al. [11], ResNet is an excellent object detection compared to Inceptionv4, VGGNet, SqueezeNet, and DenseNet. In the quest for an improved single-stage fruit detector, Lawal [22] incorporated the ResNet43 backbone into the YOLOv4 algorithm for muskmelon fruit detection, and reported an accuracy of 89.6% and speed of 96.3 frames per second (fps). However, the proposed method was not experimented on other types of fruit to ascertain the obtained findings.

This paper proposed a robust YOLOFig detection model to solve the shortcomings encountered by fruit detection and to improve fruit detection accuracy and speed. The model incorporated residual block 2,3,4,3,2 arrangements of the ResNet43 backbone [21] into YOLOv4 [14] for a deeper network and rich features extraction. The ResNet43 backbone was Leaky activated for non-linearity, including SPPNet, FPN, complete (CIoU) loss, and distance (DIoU–NMS) [23] added to the model for improved detection performance. Meanwhile, exploratory studies on YOLOv3 and YOLOv4 was experimented to compare the residual block arrangement 1,2,8,8,4 and 2,3,4,3,2 of their backbones. The obtained findings showed that the YOLOFig detection model can achieve an impressive detection accuracy and real-time detection speed.

The remainder of this paper is arranged as follows: Section 2 proposes the fig detection model. Section 3 explains the results and discussion of the proposed model, and Section 4 draws conclusions and future work.

2 | METHODOLOGY

2.1 | Dataset details

The images of figs were taken from a greenhouse in Taigu county, Jinzhong city, Shanxi, China under natural daylight

![Figure 1](image-url)
The images were captured using a digital camera with an image resolution of 2976 × 3968 pixels, stored in JPG format. Apart from 30 test images set aside, a total of 382 fig images were collected and randomly divided into 80% train set and 20% valid set. For the dataset construction, labelling image annotation tool was used to hand label all the ground truth bounding boxes, in which the annotation files were saved in YOLO format. Without minding the diverse conditions of the images such as occlusion, illumination variation, similar background and so on, the bounding boxes were drawn by the supposed shape depending on what human eyes can see as displayed in Figure 2, and annotated fig images were thoroughly checked to ensure no unannotated class. A total of 3513 bounding boxes were derived from the 306 images in the train set and 748 bounding boxes from 76 images in the valid set.

2.2 YOLOFig model development

The proposed YOLOFig detection model is made up of ResNet43 backbone with 2,3,4,3,2 new residual block arrangement as shown in Figure 3. The ResNet43 replaces the DarkNet53 backbone with block arrangement 1,2,8,8,4 of YOLOv3. The reason for ResNet43 architecture usage was to enable deeper network, promote faster network convergence and solve the problem of network degradation. Meanwhile, each layer of
ResNet43 backbone was activated with Leaky [15], because it fixes the dying ReLU and speeds up the training process. The SPPNet [18] as part of the YOLOFig model was added after the ResNet43 backbone for fruit detection optimization. It is a feature enhancement network that extracts both the multi-scale global and local features of the same detection stage. The applied SPPNet helps to avoid missed and inaccurate detection of an extracted blurred feature map. The three scales of YOLOFig are feed through 13 × 13, 26 × 26, and 52 × 52 feature vector as depicted in Figure 3.

The non-maximum suppression (NMS) algorithm used in the YOLOFig model to select a bounding box out of many overlapping entities by removing redundant detections of multiple bounding boxes in order to find the best match is DIoU−NMS. According to Zheng et al. [11], DIoU−NMS considered overlap area and the distance between two central points of bounding boxes. The original six detection layers of the YOLOv3 head were pruned to four layers in the YOLOFig detection model for detection speed improvement with FPN [17] application as the Neck to get feature pyramids. The FPN enables a well-generalized model on object scaling. Finally, the YOLOFig model applied complete IoU (Ciou) loss function [23] for faster convergence and better performance.

2.3 | Experiment and evaluation

The training and testing of models were implemented on DarkNet platform, computer with Intel 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 YOLOFig model was compared with YOLOResNet50, YOLOResNet70, YOLOv3, and YOLOv4 at different residual block arrangements. All models were trained and tested according to the stated details in Table 1. With reference to Figure 3, the 2,3,4,3,2 residual block corresponds to ResNet43, Darknet35, and CSPDarknet35, while the 1,2,8,8,4 residual block is to ResNet70, Darknet53, and CSPDarknet53.

### Table 1: Information of all trained models

| Models        | Block            | Backbone       | Activation | SPPNet | Neck   | Loss   |
|---------------|------------------|----------------|------------|--------|--------|--------|
| YOLOv3b       | 2,3,4,3,2        | Darknet35      | Leaky      | FPN    | Ciou   | MSE    |
| YOLOv3        | 1,2,8,8,4        | Darknet53      | Leaky      | FPN    | Ciou   |        |
| YOLOv4b       | 2,3,4,3,2        | CSPDarknet35   | Mish       | PANet  | Ciou   |        |
| YOLOv4        | 1,2,8,8,4        | CSPDarknet53   | Mish       | PANet  | Ciou   |        |
| YOLOResNet50  | 3,4,6,3          | ResNet50       | Leaky      | FPN    | Ciou   |        |
| YOLOResNet70  | 1,2,8,8,4        | ResNet70       | Leaky      | FPN    | Ciou   |        |
| YOLOFigA      | 2,3,4,3,2        | ResNet43       | Leaky      | FPN    | Ciou   |        |
| YOLOFigB      | 2,3,4,3,2        | ResNet43       | Leaky      | FPN    | Ciou   |        |
| YOLOFigC      | 2,3,4,3,2        | ResNet43       | Leaky      | PANet  | Ciou   |        |
| YOLOFigD      | 2,3,4,3,2        | ResNet43       | ReLU       | FPN    | Ciou   |        |
| YOLOFigE      | 2,3,4,3,2        | ResNet43       | Swish      | FPN    | Ciou   |        |
| YOLOFigF      | 2,3,4,3,2        | ResNet43       | Mish       | FPN    | Ciou   |        |

From the annotated dataset, the k-clustering algorithm was used to count the size of the anchor boxes before model training. The generated nine anchor boxes were incorporated into the models individually in descending order according to the three scales of the detection layer, i.e. 52 × 52, 26 × 26 and 13 × 13 feature. This is to improve the fig fruit detection model. The models received an input image of 416 × 416 pixels, 0.001 learning rate to reduce the training loss, iterations between 0 and 4000, 64 batches and 32 subdivisions to reduce the memory usage, the momentum of 0.9, and weight decay of 0.0005. Meanwhile, a random initialization method was applied to initialize the weights for the models’ training including YOLOv3 and YOLOv4 models. This is to maintain the models’ training consistency.

The trained models were evaluated using Precision, Recall, F1 score and AP. Precision is the ratio of correctly detected fig to the total number of detected fig. Recall is the ratio of the correctly detected fig to the total number of fig in the dataset, and F1 score is simply the trade-off between the Precision and Recall. Equations (1)–(3) describe the parameters.

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

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

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

where TP is the true positive (correct fig detection), FN is false negative (missed fig detection), and FP is the false positive (incorrect fig detection). AP is the average precision that explains the overall performance of the model under various confidence thresholds, expressed as Equation (4)

\[
AP = \sum_n (r_n - r_0) \max_{\overline{r} \in \overline{r}_{n+1}} \rho(\overline{r}) \quad (4)
\]

where \(\rho(\overline{r})\) is the measured precision at recall \(\overline{r}\).

3 | RESULTS AND DISCUSSION

3.1 | Ablation studies on YOLOFig model

The ablation studies presented in Table 2 is to determine the impact of SPPNet, different activation functions, and model Neck on YOLOFig model towards achieving an excellent performance. The YOLOFigA model without SPPNet has a weight size of 92.1 MB, but was increased to 98.1 MB in YOLOFigB, YOLOFigD, YOLOFigE and YOLOFigF with SPPNet addition. The weight size was further increased to 136 MB in YOLOFigC model, due to the incorporation of PANet as a model Neck. This indicated that PANet constituted an increase in model weight size with less performance.
**TABLE 2** Ablation studies on YOLOFig detection model

| Model        | Size (MB) | IoU (%) | P (%) | R (%) | F1 (%) | AP (%) | fps  |
|--------------|-----------|---------|-------|-------|--------|--------|------|
| YOLOFigA     | 92.1      | 58.5    | 73.0  | 89.0  | 80.0   | 89.2   | 97.9 |
| YOLOFigB     | 98.1      | 58.6    | 74.0  | 88.0  | 80.0   | 89.3   | 96.8 |
| YOLOFigC     | 136.0     | 59.1    | 75.0  | 89.0  | 81.0   | 88.6   | 79.4 |
| YOLOFigD     | 98.1      | 60.5    | 76.0  | 86.0  | 81.0   | 88.8   | 94.7 |
| YOLOFigE     | 98.1      | 57.3    | 73.0  | 87.0  | 80.0   | 88.2   | 90.2 |
| YOLOFigF     | 98.1      | 60.1    | 76.0  | 88.0  | 81.0   | 89.1   | 91.5 |

**FIGURE 4** P−R curves for different modifications of YOLOFig detection model

compared to FPN. However, the IoU of the different modifications of YOLOFig are considered as good predictions because they are greater than 50%. However, their performances differ as shown in Table 2. The F1 score as a trade-off between Precision and Recall showed little significant difference among the modifications, where YOLOFigA is 89.0%, YOLOFigB is 80.0%, YOLOFigC is 81.0%, YOLOFigD is 81.0%, YOLOFigE is 80.0% and YOLOFigF is 81.0%. AP being the global relationship between Precision−Recall is considered to be more accurate than F1 score. The higher the value of AP, the better the model performs. The obtained YOLOFig detection speed in fps is a function of the model weight size. The detection speed of model YOLOFigA is 97.9 fps, YOLOFigB is 96.8 fps, YOLOFigC is 79.4 fps, YOLOFigD is 94.7 fps, YOLOFigE is 90.2 fps, and YOLOFigF is 91.5 fps. YOLOFigA with the least weight size is faster than YOLOFigB, YOLOFigC, YOLOFigD, YOLOFigE, and YOLOFigF, but the difference between YOLOFigA and YOLOFigB is very little. Therefore, YOLOFigB with also an outstanding AP results stands selected as YOLOFig detection model.

### 3.2 Residual block arrangement performance

The compared backbone residual block of models stated in Table 3 shows that the YOLOv3b, YOLOv4b, and YOLOFig with 2,3,4,3,2 has a smaller weight size than YOLOv3, YOLOv4, and YOLOResNet70 with 1,2,8,8,4 and YOLOResNet50 with 3,4,6,3. The side by side comparison based on detection accuracy and speed respectively shows that YOLOv3b with 78.6% and 69.8 fps is greater than YOLOv3 with 77.1% and 56.3 fps, YOLOv4b with 87.6% and 57.1 fps is greater than YOLOv4 with 87.1% and 52.5 fps, and YOLOFig with 89.3% and 96.8 fps is greater than both YOLOResNet50 with 85.4% and 79.1 fps and YOLOResNet70 with 87.3% and 79 fps. The calculated average AP and fps of residual block 2,3,4,3,2 arrangements is approximately 1.33% and 15.2% respectively, which is more than 1,2,8,8,4 block arranged models. This indicated that the residual block arrangement of 2,3,4,3,2 is better than 1,2,8,8,4. The tested image to differentiate between block 2,3,4,3,2 and 1,2,8,8,4 is shown in Figure 7 for justification. Just as Figure 5, the undetected figs in Figures 7(a) and (b) are detected in Figure 7(c)−(f) to show the importance of SPPNet incorporation into the models.

**TABLE 3** Performance of different backbone residual block arrangement in models

| Model                | Size (MB) | IoU (%) | P (%) | R (%) | F1 (%) | AP (%) | fps  |
|----------------------|-----------|---------|-------|-------|--------|--------|------|
| YOLOv3b              | 165.0     | 55.3    | 75.0  | 78.0  | 76.0   | 78.6   | 69.8 |
| YOLOv3               | 234.0     | 64.0    | 83.0  | 70.0  | 76.0   | 77.1   | 56.3 |
| YOLOv4b              | 209.0     | 60.1    | 77.0  | 85.0  | 80.0   | 87.6   | 57.1 |
| YOLOv4               | 244.0     | 58.6    | 75.0  | 86.0  | 80.0   | 87.1   | 52.5 |
| YOLOResNet50         | 146.0     | 58.8    | 75.0  | 83.0  | 79.0   | 85.4   | 77.1 |
| YOLOResNet70         | 143.0     | 56.9    | 73.0  | 87.0  | 79.0   | 87.3   | 79.0 |
| YOLOFig              | 98.1      | 58.6    | 74.0  | 88.0  | 80.0   | 89.3   | 96.8 |

YOLOFigD, YOLOFigC and YOLOFigE. The results of the image tested under different modifications of YOLOFig is shown in Figure 5, where the models detected the number of figs in the image. Figure 5 shows that the undetected fig image in Figure 5(a) is found detected in Figure 5(b)−(f). The added SPPNet enhancement support is responsible for this effect [18]. Nevertheless, the different modifications of the YOLOFig model are robust under different conditions.

The obtained YOLOFig detection speed in fps is a function of the model weight size. The detection speed of model YOLOFigA is 97.9 fps, YOLOFigB is 96.8 fps, YOLOFigC is 79.4 fps, YOLOFigD is 94.7 fps, YOLOFigE is 90.2 fps, and YOLOFigF is 91.5 fps. YOLOFigA with the least weight size is faster than YOLOFigB, YOLOFigC, YOLOFigD, YOLOFigE, and YOLOFigF, but the difference between YOLOFigA and YOLOFigB is very little. Therefore, YOLOFigB with also an outstanding AP results stands selected as YOLOFig detection model.
Furthermore, the tested AP of YOLOFig detection model in Table 3 supported by P–R curves shown in Figure 6 outperformed other models including its detection speed. Meanwhile, the application of higher grade GPU or reduced image resolution to the YOLOFig model would further improve the detection speed. With this, the YOLOFig detection model could better generalize and perform excellently well for real-time fruit detection, which is applicable for agricultural harvesting robots.

3.3 Feature map visualization

The feature map of different models is presented in Figure 8 to study the mechanisms responsible for the fig fruit detection improvement. However, deep neural network mechanism is very difficult to understand. The obtained feature maps were randomly taken from different activated last convolutional layers of $208 \times 208$, $104 \times 104$, $52 \times 52$, $26 \times 26$ and $13 \times 13$ were compared as shown in Figure 8. According to different detection scales with reference to the residual block arrangement, the network goes deeper from one activated feature map to another, whereby the previously unseen regions are present in the next activated feature map. The pattern of features mapped within each network, i.e. DarkNet in Figure 8(a), CSPDarkNet in Figure 8(b) and ResNet in Figure 8(c) framework is different. The three scale of detection layers $52 \times 52$, $26 \times 26$ and $13 \times 13$ used in this paper output the end results. The residual block comparison within these detection layers in Figure 8 shows approximately similar feature map. However, the output detection
results in Table 3 and Figure 6 can better explain the outstanding performance of residual block arrangement 2,3,4,3,2 compared to 1,2,8,8,4 and 3,4,6,3. Though deeper networks have the ability to handle more complex data, the performance impediment in this paper is likely associated with vanishing gradient and overfitting during training as the network goes deeper, which might lead to degraded accuracy and generalization errors. For this reason, more features added to the network after some point is almost useless during training, making the residual block arrangement 2,3,4,3,2 to outperformed 1,2,8,8,4 and 3,4,6,3.

4 CONCLUSIONS AND FUTURE WORK

A robust YOLOFig detection model was proposed in this paper to solve fruit detection challenges and to improve fruit detection accuracy and speed. The YOLOFig model is composed of ResNet43 backbone (2,3,4,3,2 block arranged) activated with Leaky, SPPNet, FPN, DIoU−NMS, and CIoU loss to improve detection performance. The ablation studies on different modifications of YOLOFig model showed that the Leaky activation function performed better than ReLU, Swish and Mish, and FPN outperformed PANet as model Neck. This demonstrates that Leaky and FPN are most compatible with ResNet architecture, which call for future studies. The compared new residual block arrangement of 2,3,4,3,2 outperformed 1,2,8,8,4 on an average AP of 1.33% and detection speed of 15.2%.

The 2,3,4,3,2 block arrangement of backbone corresponds to YOLOv3b, YOLOv4b and YOLOFig, while 1,2,8,8,4 block corresponds to YOLOv3, YOLOv4 and YOLOResNet70. In all, the YOLOFig detection model with AP of 89.3% and 96.8 fps is outstanding in performance compared to YOLOv3b with AP of 78.6% and 69.8 fps, YOLOv3 with AP of 77.1% and 56.3 fps, YOLOv4b with an AP of 87.6% and 57.1 fps, YOLOv4 with an AP of 87.1% and 52.5 fps, YOLOResNet50 with AP of 85.4% and 77.1 fps, and YOLOResNet70 with 87.3% of AP and 79 fps. Finally, YOLOFig model is highly prospective for better generalization, real-time fruit detection, applicable to harvesting robots and can be extended to other fruits for detection.

However, the detection accuracy is not satisfactory and the model could not detect to differentiate between unripe, ripe, and over-ripe figs, which call for further improvement and investigation. The practical applications of harvesting robot equipped with the proposed detection method would also be part of the focus in the future work.

ACKNOWLEDGEMENTS

We wish to acknowledge Prof. Zheng Decong, Wang Zhe, and the College of Engineering staff for their advice and supports during this research work.

ORCID

Olarewaju Mubashiru Lawal https://orcid.org/0000-0003-0011-6313
FIGURE 8  Feature maps of model backbone block arrangement at different detection scale for (a) DarkNet in YOLOv3b, YOLOv3, (b) CSPDarkNet in YOLOv4b, YOLOv4, and (c) ResNet in YOLOFig, YOLOResNet50, YOLOResNet70

REFERENCES

1. Lianju, W., et al.: The production and research of fig (Ficus carica L.) in China. Acta Hortic. 605, 191 (2003)
2. Food and Agriculture Organization of United Nations (FAO, 2020) https://www.atlasbig.com/en-cn/countries-by-figs-production
3. Hua, Y., et al.: Recent advances in intelligent automated fruit harvesting robots. open Agric. J. 13, 101–106 (2019)
4. Tang, Y.C., et al.: Recognition and localization methods for vision-based fruit picking robots: a review. Front. Plant Sci. 11, 510 (2020)
5. Koirala, A., et al.: Deep learning – Method overview and review of use for fruit detection and yield estimation. Comput. Electron. Agric. 162, 219–234 (2019)
6. Ren, S., et al.: Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 39, 1137–1149 (2017)
7. Liu, W., et al.: SSD: Single shot multibox detector. In: European Conference on Computer Vision (ECCV), Amsterdam, Netherlands, 11–14 Oct, 2016
8. Sa, I., et al.: Deepfruits: A fruit detection system using deep neural networks. Sensors 16(8), 1222 (2016)
9. Bargoti, S., Underwood, J.: Deep fruit detection in orchards. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA), Singapore, Singapore, 29 May–3 June 2017
10. Koirala, A., et al.: Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of ‘MangoYOLO.’ Prec. Agric. 20, 1107–1135 (2019)
11. Zheng, Y.Y., et al: CropDeep: the crop vision dataset for deep learning based classification and detection in precision agriculture. Sensors 19(5), 1058 (2019)
12. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. Preprint arXiv: 1804.02767 (2018).
13. Lawal, M.O.: Tomato detection based on modified YOLOv3 framework. Sci. Rep. 11, 1447 (2021)
14. Bochkovskiy, A., et al.: YOLOv4: Optimal speed and accuracy of object detection. Preprint arXiv: 2004.10934v1 (2020)
15. Maas, A.L., et al.: Rectifier nonlinearities improve neural network acoustic models. In: International Conference on Machine Learning (ICML), 2013
16. Misra, D.: Mish: A self-regularized nonmonotonic neural activation function. Preprint arXiv:1908.08681 (2019).
17. Lin, T.Y., et al.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017
18. He, K., et al.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI), 37, 1904–1916 (2015)
19. Liu, S., et al.: Path aggregation network for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018
20. Lawal, O.M.: Development of tomato detection model for robotic platform using deep learning. Multimedia Tools Appl. 1–22 (2021)
21. He, K., et al.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016
22. Lawal, M.O., YOLOMuskmelon: quest for fruit detection speed and accuracy using deep learning. IEEE Access. 9, 15221–15227 (2021)
23. Zheng, Z., et al.: Distance-IoU loss: faster and better learning for bounding box regression. Preprint arXiv:1911.08267v1 (2019)