



Fast and accurate detection of kiwifruit in orchard using improved YOLOv3-tiny model

Longsheng Fu^{1,3,4,5} · Yali Feng^{1,6} · Jingzhu Wu² · Zhihao Liu¹ · Fangfang Gao¹ ·
Yaqoob Majeed⁵ · Ahmad Al-Mallahi⁷ · Qin Zhang⁵ · Rui Li¹ · Yongjie Cui¹

Published online: 13 September 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Automatic detection of kiwifruit in the orchard is challenging because illumination varies through the day and night and because of color similarity between kiwifruit and the complex background of leaves, branches and stems. Also, kiwifruits grow in clusters, which may result in having occluded and touching fruits. A fast and accurate object detection algorithm was developed to automatically detect kiwifruits in the orchard by improving the YOLOv3-tiny model. Based on the characteristics of kiwifruit images, two convolutional kernels of 3×3 and 1×1 were added to the fifth and sixth convolution layers of the YOLOv3-tiny model, respectively, to develop a deep YOLOv3-tiny (DY3TNet) model. It takes multiple 1×1 convolutional layers in intermediate layers of the network to reduce the computational complexity. Testing images captured from day and night and comparing with other deep learning models, namely, Faster R-CNN with ZFNet, Faster R-CNN with VGG16, YOLOv2 and YOLOv3-tiny, the DY3TNet model achieved the highest average precision of 0.9005 with the smallest data weight of 27 MB. Furthermore, it took only 34 ms on average to process an image of a resolution of 2352×1568 pixels. The DY3TNet model, along with the YOLOv3-tiny model, showed better performance on images captured with flash than those without. Moreover, the experiments indicated that the image augmentation process could improve the detection performance, and a simple lighting arrangement could improve the success rate of detection in the orchard. The experimental results demonstrated that the improved DY3TNet model is small and efficient and that it would increase the applicability of real-time kiwifruit detection in the orchard even when small hardware devices are used.

Keywords Data augmentation · Image detection · Deep learning · YOLOv3-tiny model · Convolutional kernel

✉ Longsheng Fu
fulsh@nwafu.edu.cn; longsheng.fu@wsu.edu

Extended author information available on the last page of the article

Introduction

China is the largest country producing kiwifruits worldwide, with a yield of 2,390,287 t in 2016 from a cultivated area of 197,048 ha (UN FAO 2018). Within China, Shaanxi Province has the most significant production, accounting for approximately 70% and 33% of the Chinese and global productions, respectively (Hu et al. 2017). Harvesting kiwifruits in this area mainly depends on manual picking, which is labor-intensive (Fu et al. 2016), and introducing mechanical harvesting is needed.

Kiwifruits are commercially grown on sturdy support structures such as T-bars and pergolas. The T-bar trellis is common in China because of its low cost (Lu et al. 2016). It consists of a 1.7-m high post and approximately a 1.7-m wide cross arm, which may have slightly different widths depending on the orchard geometry. Wires run on the top of cross arms and connect them from the middle on both sides. The upper stems of the kiwi plants are tied to the top wires so that the egg-sized kiwifruits are hanging downwards, making them easy to be picked during the harvest season (Fu et al. 2015; Mu et al. 2018). This workspace is more structured than with other fruit trees, and thus easier to perform mechanical work. The setback is that kiwifruits grow in clusters, which make the fruits occluded and adjacent to each other.

Like other orchard fruits such as apples (Silwal et al. 2017; Liu et al. 2018; Fu et al. 2020; Gao et al. 2020) and citrus (Wang et al. 2018; Zhuang et al. 2018; Lin et al. 2020), it is necessary to design an intelligent robotic machine with human-like perceptive capability. Fast and effective detection of kiwifruit in the orchard under natural scenes is essential and the first step for its robotic harvesting system. Research on kiwifruits detection is mainly being conducted in China and New Zealand, because China is the largest kiwi fruit producer while New Zealand is the second largest producer. Scarfe (2012) subtracted a predefined reference RGB (Red, Green and Blue) color range and used a Sobel filter to detect fruit and calyx edges, then used template matching to detect kiwifruit but didn't use the shape information of the fruit. Fu et al. (2015) segmented bottom-viewed kiwifruit images using the Otsu threshold in a 1.1R-G color component, and used minimal bounding rectangle and elliptical Hough transform to detect fruits in single cluster. Fu et al. (2017) developed a kiwifruit detection system for night use using artificial lighting by identifying the fruit calyx, which detected 94.3% of target fruits and took 0.5 s on average to recognize a fruit. Fu et al. (2019) separated linearly clustered kiwifruits by scanning each detected cluster to find the contact points between the adjacent fruits and drawing a separating line between the two closest contact points, which correctly separated and counted 92.0% of the kiwifruits. Most of these traditional methods utilized hand-engineered features to encode visual attributes that discriminate fruit from non-fruit regions. Although these approaches were well suited for the dataset that they were designed for, feature encoding was generally unique to a specific kiwifruit and the conditions under which the data were captured (Fu et al. 2018a). Therefore, it is necessary to find a general feature extraction model to overcome the limitations of the traditional image detection model limited by their algorithm.

In recent years, deep learning as a powerful technique in the artificial intelligence field is becoming a prevalent way of object detection and semantic segmentation. It could learn the differences between similar things autonomously and transform the original data into a higher level and more abstract expression through training of non-linear models (Peng et al. 2018; Russakovsky et al. 2015; Simonyan and Zisserman 2014; Zhou et al. 2018). Sa et al. (2016) might be the first work exploring the use of deep learning networks for fruit detection. Wang (2017) established PCANet deep learning model to identify kiwifruit with

a detection rate of 94.9%, but it was limited to detect objects in a single cluster with few fruits. Fu et al. (2018b) used Faster R-CNN (Region Convolutional Neural Network) with ZFNet (Zeiler and Fergus Network) that depends on feature extraction and region proposal networks named two-stage detection on kiwifruit images, which achieved an AP (average precision) of 0.92 and took 0.27 s on average to process an image with 2352×1568 pixels. Williams et al. (2019) employed Fully-Convolutional Network (FCN) with VGG16 to perform semantic segmentation for calyx, cane and wire in a kiwifruit image of 1900×1200 pixels, which detected 76.3% target fruit with an average processing time of 3 s. Liu et al. (2020) improved Faster R-CNN by combining two VGG16 architecture for feature extraction to detect kiwifruits from color and infrared images, and reached an AP of 0.91 with average processing time of 0.13 s on kiwifruit images of 512×424 pixels.

Some researches employed recent two-stage detection algorithms for other fruits. Yu et al. (2019) used Mask R-CNN with ResNet-50 and FPN (Feature Pyramid Network) architecture for feature extraction of ripe and unripe strawberry, which obtained mIoU (mean Intersection over Union), overall precision and recall rates of 89.9%, 95.8% and 95.4%, respectively and took 0.13 s on a images of 640×480 pixels. Williams et al. (2020) employed Faster R-CNN with Inception V2 to detect kiwifruit and its calyx, which required 0.2 s on an image of 2100×1700 pixels and reached APs of 0.91 and 0.94 for the calyx and kiwifruit, respectively. Jia et al. (2020) applied Mask R-CNN with ResNet and DenseNet to detect overlapped apples, which achieved precision and recall rates of 97.3% and 95.7% and took 0.12 s on an image of 512×341 pixels. Gené-Mola et al. (2020) tried Mask R-CNN for apple detection and reported an AP of 0.86 and F1-score of 0.86, which required 3.6 s to process an image of 1024×1024 pixels. Vasconez et al. (2020) applied Faster R-CNN with Inception V2 to detect avocado, lemon and apple under different field conditions, which achieved mean AP of 0.93 and needed approximately 0.22 s on average to compute an image of 640×360 pixels. However, those two-stage detection method required large amounts of resources for selecting region proposal, which still shows limitations in the detection speed so that it cannot be applied for field real time detection.

Unlike the two-stage detection pipeline that first predicts proposals and then refines them, single-stage detectors directly predict the final detections. YOLO (You Only Look Once) is the most representative work with real-time speed, which divides the image into sparse grids and makes multi-class and multi-scale predictions per grid cell (Redmon et al. 2016). Redmon and Farhadi (2017, 2018) also presented YOLOv2 and YOLOv3 to improve the performance of YOLO. The YOLOv3 uses a deeper convolutional model and three size layers to predict the detection object so that it has better ability for feature extraction for small object detection than other region-based methods. Tian et al. (2019) employed YOLOv3 with DenseNet to detect apples in orchards, which reached an F1-score of 0.817 and IoU of 0.896 and required 0.304 s to process an image of 4000×3000 pixels. Koirala et al. (2019) merged feature maps with different resolutions from intermediate layers to improve YOLOv3 network for mango detection, which achieved an AP of 0.98 and spent 0.07 s per 2048×2048 pixel image. Liu et al. (2020) replaced the traditional rectangular bounding box with a circular bounding box in YOLOv3 model for tomato detection, which obtained an AP of 0.96 and detection speed of 0.054 s on an image of 3648×2056 pixels. To the authors' knowledge, there has been no research on applying YOLO models to kiwifruit detection. Therefore, this paper considers applying the YOLO model to the detection of kiwifruit.

Although those studies drastically reduced the detection speed to around 0.03 s for an image with high resolution of more than 1920×1080 pixel, the YOLOv3 network requires a powerful GPU (Graphic Processing Unit) with more than 4 GB (Gigabyte) memory,

which is a hardware challenge for most computers. On the other hand, YOLOv3-tiny model, which is a reduced version of YOLOv3 for further faster processing and has the potential to be applied in portable devices, could be trained with only 1 GB GPU (Huang et al. 2018). It is a smaller version of YOLOv3 algorithm based on one-stage detection method. The network structure of YOLOv3-tiny model is a simple lighter model containing a reduced number of layers that enables faster performance of YOLOv3-tiny model. In general, YOLOv3-tiny model is much quicker than YOLOv3 model and can meet the requirements of real-time application. However, the network structure of YOLOv3-tiny model only has a two-size layer to predict the detection object, which may cause a problem of precision as it may miss some small objects (Yang et al. 2019), such as small kiwifruit objects in a far view image.

To improve the overall detection accuracy of deep learning networks, researchers have done some work in making the convolutional neural networks deeper. Szegedy et al. (2015) proposed a deep convolutional neural network architecture codenamed Inception, which was based on the Hebbian principle and the intuition of multi-scale processing. It allowed for increasing the depth of the network while keeping the computational budget constant. He et al. (2016) addressed the degradation problem by introducing a deep residual learning framework, which evaluated residual nets with a depth of up to 152 layers; 8 deeper than the VGG16 on the ImageNet dataset. An ensemble of these residual nets achieved 3.57% error on the ImageNet dataset. Also, the deep residual nets can easily enjoy accuracy gains from greatly increased depth, producing results substantially better than previous networks. These researchers showed that the depth of representations is of central importance for many visual recognition tasks and deep convolutional networks, which can improve the precision of network detection significantly.

To meet all-day-long operation requirements of a multi-arm kiwifruit picking robot in commercial orchards, it is necessary to increase the kiwifruit detection speeds while maintaining high detection accuracy. Therefore, in this study, a detection model based on the YOLOv3-tiny algorithm for kiwifruit in the orchard was developed. A deep YOLOv3-tiny network (DY3TNet) model was proposed and tested by introducing deep convolutional networks into the YOLOv3-tiny model. The goal was to support the multi-arm operations of robotic harvesting and fruit picking technology.

Materials

Image acquisition

The images for this application were captured using a camera placed underneath the fruits, with its central axis perpendicular to the canopy. An ordinary single-lens reflex camera (Canon S110, Canon Inc., Tokyo, Japan) on "P" mode with a resolution of 2352 × 1568 pixels was used. It was placed at around 1 m underneath the fruits, which is the same position of the vision system as in the kiwifruit harvesting robot prototype of this research work (Mu et al. 2018). RGB images of 'Hayward' kiwifruits were taken during three harvest seasons of 2016, 2017 and 2018 from Meixian Kiwifruit Experimental Station (34°07'39"N, 107°59'50"E, and 648 m above sea level), Northwest A&F University, Shaanxi, China.

Images of the kiwifruits were randomly captured at three different times (morning, afternoon and night). At each time, 400 different positions far from each other were selected to make sure the images do not contain overlapping regions. Two images in two

different illumination conditions (with or without flash of the camera) were acquired, at each position, in the morning and afternoon, respectively, as shown in Fig. 1a to Fig. 1d. At night, two images were acquired with either white LED (Light Emitting Diode) illumination or flash, as shown in Fig. 1e, f. The LED illumination produced an average illumination of 40 lx (± 10 lx) in the imaging region. In total, 1200 pairs of images (2400 total, with 800 each taken in the morning, afternoon and night) were collected, and each image included around 30 to 50 fruit.

The overall dataset of 1200 pairs of 2400 images was divided into raw training datasets (60% of the images) and testing datasets (40% of the images), as shown in Table 1. The raw training datasets included 720 pairs of 1,440 original images, which were randomly selected from the overall dataset. The remaining 480 pairs of 960 original images were set as the testing datasets. All the datasets were separated into two groups based on imaging with or without flash. For example, the raw training datasets were divided into two groups: 720 images with flash (AD720F) and 720 images without flash (AD720NF). Each group, in the case of AD720F, was divided into three subgroups based on the three different imaging times: morning (M240F), afternoon (A240F) and night (N240F). Table 1 lists the detailed information for each group and subgroup. The aim was to test the sensitivity of the proposed network under different daytime and illumination conditions.

Data augmentation

Deep learning for object detection requires a large dataset of images to provide generalization and robust performance. Zhang et al. (2017a) and Sun et al. (2017) found that a broader set of image data could improve the success rate of object detection. However, the raw training datasets in this study only contained self-collected 1440 images (AD720F and AD720NF). To address this issue, the raw training datasets were augmented. Data augmentation is a common way to expand the variability of the training data by artificially enlarging the dataset using label-preserving transformations (Bargoti and Underwood 2017). More training images can increase the network capability to generalize and reduce overfitting (Bargoti and Underwood

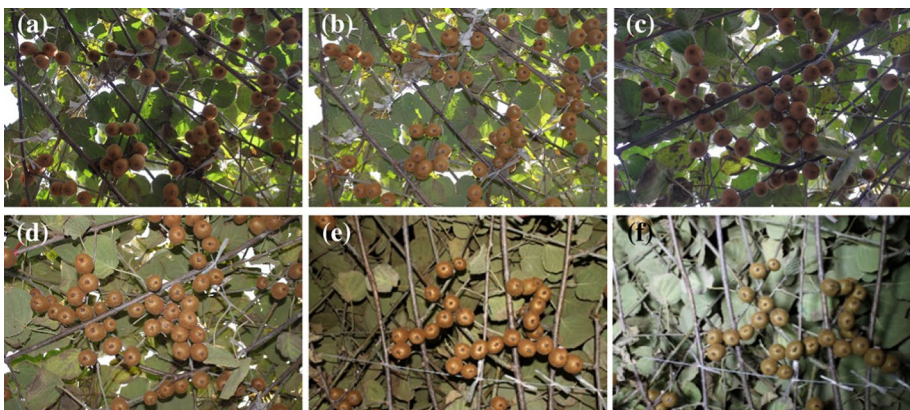


Fig. 1 Kiwifruit images under different illumination conditions in the orchard environment. **a** Morning without flash, **b** Morning with flash, **c** Afternoon without flash, **d** Afternoon with flash, **e** Night with flash, **f** Night with illuminations (Color figure online)

Table 1 Datasets of kiwifruit images for deep learning

Datasets	Time	Raw		Augmented	
		No flash/LED illumination	Flash	No flash/LED illumination	Flash
Training datasets	Morning	240 (M240NF)	240 (M240F)	3360 (N3360NF)	3360 (M3360F)
	Afternoon	240 (A240NF)	240 (A240F)	3360 (A3360NF)	3360 (A3360F)
	Night	240 (N240NF)	240 (N240F)	3360 (N3360NF)	3360 (N3360F)
	All-day	720 (AD720NF)	720 (AD720F)	10,080 (AD10080NF)	10,080 (AD10080F)
Testing datasets	Morning	160 (M160NF)	160 (M160F)	–	–
	Afternoon	160 (A160NF)	160 (A160F)	–	–
	Night	160 (N160NF)	160 (N160F)	–	–
	All-day	480 (AD480NF)	480 (AD480F)	–	–

In this study, the terms were defined as follows: M, A, N and AD represents images taken in the morning, afternoon, night and all-day, respectively. All-day images are simply the summation of the images taken at any time. For the morning and afternoon, F and NF represent kiwifruit images captured under ambient light with and without flash respectively. For the night, F and NF represent kiwifruit images captured with flash or LED illumination respectively

2017). To achieve sensitive detection of kiwifruit in the orchard, this study took into consideration most kinds of interference that may occur when detecting fruits. As described in Taylor and Nitschke (2018), Shorten and Khoshgoftaar (2019) and Tian et al. (2019), data augmentation methods such as color brightness transformation, image rotating and histogram equalization can improve the network performance. Data augmentation, including adaptive histogram equalization, brightness transformation, motion blur transformation and image rotation, were implemented using the software Matlab R2018b (Math Works Inc., Natick, MA, USA). The specific augmented methods were described as follows.

Firstly, the adaptive histogram equalization method was used to improve the quality of the training sample images and the variety of illuminations. The original RGB (Red, Green, Blue) color image was converted to HSV (Hue, Saturation, Value) color space using the Matlab function ‘rgb2hsv’ (Smith 1978). Then the adaptive histogram equalization was performed on the V component of the HSV using the Matlab function ‘adaphiseq’ with default parameters (Pizer et al. 1987). After that, the new V component with original H and S were converted back to the RGB color image using the Matlab function ‘hsv2rgb’ (Smith 1978), which was employed as the augmented image by the adaptive histogram equalization method.

Secondly, the brightness transformation was applied six times in this study to enhance the illumination range of the raw training datasets. The brightness transformation is a common data augmentation method often used to improve the robustness of a network to brightness variation in different environments, such as apple detection in the field (Tian et al. 2019). Multiplying a proportional coefficient near 1.0 by the original RGB image, which can adjust the value of each color component to make the image brightness higher or lower (Tian et al. 2019), as shown in Eq. (1). Manual annotation is based on the process of visually observing the outline of the fruit on the image and using a rectangular frame to mark the fruit one by one. Therefore, if the image brightness is too high or too low, bounding boxes will be difficult to draw during manual annotation because the edge of the target is unclear. Coefficients k of 0.7–0.9 and 1.1–1.3 in increments of 0.1 were selected based on the target edge which can be accurately identified during manual annotation.

$$g(x, y) = \begin{cases} f(x, y) \times k, & \text{if } g(x, y) < 255 \\ 255, & \text{if } g(x, y) \geq 255 \end{cases} \quad (1)$$

where $f(x, y)$ is the original RGB image and $g(x, y)$ is an RGB image after brightness change. If the multiplied value was higher than 255, it was automatically adjusted to 255.

Thirdly, the motion blur transformation was employed four times to make the convolutional network model to have strong adaptability to blurred images. The degradation function of a motion blurred image is shown in Eqs. (2) and (3) (Tani et al. 2016). Since the telephoto distance of the camera, incorrect focusing and camera movement would cause blurred images that are difficult to estimate, parameters L and θ of the motion filter were employed. L (length, represents pixels of linear motion of camera) and θ (theta, represents angle between horizontal line and direction of camera movement) were set as (20, –15), (20, 15), (30, –20) and (30, 20), respectively.

$$g(x, y) = h(x, y) * f(x, y) \quad (2)$$

$$h(x, y) = \begin{cases} 1/L, & \sqrt{x^2 + y^2} \leq L \text{ and } y/x = \tan \theta \\ 0, & \text{other} \end{cases} \quad (3)$$

where (x, y) are pixel co-ordinates in the image; $*$ is the spatial convolution operation; $g(x, y)$ is a motion blurred image, $h(x, y)$ is a degenerate function and $f(x, y)$ is the original image.

Finally, the original images were rotated by 90° and 270° using the Matlab function 'imrotate'. The rotated images can also improve the detection performance of the neural network by correctly identifying the kiwifruits at different orientations.

The raw training datasets were augmented 13 times (one time of histogram equalization, six times of brightness transformation, four times of motion blur transformation and two times of rotation) by the above methods, and the training images of each subgroup were augmented from 240 to 3360 (including the raw datasets), as shown in Table 1. In total, the training datasets were expanded from the raw 1440 images (AD720NF and AD720F) to 20,160 images (AD10080NF and AD10080F).

In order to verify whether the selected augmented method has an impact on the detection results, an ablation study on the augmented method was conducted in this study. Four tests were conducted by removing one of the four data augmentation transformations from the all expanded training dataset (AD10080NF and AD10080F), respectively. In the first test (HisTest), the dataset augmented by histogram equalization was removed to verify the effect of histogram equalization. In the second test (BriTest), the dataset augmented by brightness transformation was removed to verify the effect of brightness transformation. In the third test (MotTest), the dataset augmented by motion blur transformation was removed to verify the effect of motion blur transformation. In the fourth test (RotTest), the dataset augmented by rotation was removed to verify the effect of rotation. Ground truth data for network training and testing was created using manual labeling (using rectangular bounding boxes) of the fruits on all the training and testing dataset images.

Methodologies

Classical YOLO deep learning model

The separate components of object detection were unified into YOLO network, which uses features from target area in the image to predict each bounding box. The input image was divided into an $N \times N$ grid in YOLO. If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object. Each grid cell predicts bounding boxes and confidence scores for those boxes. If no object exists in that cell, the confidence scores should be zero. Each bounding box consists of five predictions: x, y, w, h and confidence. The (x, y) co-ordinates represent the center of the box relative to the bounds of the grid cell. The width (w) and height (h) are predicted relative to the whole image. Each grid cell also predicts conditional class probabilities. These probabilities are conditioned on the grid cell containing an object. One set of class probabilities are only predicted by each grid cell, regardless of the number of boxes (Shinde et al. 2018). The value of $\text{Pr}(\text{Object})$ is 1 when a grid cell contains a part of a ground truth box and 0 otherwise. The detection pipeline of YOLO is shown in Fig. 2.

YOLOv3-tiny model gives good trade off of speed and accuracy that developed from the one-stage detection method YOLOv3 (Redmon and Farhadi 2018; Liu et al., 2016; Ren et al. 2017). It uses deeper convolutional models than the Faster R-CNN with ZFNet and Faster R-CNN with VGG16, as shown in Table 2. It could achieve a good trade-off in improving the detection speed and accuracy to meet the real-time requirements. However,

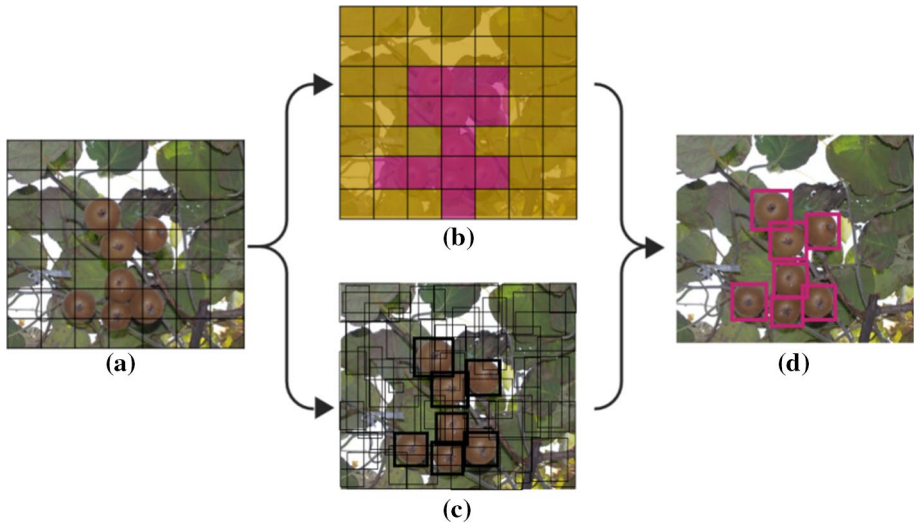


Fig. 2 Detection pipeline of YOLO. **a** $N \times N$ grids on input, **b** predict class probabilities map, **c** predict bounding boxes in each grid and confidence, **d** detection result (Color figure online)

the setback of the one-stage detector may be losing small objects as the sliding window scheme is used to detect candidate objects in the output feature map.

Improved YOLOv3-tiny model

As described earlier, the reduced nature of the YOLOv3-tiny model in comparison to YOLOv3 makes it a potential candidate for faster processing applications as it could be trained with only 1 GB GPU. Since the GeForce GTX 960 M 4 GB GPU used in the study is not powerful, the YOLOv3-tiny model that has a simple structure and low computational complexity was selected to meet the accuracy and real-time requirements. YOLOv3-tiny model contains 25 layers with 17 convolution layers and has fewer convolution layers than other single-stage detector methods. However, deeper convolution networks can contribute to the learning of objective features (He et al. 2016). In this study, two convolutional kernels of 1×1 and 3×3 were added to the fifth and sixth convolution layers of the YOLOv3-tiny model, respectively, to develop the DY3TNet model, as shown in Table 2.

The parameters in the hierarchical structure were adjusted to improve the neural network structure of the DY3TNet model. The 1×1 convolutional layers are reduction layers that can increase non-linearity without changing the receptive fields of the convolutional layers (Akçay et al. 2018; Zhang et al. 2017b). The 1×1 convolutional layer is equivalent to the cross channel parametric pooling layer, which may obtain the complex and learnable interaction information by crossing channels (Lin et al. 2013). It could maintain detailed information on small objects (Yang et al. 2019). The added 3×3 convolutional layers output feature maps of different sizes and channels, thus improving feature expression of the DY3TNet. The route layer in YOLO is mainly for concatenating shallow and deep features maps by specifying the index layer in different positions of network to improve the detection effect of kiwifruit at different scales, because shallow features contain more detailed information about kiwifruit, and deep features contain more contour information.

Table 2 Structures of the DY3TNet and four other deep learning models (Faster R-CNN with ZFNet, Faster R-CNN with VGG16, YOLOv2 and YOLOv3-tiny) used for comparison

Models	Faster R-CNN with ZFNet	Faster R-CNN with VGG16	YOLOv2	YOLOv3-tiny	DY3TNet
Input	224×224	224×224	416×416	416×416	512×512
	Conv7-96	Conv3-64	Conv3-32	Conv3-16	Conv3-16
	Maxpool/3	Conv3-64	Maxpool/2	Maxpool/2	Maxpool/2
	Conv5-256	Maxpool/2	Conv3-64	Conv3-32	Conv3-32
	Maxpool/3	Conv3-128	Maxpool/2	Maxpool/2	Maxpool/2
	Conv3-384	Conv3-128	Conv3-128	Conv3-64	Conv3-64
	Conv3-384	Maxpool/2	Conv1-64	Maxpool/2	Maxpool/2
	Conv3-256	Conv3-256	Conv3-128	Conv3-128	Conv3-128
	Conv3-256	Conv3-256	Maxpool/2	Maxpool/2	Maxpool/2
RoI Proposal		Conv3-256	Conv3-256	Conv3-256	Conv3-256
RoI Pooling		Maxpool/2	Conv1-128	Maxpool/2	Conv1-128
FC-4096		Conv3-512	Conv3-256	Conv3-512	Conv3-256
FC-4096		Conv3-512	Maxpool/2	Maxpool/2	Maxpool/2
Class_score		Conv3-512	Conv3-512	Conv3-1024	Conv3-512
Output	BoxpPrediction	Maxpool/2	Conv1-256	Conv1-256	Conv1-128
	Output	Conv3-512	Conv3-512	Conv3-512	Conv3-256
		RoI proposal	Conv1-256	Conv1-255	Maxpool/2
		RoI pooling	Conv3-512	Detection	Conv3-1024
		FC-4096	Conv3-512	Route -4	Conv1-256
		FC-4096	Maxpool/2	Conv1-128	Conv3-512
		Class_score	Conv3-1024	Upsample	Conv1-255
		Output	Conv1-512	Route -1,8	Detection
		Box_prediction	Conv3-1024	Conv3-256	Route -4
		Output	Conv1-512	Conv1-255	Conv1-128

Table 2 (continued)

Models	Faster R-CNN with ZFNet	Faster R-CNN with VGG16	YOLOv2	YOLOv3-tiny	DY3TNet
			Conv3-1024	Detection	Upsample
			Conv3-1024		Route -1,8
			Route -9		Conv3-256
			Conv1-64		Conv1-255
			Reorg/Route -1,-4		Detection
			Conv3-1024		
			Conv1-425		
			Detection		

Bold values indicate the two convolutional kernels of 1×1 and 3×3 that added to the fifth and sixth convolution layers of the YOLOv3-tiny model

Proposal boxes with different sizes, namely anchors, were generated in the detection layer to generate predicted candidates' boxes (Redmon and Farhadi 2018). The IoU is calculated by the predicted bounding box (P) and ground truth (G) using Eq. (4) to select anchors around the ground truth (kiwifruit) as candidates. The training objective is to reduce losses between the P and G, and the loss ($loss_{iou}$) is defined in Eq. (5).

$$IoU = \frac{area(P) \cap area(G)}{area(P) \cup area(G)} \tag{4}$$

$$loss_{iou} = \sum_{i=1}^{S^2} \sum_{j=1}^B 1_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=1}^{S^2} \sum_{j=1}^B 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 \tag{5}$$

where the λ_{noobj} is the weight of the IoU loss, S^2 is the number of grids in the input image, and B is the number of bounding boxes generated by each grid. $1_{ij}^{obj} = 1$ denotes that the object falls into the j_{th} bounding box in grid i , otherwise $1_{ij}^{obj} = 0$. \hat{C}_i is the predicted confidence and C_i is the true confidence.

As the kiwifruit sizes vary in the orchard, a multi-scale training strategy was employed for kiwifruit detection to let the DY3TNet model have good detection effect on different input image sizes. Because the kiwifruit in the image looked small and dense, the main problem is that the fine features of the object extracted from the shallow-layer are not obvious and the features from the deep-layer extraction may lose the object information (Yang et al. 2019). To increase the kiwifruit detection accuracy, higher resolution inputs and multi-scale strategy were employed to train the DY3TNet. The input image size was modified from 416×416 pixels of the YOLOv3-tiny to 512×512 pixels which is the highest affordable image size for the computer hardware employed in this study. During network training, 10 different training scales of 288×288 , 320×320 , 352×352 , 384×384 , 416×416 , 448×448 , 480×480 , 512×512 , 544×544 and 576×576 were resized from the input image, and each of them was randomly selected for training in every ten batches. This training strategy helped to make the network have good performance on different image sizes.

As shown in Fig. 3, the DY3TNet model was constructed using the end-to-end detection method to achieve fast operation in the orchard. It performs up-sampling in the last layer and uses a small box to detect the kiwifruit objects on a large-scale feature map. This study used the layer extraction network with a 16×16 feature map to predict the kiwifruit bounding box co-ordinates and confidence value of the kiwifruit probabilities by three anchor boxes. Besides, a 32×32 feature map of up-sampling on the last layer was also

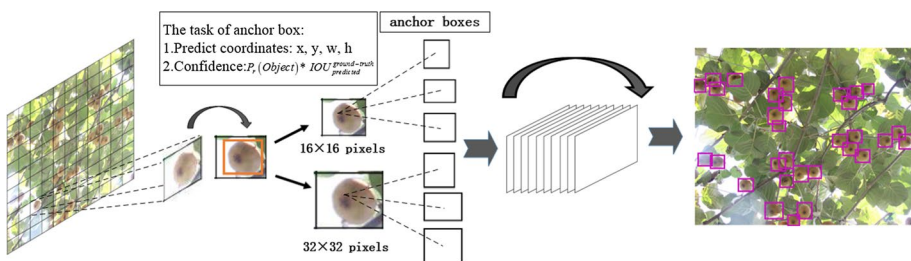


Fig. 3 The pipeline of the DY3TNet model for kiwifruit detection using two-size feature maps (Color figure online)

used to predict the detection results. The detection results of both feature maps were then compared to determine the final detection results, which might be possible to improve the accuracy by using the two features maps instead of just one.

To observe the accuracy, applicability and stability of the proposed DY3TNet model for kiwifruit detection, four other competing techniques for contemporary detection models, including Faster R-CNN with ZFNet, Faster R-CNN with VGG16, YOLOv2 and YOLOv3-tiny model, were also carried out on the same datasets, as shown in Table 2. Each of these networks has its strengths on image detection and presents many efficient tricks by building blocks for constructing deep learning networks (Zhang et al. 2017b). Also, all of them can be implemented on the 4 GB GPU computer employed in this study.

Network training

The training platform was a desktop computer with Intel i5 6400 (2.70 GHz) quad-core CPU, a GeForce GTX 960 M 4 GB GPU (1536 CUDA cores), and 16 GB of memory, running on a Windows 7 64 bits system. The software tools used included CUDA 7.5, CUDNN 5.0, OpenCV3.0, Pthread and Microsoft Visual Studio 2013.

To train the deep learning networks, two sets of data were required, including the images and a corresponding label for each image. The labeling data comprised the object type along with the normalized center co-ordinates, followed by the normalized width and height of the kiwifruit bounding box. The stochastic gradient descent (SGD) was used to train the DY3TNet model with a mini-batch size of 64, and the momentum of the network was set to a fixed value of 0.9 and a weight decay of 0.0005. In this work, a learning rate of 0.001 was applied for all layers in the network. It took about 12 h to perform a total of 10,000 iterations over the training set. To provide well-differentiated weights for object and background, leading to faster and more accurate training results, the transfer learning method was utilized for training the DY3TNet model. One of the advantages of transfer learning is that a network trained with a small ground-truth dataset can also reach a high detection accuracy. Therefore, transfer learning from ImageNet for YOLOv3-tiny darknet framework was carried out. A fine-tuning method was also applied to modify the DY3TNet model so that it can be more suitable for the detection of kiwifruit images. During training, the convolutional neural network fine-tuned these weights by adjusting them to minimize the functional loss to classify the object as annotated training images through a supervised learning process. The other deep learning networks were also trained in the same parameters.

All the deep learning networks were firstly trained using the all-day augmented training datasets, which included both the images with flash (AD10080F) and without flash (AD10080NF); and tested on the all-day testing datasets that also included both the images with flash (AD480F) and without flash (AD480NF). The new DY3TNet model and its original YOLOv3-tiny model were then compared specifically on all-day augmented training datasets of images with flash (AD10080F) or without flash (AD10080NF) only and also tested using all-day testing datasets AD480F or AD480NF, respectively. The goal was to investigate how the flash could influence kiwifruit detection in the orchard. Besides, the new DY3TNet model was trained and tested on all the different datasets, as well as small datasets (such as the images with flash in the morning M240F and M160F), to evaluate its performance on different imaging times, illumination condition and data size.

Evaluation

The performance of the models was evaluated by precision (P), recall (R), average precision (AP) and detection speed. Among them, the P and R are defined in Eq. (6) and Eq. (7) respectively.

$$P = TP/(TP + FP) \quad (6)$$

$$R = TP/(TP + FN) \quad (7)$$

where TP, FP and FN mean the number of correctly detected kiwifruit objects (true positives), the number of falsely detected kiwifruit objects (false positives), and the number of missed kiwifruit objects (false negatives), respectively.

AP is defined in Eq. (8) as the area under the P and R curve. It is a standard for measuring the sensitivity of the network to an object, and an indicator that reflects the global performance of the network. The speed of the five models were tested on the same computer for network training with an input image resolution of 2352×1568 pixels.

$$AP = \int_0^1 P_{(R)} dR \quad (8)$$

Results and discussion

Comparison of DY3TNet model with other deep learning models

The detection results of all the deep learning networks trained on AD10080NF and AD10080F datasets and tested on AD480NF and AD480F datasets are shown in Table 3. The AP of the DY3TNet model was higher than the other four networks, and it was 0.9005 for kiwifruit images acquired from the orchard under different illumination conditions of day and night, which was 17.55%, 2.89%, 11.45% and 2.04% higher than Faster R-CNN with ZFNet (0.7250), Faster R-CNN with VGG16 (0.8761), YOLOv2 (0.7860) and YOLOv3-tiny model (0.8801) respectively. It shows that the DY3TNet model has more sensitivity to lighting variations than the other four networks. The AP of the Faster R-CNN with ZFNet was less than the 0.9230 that was obtained by Fu et al. (2018b), which was trained and tested only on kiwifruit images in the daytime and captured without flash.

Table 3 Detection results of each deep learning network trained on AD10080NF and AD10080F datasets and tested on AD480NF and AD480F datasets

Models	AP	Detection speed (ms)	Size (MB)
Faster R-CNN with ZFNet	0.7250	270 ± 10.5^a	225
Faster R-CNN with VGG16	0.8761	347 ± 12.0^b	512
YOLOv2	0.7860	54 ± 0.4^c	192
YOLOv3-tiny	0.8801	31 ± 0.5^d	33
DY3TNet	0.9005	34 ± 0.4^e	27

Same letters in the third column represent no significant difference at the 0.05 level

On the same image dataset of Fu et al. (2018b), another deep learning network (LeNet) obtained an AP of 0.8929 (Fu et al. 2018a).

Kiwifruit image detection examples of the five models on an image captured at morning with flash are shown in Fig. 4. Many kiwifruits in Fig. 4a (12 out of 50) were false negatives detected in the Faster R-CNN with ZFNet model showing its low performance. Likewise, the low AP of YOLOv2 was caused by the many false positive detected kiwifruits where fruits were adjacent to each other, as shown in Fig. 4c. The same phenomenon was also reported by Xue et al. (2018), who applied YOLOv2 to detect mango in an orchard. Some of the fruits partly covered by branches (the yellow circle mark at the upright of Fig. 4b) or leaves (two yellow circle marks at the bottom-left of Fig. 4d) could not be detected by the Faster R-CNN with VGG16 or YOLOv3-tiny but were successfully identified by the DY3TNet.

In terms of the detection time, the DY3TNet model took 34 ms on average per image, which is 3 ms longer than the YOLOv3-tiny model (31 ms) but noticeably shorter than that of the YOLOv2 (54 ms), Faster R-CNN with VGG16 (347 ms) and Faster R-CNN with ZFNet (270 ms). The detection time of the Faster R-CNN with ZFNet was similar to the speed obtained by Fu et al. (2018b) (274 ms), who used the same image resolution (2352×1568 pixels). All the above models were faster than LeNet (Fu et al. 2018a), which required 270 ms to detect each fruit on average. If every kiwifruit image has 40 fruits on average, the LeNet would spend 10,800 ms to process one image. LeNet, in turn, still spends less average time in detecting each kiwifruit if compared to any of the traditional image processing algorithms, which were 280 ms (Scarfe 2012), 1640 ms (Fu et al. 2015) and 500 ms (Fu et al. 2017).

The size of the network is another index that is used to evaluate different deep learning models, especially for off-line field real-time application and further application in a portable device. YOLOv3-tiny was 33 MB, which is less than Faster R-CNN with ZFNet

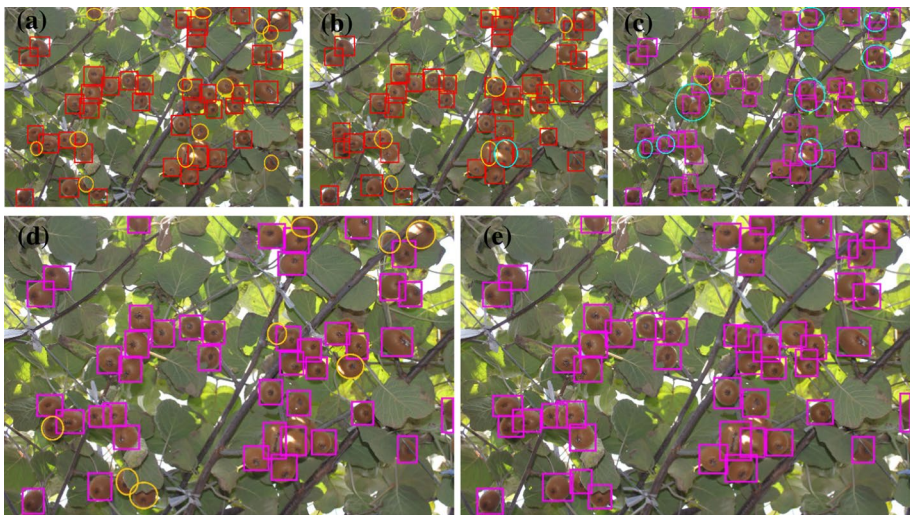


Fig. 4 Kiwifruit image detection examples of the five deep learning models. **a** Faster R-CNN with ZFNet. **b** Faster R-CNN with VGG16. **c** YOLOv2. **d** YOLOv3-tiny and **e** DY3TNet on an image captured in the morning with flash. *Note* The yellow and aqua circles highlight the undetected and wrongly detected kiwifruits, respectively (Color figure online)

(225 MB), Faster R-CNN with VGG16 (512 MB) and YOLOv2 (192 MB). However, the developed DY3TNet model was the smallest (27 MB), although it was modified from the YOLOv3-tiny model by adding two convolutional kernels of 3×3 and 1×1 to the fifth and sixth convolution layers to the model. The reason might be that the 1×1 followed by the 3×3 convolutional layers are reduction layers, which can increase the non-linearity without changing the receptive fields of the layers and avoid the computational complexity of the new structure.

Besides, the AP of the DY3TNet was decreasing as the required IoU threshold increased, as shown in Fig. 5. The IoU was changed from 0.1 to 1.0 with an interval of 0.05. The AP was evaluated on the AD480NF and AD480F datasets by the DY3TNet trained on the AD10080NF and AD10080F datasets. The AP was slowly decreasing as the IoU increased from 0.1 to 0.75, but suddenly dropped largely from 0.75 to 1.0. For a more accurate localization while maintaining a high success rate, IoU of 0.75 can be applied.

Overall, the DY3TNet model was more efficient and sensitive than the other deep learning models for the 960 testing images of all-day, and it maintained the fast speed of one-stage detectors and achieved a breakthrough in detection accuracies for kiwifruit in the orchard. In addition, the DY3TNet model produced a good trade-off in improving the running speed and reducing memory. The algorithm of this work indicated that the DY3TNet model could provide reliable support for field working requirements.

Comparison of DY3TNet and YOLOv3-tiny models on images with/without flash

As the two best-performed models on all the images datasets, the DY3TNet and YOLOv3-tiny models were evaluated on kiwifruit images with and without flash. The training was on the AD10080F and AD10080NF and the testing on the AD480F and AD480NF datasets, which are illustrated in Fig. 6; Table 4.

The P–R curves of the YOLOv3-tiny and DY3TNet models are shown in Fig. 6. The P of the DY3TNet model was higher than the YOLOv3-tiny model under the same R condition on both datasets. The detection results of the AD480F datasets (images with flash)

Fig. 5 AP of the DY3TNet decreased as the required IoU threshold increased

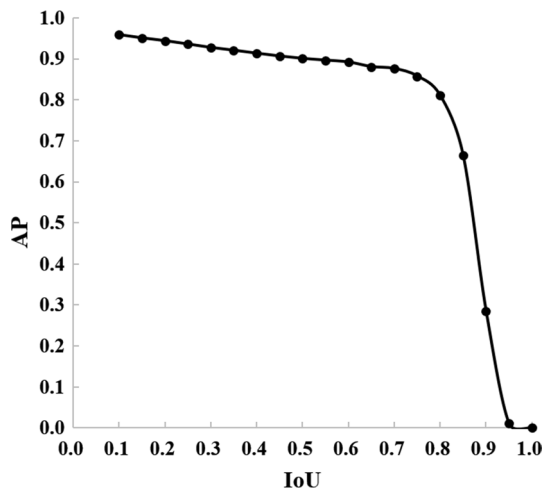


Fig. 6 Precision-Recall (P-R) curves of the YOLOv3-tiny and DY3TNet models which were evaluated on kiwifruit images with/without flash respectively (Color figure online)

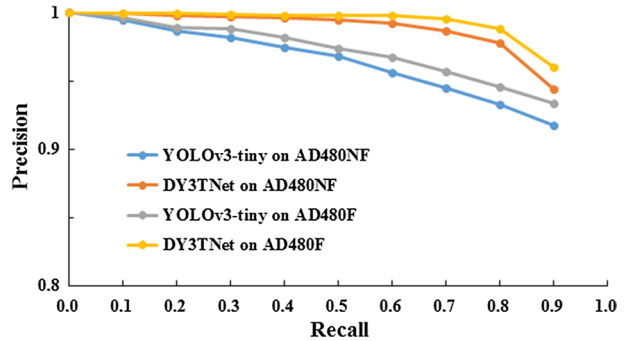


Table 4 Results of the YOLOv3-tiny and DY3TNet models trained on the AD10080F and AD10080NF datasets respectively and then tested on the AD480F and AD480NF datasets respectively

Networks	Training datasets	Testing datasets	AP	Detection time (ms/per image)
YOLOv3-tiny	AD10080NF	AD480NF	0.8778	31 ± 0.5 ^a
	AD10080F	AD480F	0.8849	31 ± 0.5 ^a
DY3TNet	AD10080NF	AD480NF	0.8988	34 ± 0.4 ^b
	AD10080F	AD480F	0.9032	34 ± 0.4 ^b

Same letter in the fifth column represent no significant difference at the 0.05 level.

were higher than the AD480NF datasets (images without flash) on both deep learning models. As shown in Table 4, the DY3TNet model obtained higher AP than the YOLOv3-tiny model on both datasets. Both models showed higher AP on images with flash than that without. The DY3TNet model achieved the highest AP of 0.9032 on the images with flash, which was 1.83% higher than the YOLOv3-tiny model on the same dataset. It can be concluded that the flash is promising on kiwifruit image detection, but statistical significance tests are needed for a further conclusion. The flash would reduce ambient light effects through the canopy gaps and highlight the calyx of the fruits, as also reported by Scarfe (2012) and Fu et al. (2018c).

DY3TNet model on different image datasets

The DY3TNet model trained and tested on different images datasets with different number of images and illumination conditions are shown in Table 5. Same as the results in Table 4, the flash may help the image datasets as the APs were higher than those of image datasets without flash in the morning and afternoon. Taking the raw training datasets of A240NF and A240F in the afternoon as an example, the DY3TNet model trained on the image dataset with flash A240F and tested on image dataset with flash A160F showed a higher AP of 0.8973 than that trained on the image dataset without flash A240NF and tested on image dataset without flash A160NF 0.8971.

When the images with and without flash were combined for training and testing, the lowest APs were found for all times. Taking the raw training datasets M240NF and M240F

in the morning as an example, the DY3TNet model was trained on the combined dataset M240NF & M240F and tested on the combined testing dataset M160NF & M160F. The AP was 0.8957, which is lower than that of the DY3TNet model trained and tested on the image datasets with and without flash separately. Also, in the all-day raw image datasets of AD720NF & AD720F, the same results were obtained. It can be said that a simple and consistent illumination condition is positive for kiwifruit detection in the orchard.

In terms of data augmentation, all the augmented training datasets showed the same trend of higher AP than their corresponding raw training datasets when tested on the same datasets. Taking the augmented training dataset N3360NF in the night as an example, the AP of the DY3TNet model improved from 0.9038 to 0.9050 when trained on the augmented dataset N3360NF and raw dataset N240NF and tested on the same testing dataset N160NF. This showed that more image data could improve object detection (Zhang et al. 2017a; Sun et al. 2017). Xue et al. (2018), Al-masni et al. (2018) and Roy et al. (2018) also reached the same conclusion that image augmentations method can further improve detection accuracy or increase sensitivity.

The image augmentation process could slightly improve the detection performance for the DY3TNet model. The biggest improvement happened on the combined augmented training datasets A3360NF & A3360F with an AP of 0.9027 and corresponding raw mixed

Table 5 Detection results of the DY3TNet model training on different images datasets with different size and illumination conditions

	Training datasets	Testing datasets	AP
Morning	M240NF	M160NF	0.8972
	M240F	M160F	0.8973
	M240NF & M240F	M160NF & M160F	0.8957
	M3360NF	M160NF	0.8992
	M3360F	M160F	0.8998
	M3360NF & M3360F	M160NF & M160F	0.8971
Afternoon	A240NF	A160NF	0.8971
	A240F	A160F	0.8973
	A240NF & A240F	A160NF & A160F	0.8957
	A3360NF	A160NF	0.9028
	A3360F	A160F	0.9030
	A3360NF & A3360F	A160NF & A160F	0.9027
Night	N240NF	N160NF	0.9038
	N240F	N160F	0.9028
	N240NF & N240F	N160NF & N160F	0.9034
	N3360NF	N160NF	0.9050
	N3360F	N160F	0.9040
	N3360NF & N3360F	N160NF & N160F	0.9035
All-day	AD720NF	AD480NF	0.8950
	AD720F	AD480F	0.8968
	AD720NF & AD720F	AD480NF & AD480F	0.8947
	AD10080NF	AD480NF	0.8988
	AD10080F	AD480F	0.9032
	AD10080NF & AD10080F	AD480NF & AD480F	0.9005

training datasets A240NF & A240F of an AP of 0.8957 when they were tested on the same mixed testing dataset A160NF & A160F. On the other hand, the DY3TNet model showed a detection performance with the lowest AP obtained of 0.8971 (A240NF) when it was trained on the raw datasets that have several images as small as 240. Therefore, the DY3TNet model that was based on the YOLO networks could achieve acceptable performance with few training samples. This was also reported by other researchers such as Xue et al. (2018), who obtained a precision rate of 0.9702 for on-tree mango detection when the YOLOv2 model was trained by 660 images. Also, Tian et al. (2019) achieved a F1 score of 0.8170 through augmented the 480 images to 4800 images by motion blur transformation, image rotation, brightness transformation, color balance.

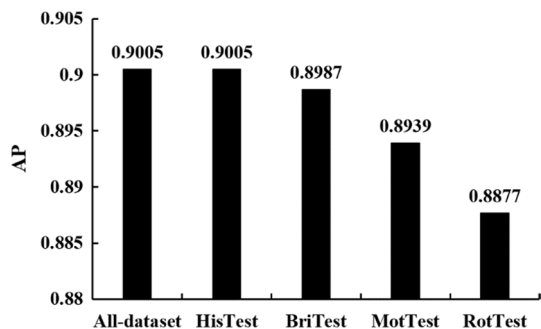
The highest AP of 0.9050 was shown in the images captured at night with the LED illumination for N3360NF. All the datasets of images captured at night obtained higher APs than their corresponding datasets in the morning and afternoon. The reason is that the images captured at night do not suffer from variable ambient light while the artificial light can provide constant illumination. Same conclusions were also reported by Scarfe (2012) and Fu et al. (2015) on kiwifruit detection and Linker (2018) on apple counting.

The results trained and tested on the different datasets generated by different augmented methods using DY3TNet model are shown in Fig. 7. The AP of RotTest was 1.28% lower than the AP of the All-dataset, which has the greatest influence on the detection results. MotTest reduced AP by 0.66% and BriTest reduced AP by 0.18%. Although the effect of motion blur and brightness transformation on the detection result was smaller than that of rotation transformation, the effect of motion blur transformation and brightness transformation cannot be ignored. However, HisTest had basically no effect on the detection result, and its AP was still about 90.05%. Therefore, the augmented method with histogram equalization should not be adopted in future studies. The detection results of these tests provide a basis for future related studies to select the appropriate augmented methods.

Conclusion

According to the characteristics of kiwifruit images in the orchard, two convolutional kernels of 3×3 and 1×1 were respectively added to the fifth and sixth convolution layers of the YOLOv3-tiny model to develop a deep YOLOv3-tiny network (DY3TNet). It took several 1×1 convolutional layers in the intermediate layers of the DY3TNet to reduce the computational complexity. Field images were captured at different time and illumination conditions and augmented to test the proposed DY3TNet, which was compared to the

Fig. 7 Testing results of the DY3TNet model training on different images datasets with different augmented methods



other four deep-learning models (Faster R-CNN with ZFNet, Faster R-CNN with VGG16, YOLOv2 and YOLOv3-tiny). The AP, detection speed and model size (weights) were used to evaluate the performance of these models on detecting kiwifruit in the orchard. For the same training and testing datasets, the AP (0.9005) of the DY3TNet model was the highest, and it maintained a short detection time (34 ms per image).

Moreover, the weight of the DY3TNet model was the smallest with only 27 MB. The results illustrated that the DY3TNet model had better performance than the other deep learning models. Therefore, the DY3TNet model can provide reliable support for field working requirements in the practical application of a multi-arm kiwifruit picking robot. The DY3TNet model and the YOLOv3-tiny model showed better performance on images with flash than that without flash. It can be concluded that the flash is positive on kiwifruit image detection. Besides, the experiments indicated that the image augmentation process could improve the detection performance for the DY3TNet model, and a simple and consistent illumination condition can improve the success rate of detection in the orchard. Overall, the results demonstrated that the DY3TNet model is sensitive to light variance. It runs fast and is promising for detecting multi-cluster kiwifruit in all-day field conditions.

Acknowledgments The authors would like to express their deep gratitude to the “Young Faculty Study Abroad Program” of the Northwest A&F University Scholarship who sponsored Dr. Longsheng Fu in conducting post-doctoral research at the Centre for Precision and Automated Agricultural Systems, Washington State University; and to the Mexian Kiwifruit Experimentation Station of Northwest A&F University for providing the experimental orchard.

Funding This work was supported by the Key Research and Development Program in Shaanxi Province of China (Grant Number 2018TSCXL-NY-05-04, 2019ZDLNY02-04); Fund of the Beijing Key Laboratory of Big Data Technology for Food Safety, Beijing Technology & Business University (Grant Number BTBD-2019KF03); the International Scientific and Technological Cooperation Foundation of Northwest A&F University (Grant Number A213021803).

References

- Akcaay, S., Kundegorski, M. E., Willcocks, C. G., & Breckon, T. P. (2018). Using deep convolutional neural network architectures for object classification and detection within x-ray baggage security imagery. *IEEE Transactions on Information Forensics and Security*, 13(9), 2203–2215. <https://doi.org/10.1109/TIFS.2018.2812196>.
- Al-masni, M. A., Al-antari, M. A., Park, J. M., Gi, G., Kim, T. Y., Rivera, P., et al. (2018). Simultaneous detection and classification of breast masses indigital mammograms via a deep learning YOLO-based CAD system. *Computer Methods and Programs in Biomedicine*, 157, 85–94. <https://doi.org/10.1016/j.cmpb.2018.01.017>.
- Bargoti, S., & Underwood, J. (2017). Image segmentation for fruit detection and yield estimation in apple orchards. *Journal of Field Robotics*, 34, 1039–1060. <https://doi.org/10.1002/rob.21699>.
- Fu, L., Feng, Y., Elkamil, T., Liu, Z., Li, R., & Cui, Y. (2018a). Image recognition method of multi-cluster kiwifruit in field based on convolutional neural networks. *Transactions of the Chinese Society of Agricultural Engineering*, 34(2), 205–211. <https://doi.org/10.11975/j.issn.1002-6819.2018.02.028> (in Chinese with English abstract).
- Fu, L., Feng, Y., Majeed, Y., Zhang, X., Zhang, J., Karkee, M., et al. (2018b). Kiwifruit detection in field images using faster R-CNN with ZFNet. *IFAC-PapersOnLine*, 51(17), 45–50. <https://doi.org/10.1016/j.ifacol.2018.08.059>.
- Fu, L., Liu, Z., Majeed, Y., & Cui, Y. (2018c). Kiwifruit yield estimation using image processing by an android mobile phone. *IFAC-PapersOnLine*, 51(17), 185–190. <https://doi.org/10.1016/j.ifacol.2018.08.137>.
- Fu, L., Majeed, Y., Zhang, X., Karkee, M., & Zhang, Q. (2020). Faster R-CNN-based apple detection in dense-foliage fruiting-wall trees using RGB and depth features for robotic harvesting. *Biosystems Engineering*, 197, 245–256. <https://doi.org/10.1016/j.biosystemseng.2020.07.007>.

- Fu, L., Sun, S., Li, R., & Wang, S. (2016). Classification of kiwifruit grades based on fruit shape using a single camera. *Sensors*, *16*(7), 1012. <https://doi.org/10.3390/s16071012>.
- Fu, L., Sun, S., Vázquezarellano, M., Li, S., Li, R., & Cui, Y. (2017). Kiwifruit recognition method at night based on fruit calyx image. *Transactions of the Chinese Society of Agricultural Engineering*, *33*(2), 199–204. <https://doi.org/10.11975/j.issn.1002-6819.2017.02.027> (in Chinese with English abstract).
- Fu, L., Tola, E., Al-Mallahi, A., Li, R., & Cui, Y. (2019). A novel image processing algorithm to separate linearly clustered kiwifruits. *Biosystems Engineering*, *183*, 184–195. <https://doi.org/10.1016/j.biosystemseng.2019.04.024>.
- Fu, L., Wang, B., Cui, Y., Su, S., Gejima, Y., & Kobayashi, T. (2015). Kiwifruit recognition at nighttime using artificial lighting based on machine vision. *International Journal of Agricultural and Biological Engineering*, *8*(4), 52–59. <https://doi.org/10.3965/j.ijabe.20150804.1576>.
- Gao, F., Fu, L., Zhang, X., Majeed, Y., Li, R., Karkee, M., et al. (2020). Multi-class fruit-on-plant detection for apple in SNAP system using Faster R-CNN. *Computers and Electronics in Agriculture*, *176*, 105634. <https://doi.org/10.1016/j.compag.2020.105634>.
- Gené-Mola, J., Sanz-Cortella, R., Rosell-Polo, J. R., Morros, J.-R., Ruiz-Hidalgo, J., Vilaplana, V., et al. (2020). Fruit detection and 3D location using instance segmentation neural networks and structure-from-motion photogrammetry. *Computers and Electronics in Agriculture*, *30*, 105591. <https://doi.org/10.1016/j.compag.2019.105165>.
- He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, 27-30 June 2016
- Hu, F., Shi, L., Li, R., Li, S., Li, X., Wang, X., et al. (2017). Fertilization evaluation of kiwifruit in Guanzhong region of Shaanxi province. *Soils and Fertilizers Sciences in China*, *54*(3), 44–49. <https://doi.org/10.11838/sfsc.20170308> (in Chinese with English abstract)
- Huang, R., Pedoeem, J., Chen, C. (2018). YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers. <https://arxiv.org/pdf/1811.05588v1.pdf>. Accessed 30 August 2020.
- Jia, W., Tian, Y., Luo, R., Zhang, Z., Lian, J., & Zheng, Y. (2020). Detection and segmentation of overlapped fruits based on optimized mask R-CNN application in apple harvesting robot. *Computers and Electronics in Agriculture*, *172*, 105380. <https://doi.org/10.1016/j.compag.2020.105380>.
- Koirala, A., Walsh, K. B., Wang, Z., & McCarthy, C. (2019). Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of ‘MangoYOLO’. *Precision Agriculture*, *20*(6), 1107–1135. <https://doi.org/10.1007/s11119-019-09642-0>.
- Lin, G., Tang, Y., Zou, X., Xiong, J., & Fang, Y. (2020). Color-, depth-, and shape-based 3D fruit detection. *Precision Agriculture*, *21*(1), 1–17. <https://doi.org/10.1007/s11119-019-09654-w>.
- Lin, M., Chen, Q., Yan, S. (2013). Network in network. <https://arxiv.org/pdf/1312.4400.pdf>. Accessed 30 August 2020.
- Linker, R. (2018). Machine learning based analysis of night-time images for yield prediction in apple orchard. *Biosystems Engineering*, *167*, 114–125. <https://doi.org/10.1016/j.biosystemseng.2018.01.003>.
- Liu, G., Nouaze, J. C., Mbouembe, P. L. T., & Kim, J. H. (2020). YOLO-tomato: a robust algorithm for tomato detection based on YOLOv3. *Sensors*, *20*(7), 2145. <https://doi.org/10.3390/s20072145>.
- Liu, W., Anguelov, D., Erhan, D., Szegegy, D., Reed, S., Fu, C., et al. (2016). SSD: single shot multi-box detector. In B. Leibe, et al. (Eds.), *Lecture notes in computer science* (pp. 21–37). Switzerland: Springer International Publishing.
- Liu, X., Jia, W., Ruan, C., Zhao, D., Gu, Y., & Chen, W. (2018). The recognition of apple fruits in plastic bags based on block classification. *Precision Agriculture*, *19*(4), 735–749. <https://doi.org/10.1007/s11119-017-9553-2>.
- Lu, Y., Chen, Z., Kang, T., Zhang, X., Bellarby, J., & Zhou, J. (2016). Land-use changes from arable crop to kiwi-orchard increased nutrient surpluses and accumulation in soils. *Agriculture Ecosystems and Environment*, *223*, 270–277. <https://doi.org/10.1016/j.agee.2016.03.019>.
- Mu, L., Liu, H., Cui, Y., Fu, L., & Gejima, Y. (2018). Mechanized technologies for scaffolding cultivation in the kiwifruit industry: a review. *Information Processing in Agriculture*, *5*(4), 401–410. <https://doi.org/10.1016/j.inpa.2018.07.005>.
- Peng, H., Huang, B., Shao, Y., Li, Z., Zhang, C., Chen, Y., et al. (2018). General improved SSD model for picking object recognition of multiple fruits in natural environment. *Transactions of the Chinese Society of Agricultural Engineering*, *34*(6), 155–162. <https://doi.org/10.11975/j.issn.1002-6819.2018.16.020> (in Chinese with English abstract)
- Pizer, S. M., Amburn, E. P., Austin, J. D., Cromartie, R., Geselowitz, A., Greer, T., et al. (1987). Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing*, *39*(3), 355–368. [https://doi.org/10.1016/S0734-189X\(87\)80186-X](https://doi.org/10.1016/S0734-189X(87)80186-X).

- Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, 27–30 June 2016
- Redmon, J., Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA, 21–26 July 2017
- Redmon, J., Farhadi, A. (2018). YOLOv3: An incremental improvement. <https://arxiv.org/pdf/1804.02767.pdf>. Accessed 30 August 2020.
- Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149. <https://doi.org/10.1109/TPAMI.2016.2577031>.
- Roy, S., Haque, A., Neubert, J. (2018). Automatic diagnosis of melanoma from dermoscopic image using real-time object detection. In Annual Conference on Information Sciences and Systems, Princeton, USA, 21–23 March 2018
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3), 211–252. <https://doi.org/10.1007/s11263-015-0816-y>.
- Sa, I., Ge, Z., Dayoub, F., Uppcroft, B., Perez, T., & McCool, C. (2016). Deepfruits: a fruit detection system using deep neural networks. *Sensors*, 16(8), 1222. <https://doi.org/10.3390/s16081222>.
- Scarfe, A. J. (2012). Development of an autonomous kiwifruit harvester, PhD thesis. Manawatu, New Zealand: Massey University.
- Shinde, S., Kothari, A., & Gupta, V. (2018). YOLO based human action recognition and localization. *Procedia Computer Science*, 133, 831–835. <https://doi.org/10.1016/j.procs.2018.07.112>.
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 1–48. <https://doi.org/10.1186/s40537-019-0197-0>.
- Silwal, A., Davidson, J. R., Karkee, M., Mo, C. K., Zhang, Q., & Lewis, K. (2017). Design, integration, and field evaluation of a robotic apple harvester. *Journal of Field Robotics*, 34, 1140–1159. <https://doi.org/10.1002/rob.21715>.
- Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. *Advances in Neural Information Processing Systems*, 1, 568–576. <https://doi.org/10.1002/14651858.CD001941.pub3>.
- Smith, A. R. (1978). Color Gamut transform pairs. *Computer Graphics (ACM)*, 12(3), 12–19. <https://doi.org/10.1145/965139.807361>.
- Sun, C., Shrivastava, A., Singh, S., Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 Oct 2017
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). Going deeper with convolutions. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Boston, USA, 7–12 June 2015
- Tani, J., Mishra, S., & Wen, J. T. (2016). Motion blur-based state estimation. *IEEE Transactions on Control Systems Technology*, 24(3), 1012–1019. <https://doi.org/10.1109/TCST.2015.2473004>.
- Taylor, L., Nitschke, G. (2018). Improving deep learning with generic data augmentation. <https://arxiv.org/pdf/1708.06020.pdf>. Accessed 30 August 2020.
- Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., & Liang, Z. (2019). Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Computers and Electronics in Agriculture*, 157, 417–426. <https://doi.org/10.1016/j.compag.2019.01.012>.
- UN Food & Agriculture Organization (2018). Production of kiwi (fruit) by countries. <https://www.fao.org/faostat/en/#data>. Accessed 30 August 2020.
- Vasconez, J. P., Delpiano, J., Vougioukas, S., & Auat Cheein, F. (2020). Comparison of convolutional neural networks in fruit detection and counting: a comprehensive evaluation. *Computers and Electronics in Agriculture*, 173, 105348. <https://doi.org/10.1016/j.compag.2020.105348>.
- Wang, C., Lee, W. S., Zou, X., Choi, D., Gan, H., & Diamond, J. (2018). Detection and counting of immature green citrus fruit based on the Local Binary Patterns (LBP) feature using illumination-normalized images. *Precision Agriculture*, 19(6), 1062–1083. <https://doi.org/10.1007/s11119-018-9574-5>.
- Wang, R. (2017). Kiwifruit recognition in natural scene. Master thesis, Northwest Agriculture and Forestry University: Shanxi, China (**in Chinese with English abstract**)
- Williams, H., Jones, M., Nejati, M., Seabright, M., Bell, J., Penhall, N., et al. (2019). Robotic kiwifruit harvesting using machine vision, convolutional neural networks, and robotic arms. *Biosystems Engineering*, 181, 140–156. <https://doi.org/10.1016/j.biosystemseng.2019.03.007>.

- Williams, H., Ting, C., Nejati, M., Jones, M., Penhall, N., Lim, J. Y., et al. (2020). Improvements to and large-scale evaluation of a robotic kiwifruit harvester. *Journal of Field Robotics*, 37(2), 187–201. <https://doi.org/10.1002/rob.21890>.
- Xue, Y., Huang, N., Tu, S., Mao, L., Yang, A., Zhu, X., et al. (2018). Immature mango detection based on improved YOLOv2. *Transactions of the Chinese Society of Agricultural Engineering*, 34(7), 173–179. <https://doi.org/10.11975/j.issn.1002-6819.2018.07.022>.
- Yang, S., Zhang, J., Bo, C., Wang, M., & Chen, L. (2019). Fast vehicle logo detection in complex scenes. *Optics and Laser Technology*, 110, 196–201. <https://doi.org/10.1016/j.optlastec.2018.08.007>.
- Yu, Y., Zhang, K., Yang, L., & Zhang, D. (2019). Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. *Computers and Electronics in Agriculture*, 163, 104846. <https://doi.org/10.1016/j.compag.2019.06.001>.
- Zhang, J., Huang, M., Jin, X., & Li, X. (2017a). A real-time Chinese traffic sign detection algorithm based on modified YOLOv2. *Algorithms*, 10(4), 1–13. <https://doi.org/10.3390/a10040127>.
- Zhang, S., Benenson, R., Schiele, B. (2017a). CityPersons: A diverse dataset for pedestrian detection, In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA, 21–26 July 2017
- Zhou, Y., Xu, T., Deng, H., & Miao, T. (2018). Real-time recognition of main organs in tomato based on channel wise group convolutional network. *Transactions of the Chinese Society of Agricultural Engineering*, 34(10), 153–162. <https://doi.org/10.11975/j.issn.1002-6819.2018.10.019> (in Chinese with English abstract)
- Zhuang, J., Luo, S., Hou, C., Tang, Y., He, Y., & Xue, X. (2018). Detection of orchard citrus fruits using a monocular machine vision-based method for automatic fruit picking applications. *Computers and Electronics in Agriculture*, 152, 64–73. <https://doi.org/10.1016/j.compag.2018.07.004>.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Longsheng Fu^{1,3,4,5}  · Yali Feng^{1,6} · Jingzhu Wu² · Zhihao Liu¹ · Fangfang Gao¹ ·
Yaqoob Majeed⁵ · Ahmad Al-Mallahi⁷ · Qin Zhang⁵ · Rui Li¹ · Yongjie Cui¹

¹ College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling 712100, Shaanxi, China

² Beijing Key Laboratory of Big Data Technology for Food Safety, Beijing Technology and Business University, Beijing 100048, China

³ Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture and Rural Affairs, Yangling 712100, Shaanxi, China

⁴ Shaanxi Key Laboratory of Agricultural Information Perception and Intelligent Service, Yangling 712100, Shaanxi, China

⁵ Center for Precision and Automated Agricultural Systems, Washington State University, Prosser, WA 99350, USA

⁶ College of Engineering, Shanxi Agricultural University, Jinzhong 030801, Shaanxi, China

⁷ Faculty of Agriculture, Department of Engineering, Dalhousie University, Truro, NS B2N 5E3, Canada