SPECIAL ISSUE | RESEARCH PAPER
https://doi.org/10.1071/CP21710

CROP & PASTURE SCIENCE

Insect detection from imagery using YOLOv3-based adaptive feature fusion convolution network

Abderraouf Amrani[A,B], Ferdous Sohel[A,B,*], Dean Diepeveen[B,C], David Murray[A] and Michael G. K. Jones[B]

For full list of author affiliations and declarations see end of paper

***Correspondence to:**
Ferdous Sohel
Information Technology, Murdoch University, Murdoch, WA 6150, Australia
Email: F.Sohel@murdoch.edu.au

Handling Editor:
Davide Cammaran

ABSTRACT

Context. Insects are a major threat to crop production. They can infect, damage, and reduce agricultural yields. Accurate and fast detection of insects will help insect control. From a computer algorithm point of view, insect detection from imagery is a tiny object detection problem. Handling detection of tiny objects in large datasets is challenging due to small resolution of the insects in an image, and other nuisances such as occlusion, noise, and lack of features. **Aims.** Our aim was to achieve a high-performance agricultural insect detector using an enhanced artificial intelligence machine learning technique. **Methods.** We used a YOLOv3 network-based framework, which is a high performing and computationally fast object detector. We further improved the original feature pyramidal network of YOLOv3 by integrating an adaptive feature fusion module. For training the network, we first applied data augmentation techniques to regularise the dataset. Then, we trained the network using the adaptive features and optimised the hyper-parameters. Finally, we tested the proposed network on a subset dataset of the multi-class insect pest dataset Pest24, which contains 25 878 images. **Key results.** We achieved an accuracy of 72.10%, which is superior to existing techniques, while achieving a fast detection rate of 63.8 images per second. **Conclusions.** We compared the results with several object detection models regarding detection accuracy and processing speed. The proposed method achieved superior performance both in terms of accuracy and computational speed. **Implications.** The proposed method demonstrates that machine learning networks can provide a foundation for developing real-time systems that can help better pest control to reduce crop damage.

Keywords: adaptive feature fusion, crop protection, deep learning, insect detection, object detection, pest management, small object detection, YOLO.

Received: 9 October 2021
Accepted: 29 April 2022
Published: 7 June 2022

Cite this:
Amrani A *et al.* (2023)
Crop & Pasture Science, **74**(6), 615–627.
doi:10.1071/CP21710

Introduction

Agriculture is critical for the world's economy and is the economic backbone of many countries. Agriculture also provides food, raw materials, and jobs to a large part of the population. However, plants and crops can suffer from various factors, e.g. chemical (Pimentel 2009), frost (Shammi *et al.* 2022), weeds (Hasan *et al.* 2021), and insects (Liu and Wang 2021). Insects are a major problem in the agricultural sector, and are one of the main biotic factors which cause agricultural losses. They can damage plants by transmitting bacterial, viral, or fungal infections (Hogenhout *et al.* 2008), and cause serious injury by eating leaves and entering fruits, roots or stems (Strauss and Zangerl 2002). Crop health experts in 67 countries undertook a recent study reported by (Savary *et al.* 2019), which demonstrated that pathogens and insects cause 10–28% of lost yield in wheat, 25–41% losses for rice, 20–41% losses for maize, 8–21% losses in potato, and 11–32% losses in soybean. Therefore, it is essential to control insects to minimise yield losses by accurate and fast detection with early intervention and automated control. When crop plants are infested with multiple insect species this results in greater losses to yields and poorer product quality. Developing multi-insect detection strategies has become an essential part of pest management (Dangles *et al.* 2009).

Recently, several agriculture datasets have been released publicly. The IP102 released by (Wu *et al.* 2019) is a large dataset for single target insect pest recognition. It contains 75 000 images with 102 insect pest categories. Insects in this dataset are divided into 8 sub-classes, each sub-class damaging a specific crop: rice, corn, and wheat. Agripest (Wang *et al.* 2021) provides a multi-target dataset for insect recognition and detection, it contains 49 700 images of 14 pest species damaging four types of crops: wheat, rice, corn, and rapeseed. Images were collected in real-field conditions, with extremely small-size insects and complicated backgrounds. Another large multi-target insect dataset, Pest24, was released by (Wang *et al.* 2020) containing 28 958 raw images collected using an automated insect images acquisition device, that can trap field insects in crops and take photos. This dataset consists of 38 insect categories belonging to five insect orders, i.e. Coleoptera, Homoptera, Orthoptera and Lepidoptera. Coleoptera (beetles, weevils) and Lepidoptera (moths) species are the most important insects that affect agriculture crops; they feed on flowers and foliage, attack plant roots, and ingest leaf or grain tissue. The three Australian unwanted plant pests that have the greatest potential cost and impact on crops are *Xylella fasidiosa*, Khapra beetle and exotic fruit flies (Australian Department of Agriculture, Water and the Environment 2021). Below ground parts of plants, including cereal crop, e.g. wheat, oilseed crop, e.g. soybeans, and tuberous crops, e.g. potato and radish, are often infested by Orthoptera species (Gryllotalpidae family), which can cause severe damage to host plants by entering and damaging the root systems, resulting in an increased susceptibility to water stress, so that the infested plant may eventually die.

Precise control and management of insect pests in the crop is an active research topic. Real-time monitoring of agricultural insect pests is crucial in precision agriculture. Traditionally, visual inspection and manual counting were done to acquire information on insect populations. However, these methods are labour-intensive, time-consuming, and potentially inconsistent due to the human factor. With the rapid development of machine learning and deep learning techniques, automatic detection of agricultural insects is now feasible. Recent developments in deep neural networks have allowed researchers to improve the accuracy of object detection and recognition systems. Typically, object detection consists of two main steps: first, the localisation of the target objects, and second, the classification of the objects on images. From a computer science and image point of view, insect detection from imagery can be seen as a tiny object detection problem. Detection of medium and large-size objects in images has been achieved for many applications. Several Convolutional Neural Network (CNN)-based object detection models have been proposed to handle the object detection problem. Based on the architecture of the networks, CNN-based object detectors can be separated into major categories: two-stage detectors and one-stage detectors, where the first category frames the detection as a coarse-to-fine process such as RCNN (Region-based CNN, Girshick *et al.* 2014), Faster RCNN (Ren *et al.* 2015), and Mask RCNN (He *et al.* 2017). In contrast, the one-stage detectors frame it in one step, e.g. YOLO ('You Only Look Once', Redmon *et al.* 2016), SSD ('Single Shot Multibox Detector', Liu *et al.* 2016), and RetinaNet (Lin *et al.* 2017). CNN-based object detection models can achieve high accuracy when dealing with single-scale large and medium-sized objects. However, detecting tiny objects, such as a 15 × 15 pixel bird in an aerial image, remains challenging (Liu *et al.* 2021). Lack of features, low resolution, complex backgrounds, and limited contextual information are the main difficulties when dealing with the tiny object detection problem (Zheng *et al.* 2012). Several deep learning models have been developed for tiny object detection (Zhao *et al.* 2019). Some studies have demonstrated that combining different feature layers is important to detect small-sized objects. Others have used contextual information to increase the recognition rates of objects. Moreover, techniques to improve classification accuracy have achieved superior results, such as those addressing imbalanced class examples and insufficient training data. Occlusion is another major nuisance when performing insect detection on images. This is a common real-world scenario, and it occurs when objects come too close or overlap others. Insect images with rich texture can be detected under occlusion, thanks to the distinctive local features, such as SIFT ('Scale Invariant Feature Transform' Lowe 2004). However, the detection of objects with less texture remains very challenging. They have large uniform regions characterised by their contour structure, which is ambiguous even without occlusions. Many research papers have proposed techniques to address this issue (Plantinga and Dyer 1990; Toshev *et al.* 2010; Gao *et al.* 2011; Lai *et al.* 2011; Hinterstoisser *et al.* 2012). Although, in these cases, occlusion problems have been divided into different sub-problems such as texture-less objects, arbitrary viewpoint, and occlusions, since addressing them together is extremely hard. Another challenge when dealing with the detection of tiny objects is the lack of good positive examples (Liu *et al.* 2021). It is hard to generate a large number of small anchor boxes that fit tiny objects during network training. Anchor boxes need to be matched with ground truth boxes. Tang *et al.* (2021) introduced the YOLO pest detection network, using deep image mining and multi-feature fusion: they based their work on YOLOv4. They improved the existing feature pyramidal network (FPN) of YOLOv4 by using a cross-stage multi-feature fusion (CSFF) method, this model was evaluated on Pest24 insect images dataset and achieved a *mAP* (Mean Average Precision) of 71.6%. Li *et al.* (2021) developed an insect detection and counting model based on YOLOv3. They improved the insect detection accuracy from complicated insect images with complicated background by using CSPDarket53 as the network

backbone. They improved the detection accuracy by 3% compared to the default YOLOv3.

This paper introduces an object detection framework based on an improved YOLOv3 network to detect insects in a subset of the agricultural pest dataset, namely Pest24. First, we applied data augmentation techniques to regularise the training set and avoid overfitting. Then we integrated an adaptive feature fusion module (AFF) to reuse features of different scales of FPN (feature pyramid network), which will increase significantly the feature extraction of tiny insects by learning the spatial weight at different scales. The resulting feature maps were then applied to the YOLOv3 pipeline (Redmon and Farhadi 2018) for insect detection. Finally, we compared our method with state-of-the-art object detectors regarding accuracy and processing time. The section 'Data set and pre-processing' presents the dataset characteristics and describes the pre-processing steps. The section 'Object detection network' presents the proposed pest detection technique, and is followed by an evaluation and the results. Finally, we sum up with a conclusion and recommendations for future work.

Related work

Significant yield losses in crops are caused by Lepidoptera species including butterflies, skippers and moths (Bradshaw *et al.* 2016). Because Lepidoptera deposit many eggs, the larval feeding from plant leaves causes direct defoliation. The most common methods used to control these insects are delta traps. The different positions and orientations that these insects can display when attached to sticky traps present a challenge for developing detection and classification models (Wen *et al.* 2015). Silveira and Monteiro (2009) developed a tool that automatically detects eyespots on butterfly wings from digital images. They used a machine learning model with features based on circularity and symmetry. This model was able to detect eyespot patterns of different insect species. However, this method has limitations with small wing sizes. Wen *et al.* (2015) proposed a pose estimation-dependent method for automated identification of field moths. This method is based on a pyramidal stacked de-noising auto-encoder (IpS-DAE) deep learning model. The model combines the shape, colour, and texture features extracted for insect description. This model achieved highly accurate *mAP* (Mean Average Precision) of moth detection. However, this work does not perform the classification of the insects. Guarnieri *et al.* (2011) designed an automatic electronic trap, which was able to monitor the codling moth (*Cydia pomonella*) for remote visual inspection. Many models have been developed based on Artificial Neural Networks (ANNs) for insect pest detection and classification. ANNs are computational trained models which can detect objects in images. Kaya *et al.* (2015) presented a computer vision method for the automatic detection of butterfly species. Based on local binary patterns and ANNs, this method could identify five butterfly species from the family Papilionoideae. This model could effectively describe the main characters of butterfly images with high classification accuracy. Wang *et al.* (2012) developed an automatic insect identification system combining ANNs and a support vector machine (SVM). They identified more than 200 insect species from 64 families such as Hymenoptera, Coleoptera, Odonata, and Orthoptera. Kang *et al.* (2014) presented a novel method for butterfly identification when viewed from different angles based on BTS (branch length similarity) entropy. This system performed well for simple butterfly images. However, multi-class recognition was not performed. Kaya *et al.* (2013) presented a CNN-based model with transfer learning to classify crop field insects. This model was applied to three public datasets to detect different Lepidoptera species. The results showed an improvement in classification performance for three insect datasets. Thenmozhi and Srinivasulu Reddy (2019) used histograms of multi-scale curvature (HoMSC) and grey-level co-occurrence matrix of image blocks (GLCMoIB) to describe the shape of butterfly wings for insect classification. This method was effective in distinguishing different species of butterflies from digital images. However, images containing butterflies have simple backgrounds and insects of similar sizes. Liu *et al.* (2019) deployed an autonomous robot vehicle for pest monitoring to implement a new method for Pyralidae pest identification. They proposed a segmentation algorithm by inverse histogram mapping for the pest image segmentation, followed by a recognition approach aspired by Hu moment. Results showed high identification accuracy with acceptable computation complexity. Xia *et al.* (2018) proposed a CNN model for multi-class insect detection. They used a Region Proposal Network instead of a traditional selective search, which improved prediction accuracy. However, this method suffers from errors in target detection that affect real-time operations. Shen *et al.* (2018) proposed an optimised deep neural network based on the Faster RCNN to detect grain storage insects. The air of this work was multi-scale feature map extraction of insects under field conditions with visual noise. This method can detect insects that are overlapping and achieve a high *mAP* of 88%.

Dataset and pre-processing

Dataset characteristics

This research used a subset of the Pest24 dataset (Wang *et al.* 2020). Pest24 is a large-scale multi-class dataset consisting of 25 878 annotated images of 800 × 600 pixels. The insect images were collected in-field using an automatic pest trap and image acquisition device. Thirty-seven insect categories are in the dataset, of which 24 categories were considered for

Fig. 1. Examples of insect classes in the Pest24 insect dataset subset.

this research, as shown in Fig. 1. These included Coleoptera, Homoptera, Hemiptera, Orthoptera, Lepidoptera and 13 other families.

The Pest24 dataset has large-scale, multi-scale, multi-class image data, small objects, non-target specimens, high object similarity, and dense object distributions. Some features are shown in Fig. 2. The relative scales of insects in Pest24-subset images are generally small. The largest insect is *Gryllotalpa orientalis*, with a relative scale 0.95%. The smallest insect is *Nilaparvata lugens*, which means that the insects are considered tiny objects. The most common insect in the dataset is *Anomala corpulenta*, with 53 347 instances, and the least present is *Holotrichia oblita*, with only 108 instances.

The images were divided into 12 701 as a training set, 5077 as the validation set, and 7600 as a test set for evaluation. Because there were non-target insects in the images, not all objects are labelled in the dataset. Additionally, some non-target insects are similar to target insects, affecting detection accuracy. Eleven hundred images contained non-target insects, and 5000 images were characterised by distortion due to the shooting angle. Furthermore, occlusion and shadows were present in 600 images. Table 1 summarises the number of images and instances of each insect category in the dataset.

Data augmentation

Unlike two-stage detectors such as Faster-RCNN or Cascade-RCNN, the performance of one-stage detectors can be significantly improved by applying data augmentation

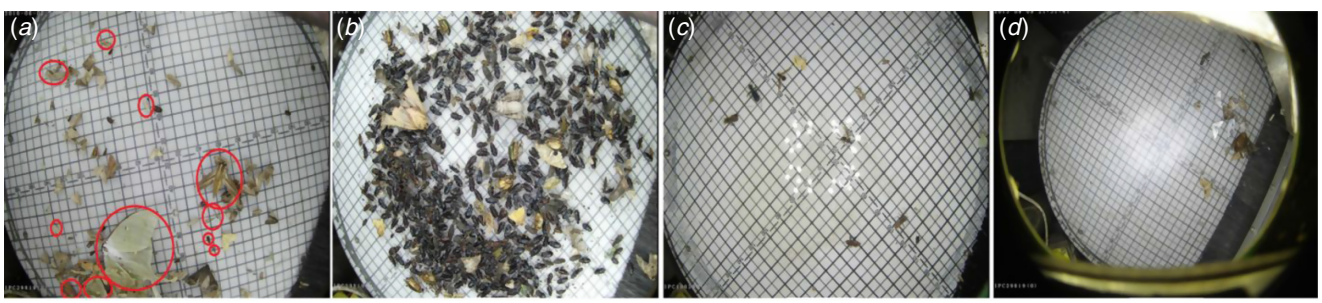

Fig. 2. Pest24 dataset characteristics: (*a*) non-target insects (red circles), (*b*) overlapping, (*c*) inflection spots caused by illumination problems, (*d*) too large non-target background.

Table 1. Number of images and instances of each pest category used from the subset Pest24 dataset.

Index	Species	Images	Instances	Index	Insect	Images	Instances
1	*Nilaparvata lugens*	316	1511	13	*Mamestra brassicae*	1707	2302
2	*Cnaphalocrocis medinalis*	944	1240	14	*Scotogramma trifolii*	3223	4679
3	*Chilo suppressalis*	454	1285	15	*Parantica menadensis*	1388	1686
4	*Spodoptera frugiperda*	3828	8880	16	*Agrotis tokionis*	369	475
5	*Helicoverpa armigera*	9049	28 014	17	*Xestia c-nigrum*	154	168
6	*Loxostege sticticalis*	5526	16 516	18	*Holotrichia oblita*	90	108
7	*Athetis lepigone*	7520	30 339	19	*Holotrichia parallela*	3111	11 675
8	*Spodoptera litura*	1588	1951	20	*Anomala corpulenta*	5228	53 347
9	*Spodoptera exigua*	3614	7263	21	*Gryllotalpa orientalis*	3629	6528
10	*Chilo polychrysus*	1357	1804	22	*Elateridae* spp.	118	167
11	*Agrotis ypsilon*	2503	4279	23	*Agriotes fuscicollis Miwa*	1814	6484
12	*Plutella xylostella*	531	953	24	*Melanotus*	239	768

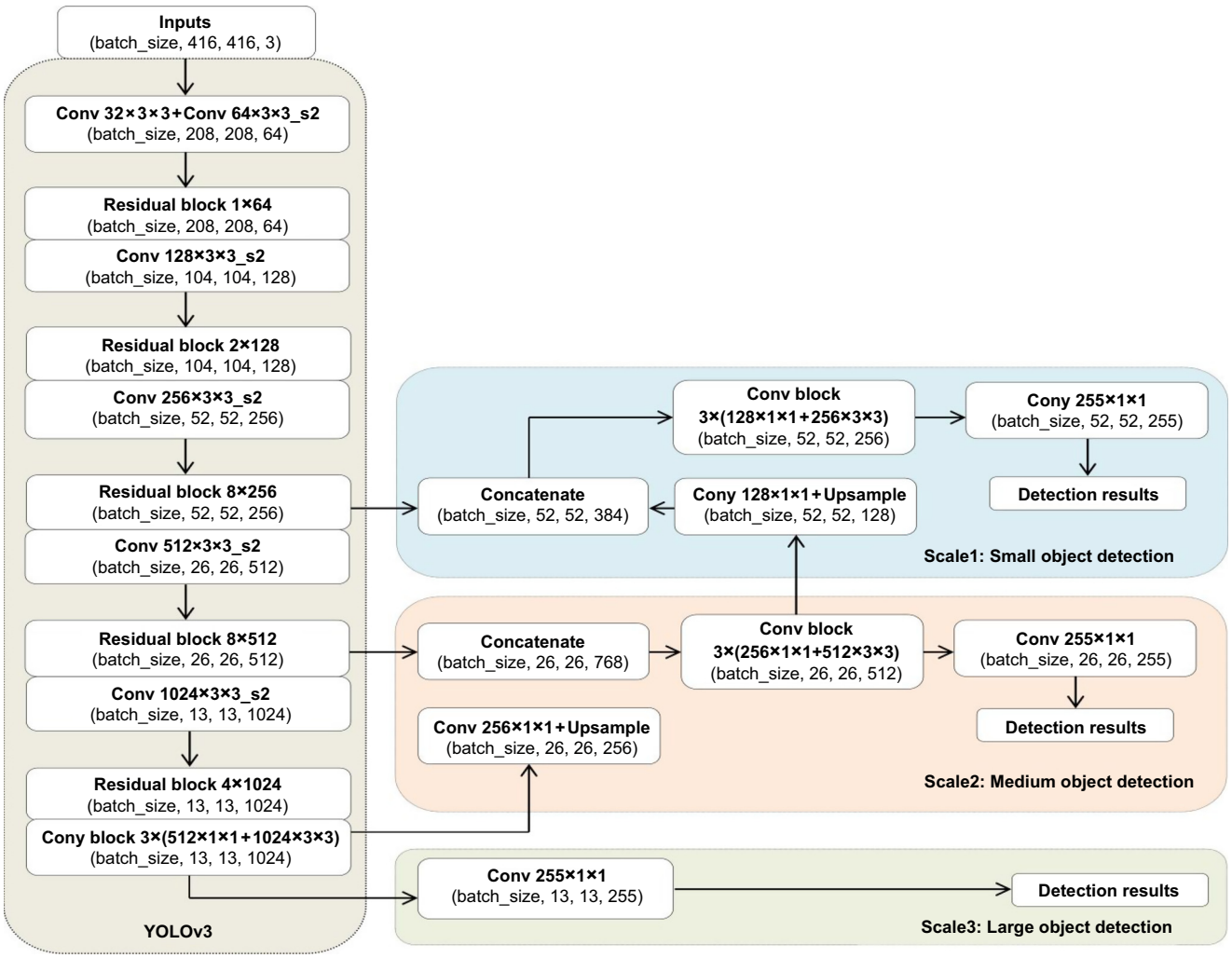

Fig. 3. YOLOv3 architecture.

techniques (Zhang *et al.* 2019). This is because the repeated cropping of Regions of Interest (ROIs) on the related feature maps generates detection outputs in two-stage detectors. Two-stage detectors substitute the process of

random crop of the input images, which means that extensive geometric augmentations are not required in this type of network. YOLOv3 is an anchor-based detector; it uses anchor boxes for object prediction. However, it is hard to generate anchor boxes that fit the insect with small-sized insects in images. Therefore, generating more data on the training set *via* data augmentation techniques can help increase the number of anchor boxes and help prevent overfitting. Our experiments applied random cropping, horizontal flip, and resized the input images to 608 × 608 pixels. After data augmentation, the number of images in the training set increased from 12 701 to 18 344 (only used to train YOLOv3 and YOLOv3-AFF).

Object detection network

YOLOv3

This work used the one-stage object detector YOLOv3, a fully connected convolutional network proposed by Redmon and Farhadi (2018). It is an efficient and simple detector in which object detection is considered as a regression problem using anchor boxes and three scales for prediction. YOLOv3 is characterised by a Darknet-53 backbone and a Feature Pyramid Network (FPN) of three scales. The structure of YOLOv3 is shown in Fig. 3. The YOLOv3 object detector uses an element-wise sum for high, medium, and low-level feature integration. FPN is a topology in which two opposite operations in the spatial dimension occur; to decrease and then expand the feature map. This step is a repeated mechanism that allows detectors to learn objects of different sizes. To perform object detection in YOLOv3, a large detection block 76 × 76 is used for detecting large objects, and a small detection block of 19 × 19 is used for small objects. However, the efficiency of this method decreases in multi-class and imbalanced datasets.

Darknet-53 feature extractor

YOLOv3 use the Darknet-53 network structure, as shown in Fig. 4. This network structure contains 53 convolution layers $(3(2+1\times2+1+2\times2+1+8\times2+1+8\times2+1+4\times2+1=53))$, and five pooling layers to overcome overfitting problems. After each convolutional layer, batch normalisation and dropout layers were added. Darknet-53 adopts the residual neural network of five residual blocks. In YOLOv3, network depth increases using residual units to avoid gradient disappearance.

YOLOv3-AFF model

In the Pest24 dataset, insects are considered tiny objects in the images. YOLOv3 adopts a multi-scale strategy and feature pyramid network to predict objects on three different scales, enabling YOLOv3 to handle small, medium, and large object detection problems. However, when multiple object sizes are present in one image, some image features can be lost during the successive resizing at each scale. As a result, the feature maps may not contain features of some objects of different sizes affecting the detection accuracy. Therefore, to improve the detection accuracy, we integrated an adaptive feature fusion technique to reuse features at different scales in the YOLOv3 network.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3/1	416 × 416
	Convolutional	64	3 × 3/2	208 × 208
	Convolutional	32	2 × 1/1	
1×	Convolutional	64	3 × 3/2	
	Residual			208 × 208
	Convolutional	128	3 × 3/2	104 × 104
	Convolutional	64	1 × 1/1	
2×	Convolutional	128	3 × 3/2	
	Residual			104 × 104
	Convolutional	256	3 × 3/2	52 × 52
	Convolutional	128	1 × 1/1	
8×	Convolutional	256	3 × 3/2	
	Residual			52 × 52
	Convolutional	512	3 × 3/2	26 × 26
	Convolutional	256	1 × 1/1	
8×	Convolutional	512	3 × 3/2	
	Residual			26 × 26
	Convolutional	1024	3 × 3/2	13 × 13
	Convolutional	512	1 × 1/1	
8×	Convolutional	1024	3 × 3/2	
	Residual			13 × 13

Fig. 4. Darknet-53 network structure.

Adaptive feature fusion (AFF)

Object detectors such as YOLOv3 and RetinaNet use FPN to perform multi-layer feature learning and element-wise sum or concatenation for multi-level feature integration. However, for FPN-based single-stage detectors, the inconsistency between different feature scales represents the main limitation. To overcome this issue, it is necessary to fully use the semantic information of different feature-levels. For that, we deployed an adaptive feature fusion model (Liu *et al.* 2019*b*). The idea of this method is as follows: modify the up and down-sampling strategies used in the original YOLOv3 to allow the detector to learn the spatial weight at different scales. This approach consists of two main steps: identical re-scaling and adaptive fusion. Fig. 5 illustrates the adaptive feature fusion module.

Re-scaling. For the YOLOv3 framework, the number of channels and the resolution of features is different at three levels $l \in (1,2,3)$. We denote x^1, the feature of resolution at each level. For up-sampling, 1 × 1 convolutional layer is applied. As a result, the number of channels at level l as

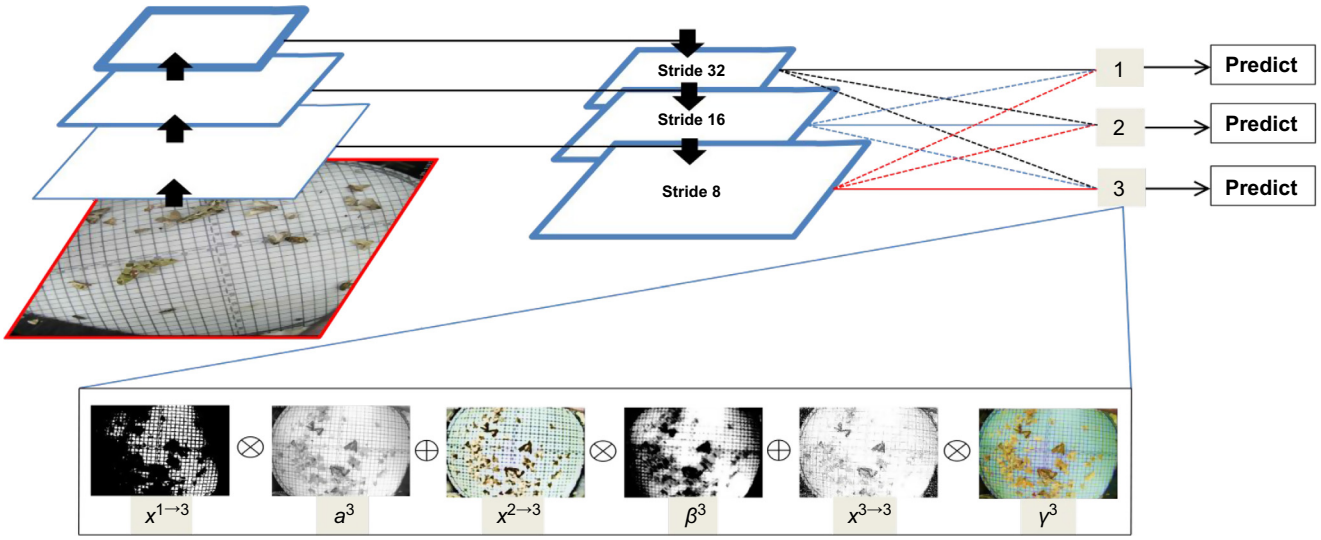

Fig. 5. Illustration of the adaptive fusion feature module integration with FPN on YOLOv3.

well as features are compressed. After that, the feature resolutions are up-scaled respectively with interpolation. The next step is applying a 3 × 3 convolution layer with a stride of two for down-sampling at a 0.5 ratio. This will simultaneously change the number of channels and the resolution of features.

Adaptive fusion. After feature resizing from level l to level n, feature fusion at each corresponding level l can be formulated as follow:

$$y_{ij}^{l} = a_{ij}^{l} \cdot x_{ij}^{1\to l} + \beta_{ij}^{l} \cdot x_{ij}^{2\to l} + \gamma_{ij}^{l} \cdot x_{ij}^{3\to l} \quad (1)$$

Here: $x_{ij}^{1\to l}$, $x_{ij}^{2\to l}$, and $x_{ij}^{3\to l}$ represent the feature maps from three layers at levels (1, 2, 3) corresponding to strides (32, 16, 8). a_{ij}^{l}, β_{ij}^{l} and γ_{ij}^{l} refer to the spatial weights calculated using the activation function.

We use the method introduced by (Wang *et al.* 2019) to force $a_{ij}^{l} + \beta_{ij}^{l} + \gamma_{ij}^{l} = 1$ and define

$$a_{ij}^{l} = \frac{e^{\lambda_{a_{ij}}^{l}}}{e^{\lambda_{a_{ij}}^{l}} + e^{\lambda_{\beta_{ij}}^{l}} + e^{\lambda_{\gamma_{ij}}^{l}}} \quad (2)$$

Where: $\lambda_{a_{ij}}^{l}, \lambda_{\beta_{ij}}^{l}, \lambda_{\gamma_{ij}}^{l}$ are the control parameters used to define the spatial weights a_{ij}^{l}, β_{ij}^{l} and γ_{ij}^{l} respectively.

After calculating feature maps y^1, y^2, and y^3, the same YOLOv3 pipeline performs object detection.

Evaluation and results

Implementation and experiment

We evaluated the insect detection accuracy on the Pest24 dataset using the YOLOv3-AFF on an Ubuntu 20.04 operating system. We used the PyTorch framework with CUDA 11.1. The model was trained on two GPUs and 120 epochs for training. We applied the cosine learning schedule from 0.001 to 0.00001. All experiments were performed on the bounding box detection track on the images. The dataset was divided into training, validation, and testing sets of: 18 344, 5077 and 7600 respectively. We kept the distribution for other detectors at 12 701, 5077, and 7600. We also evaluated insect detection performance on single-shot state-of-the-art object detectors SSD, YOLOv3, and RetinaNet; and two-stage object detectors, Faster-RCNN, Cascade-RCNN and Fast-RCNN. Table 2 summarises the configurations of the experimental environments.

Detection results and discussion

The detection accuracy on the dataset was evaluated using different object detectors YOLOv3-AFF, YOLOv3, SSD, Faster-RCNN, Cascade-RCNN, Fast-RCNN and RetinaNet. We trained each of these methods on the training sets and tested them on the test set. Also, the hyper-parameters for object detectors have been optimised for better detection accuracy as follows: varying the *base_size* of (2, 4, 8, 16) for Faster-RCNN, and the *anchor_scales* of (2, 4, 8) for

Table 2. Configurations of experimental hardware environment.

Parameters	Configuration
CPU	Intel i9-9900x
GPUs	TITAN RTX + GeForce GT
Accelerated environment	CUDA 11.1, cuDNN 8.0.5
Framework	PyTorch 1.7
Operating system	Ubuntu 20.04
Training framework	Darknet-53

Table 3. Detection performance in terms of *mAP* (%) and frames per second (fps) numbers in bold indicate the highest fps and mAP.

Method	Backbone	Batch size	fps	*mAP* (%)
SSD300	VGG	12	43.0	50.52
Faster-RCNN	ResNet-152	64	11.1	51.72
Cascade-RCNN	ResNet-101	12	9.5	59.97
RetinaNet	ResNet-50-FPN	64	6.9	63.01
Fast-RCNN	ResNet-101-FPN	64	31.3	54.68
YOLOv3	Darknet-53	32	**68.9**	61.82
YOLOv3-AFF	Darknet-53	32	63.8	**72.10**

Cascade-RCNN. For the one-stage detector SSD, we adjusted $s_{\min}$ and $s_{\max}$ parameters that define the *anchor_size* on each feature map to [(0.1–0.7), (0.1–0.8), (0.2–0.7), (0.2–0.9)]. The regulator hyper-parameter that controls two task losses for Fast-RCNN was set to =1, for RetinaNet, we used three ratios [0.006, 1.65, 1.53] with *bash_size* = 16. We used the *k*-means algorithm in YOLOv3 and YOLOv3-AFF to optimise the *scale_range*.

Evaluation metrics

We use the *mAP* (Mean Average Precision) evaluation criteria to evaluate the detection accuracy. The *mAP* can be obtained using each class's *AP* (Average Precision) as mentioned in the Eqn 6. By using the TP (True Positive), FP (False Positive), and FN (False Negative), precision and recall were calculated following Eqns 3 and 4, respectively.

$$\text{Precision}(P) = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3}$$

$$\text{Recall}(R) = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4}$$

After calculating the precision and recall, the Average Precision (AP) represents the area under the precision–recall curve and can be calculated as follows:

$$\text{AP} = \int_0^1 P(R)\text{d}R \tag{5}$$

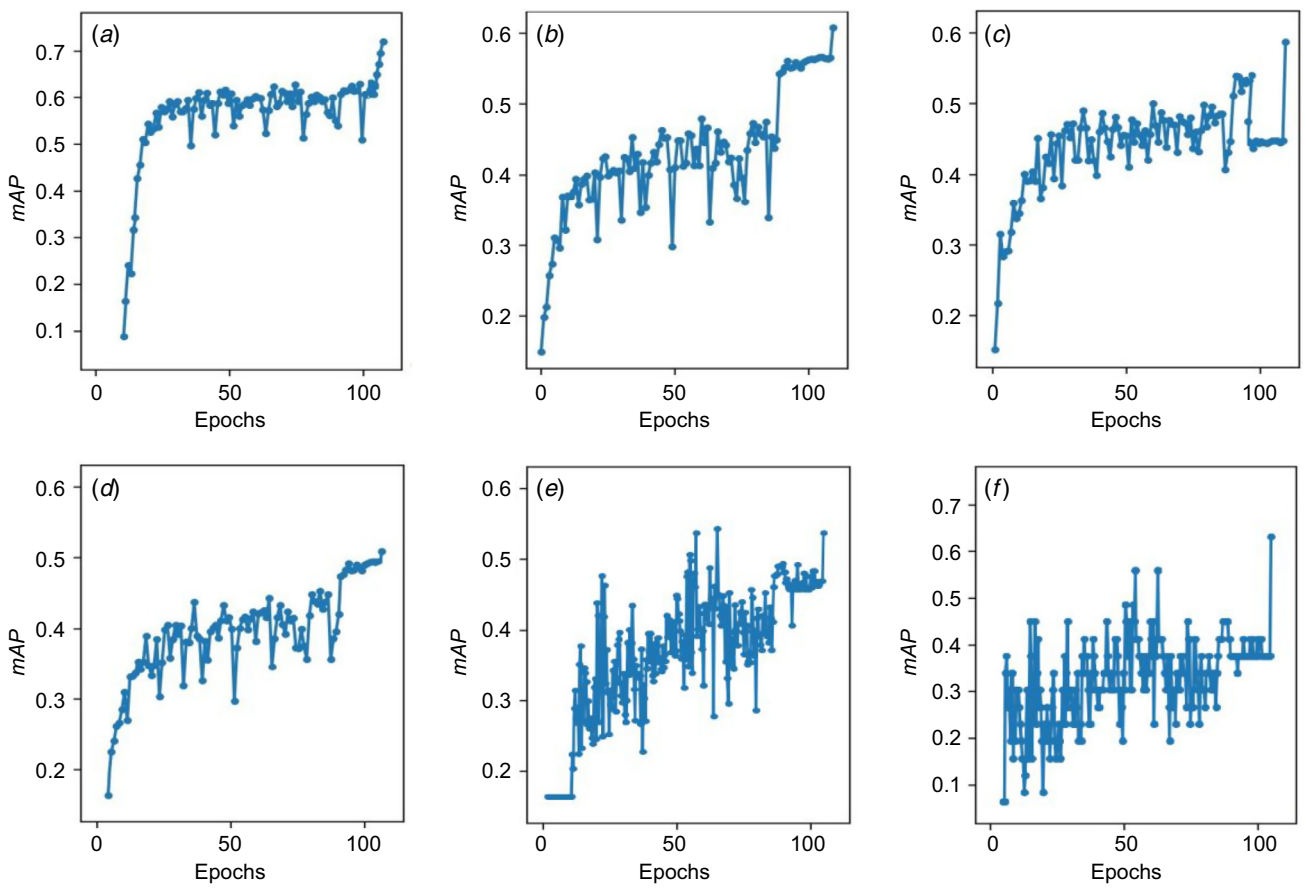

Fig. 6. Evolution of *mAP* with the number of epochs. (*a*) YOLOv3-AFF, (*b*) YOLOv3, (*c*) Cascade- RCNN, (*d*) SSD, (e) Fast-RCNN, (*f*) RetinaNet.

mAP score is then calculated by taking the AP of each class as following:

$$mAP = \frac{1}{2}\sum_{k=n}^{k=1} \mathrm{AP}_k \quad (6)$$

Where: n represents the number of classes, and AP_k represents AP of class k.

Comparison with the state-of-the-art methods

We evaluated the proposed YOLOv3-AFF model on the dataset and compared the detection accuracy and performance with other state-of-the-art methods in Table 3.

The experiment results demonstrate that our approach achieved the best detection accuracy, with *mAP* of 72% compared to YOLOv3 with 61.82%, Cascade-RCNN 59.97%, Faster-RCNN 51.72%, SSD 50.52%, Fast-RCNN 53.68%, and RetinaNet with 63.01%. The default YOLOv3 has the fastest processing speed, with 68.9 frames per second (fps). Our method has 63.8 fps, which represents a good processing speed compared to other state-of-the-art models. Fig. 6 shows the progress of *mAP* scores with the number of training epochs, comparing four methods. Fig. 6*a* shows that the *mAP* score of YOLOv3-AFF stabilised after 30 epochs with the highest *mAP* reached after 100 epochs. Fig. 6*b* (YOLOv3) shows stabilisation after 90 epochs, similar to Fig. 6*d* (SSD). In Fig. 6*c* (Cascade-RCNN), stabilisation occurred after 100 epochs, which is similar to YOLOv3-AFF (Fig. 6*a*). We have evaluated the network processing time by comparing the fps of each method. Table 3 shows computational speed results. The YOLOv3 method achieved the fastest processing time of 68.9 fps, and the YOLOv3-AFF had 63.8 fps. The processing speed of our method is slightly slower than the YOLOv3 due to the integration of the adaptive feature module. The multiple resizing of images in our method increases the number of pixels on images, and hence, the processing time becomes longer.

To demonstrate the efficiency of our model, we calculated APs for each of the 24 insect species at different relative scales. Table 4 summarises the test results. Also, we further validated YOLOv3-AFF performance by conducting insect detection experiments on the selected insects of small average scale and insects with the lowest distribution in the Pest24 dataset. We compared APs of each

Table 4. APs of 24 classes of insects by different evaluation methods. Numbers in bold indicates the highest APs.

Method	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
SSD300	0.1	38.3	49.4	63.1	78.9	66.0	46.5	59.2	34.2	56.4	73.7	0.3	35.2	30.2	33.6	55.1	3.8	51.4	85.1	93.3	94.7	42.8	66.6	53.3
Faster-RCNN	0.0	38.5	51.7	72.0	81.3	66.8	48.3	68.9	35.5	64.5	81.9	0.0	48.8	43.3	46.5	62.9	4.8	49.2	79.1	83.8	95.7	44.8	58.1	31.2
Cascade-RCNN	4.7	58.5	65.8	75.6	88.3	77.8	67.1	67.1	46.6	62.3	84.6	3.9	53.3	50.5	47.5	57.7	2.7	48.2	89.0	91.9	97.0	51.2	80.4	64.0
RetinaNet	3.1	48.9	66.3	80.7	**93.2**	76.9	72.8	**77.0**	50.2	78.1	85.3	0.8	**66.4**	45.9	48.9	**77.6**	3.9	63.9	90.7	94.2	98.3	45.9	77.7	64.0
Fast-RCNN	1.1	44.5	53.7	71.8	85.0	67.9	53.1	70.2	34.8	69.8	83.0	0.7	52.5	44.9	50.1	63.0	**4.7**	53.5	82.0	86.1	96.9	44.9	61.3	35.0
YOLOv3	0.6	51.7	72.1	82.9	91.7	80.7	68.9	76.8	52.5	75.9	88.7	1.6	60.4	51.5	50.2	74.2	1.5	61.4	93.3	**97.3**	98.6	40.0	79.7	73.6
YOLOv3-AFF	**5.9**	**76.4**	**86.1**	**85.6**	90.4	**84.9**	**90.1**	75.2	**74.1**	**81.6**	**93.7**	**5.1**	65.0	**67.9**	**61.5**	71.3	4.3	**77.6**	**94.8**	97.0	**99.1**	**69.8**	**87.2**	**77.1**

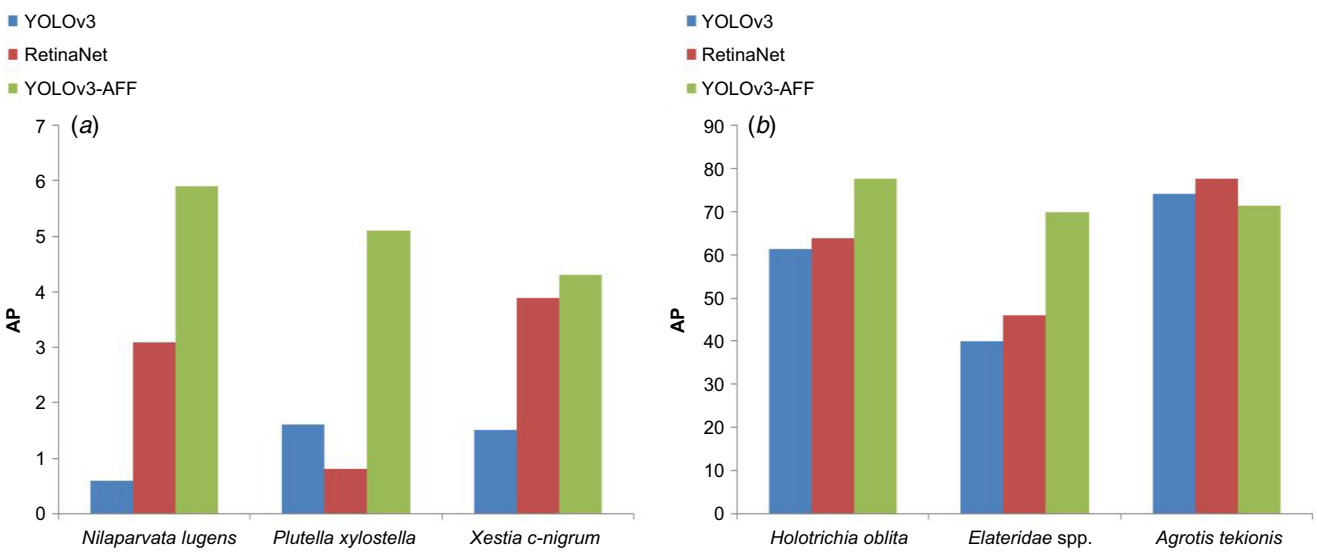

Fig. 7. APs (%) of different detection methods on insects of small average scale (*a*), and insects of low distributions on dataset images (*b*).

selected insect; Fig. 7 shows that YOLOv3-AFF achieved higher accuracy among other networks especially for insects with small scale.

Additionally, we selected several images from YOLOv3 and the proposed YOLOv3-AFF, as shown in Fig. 8. We show that the category and location of insects are correctly detected in

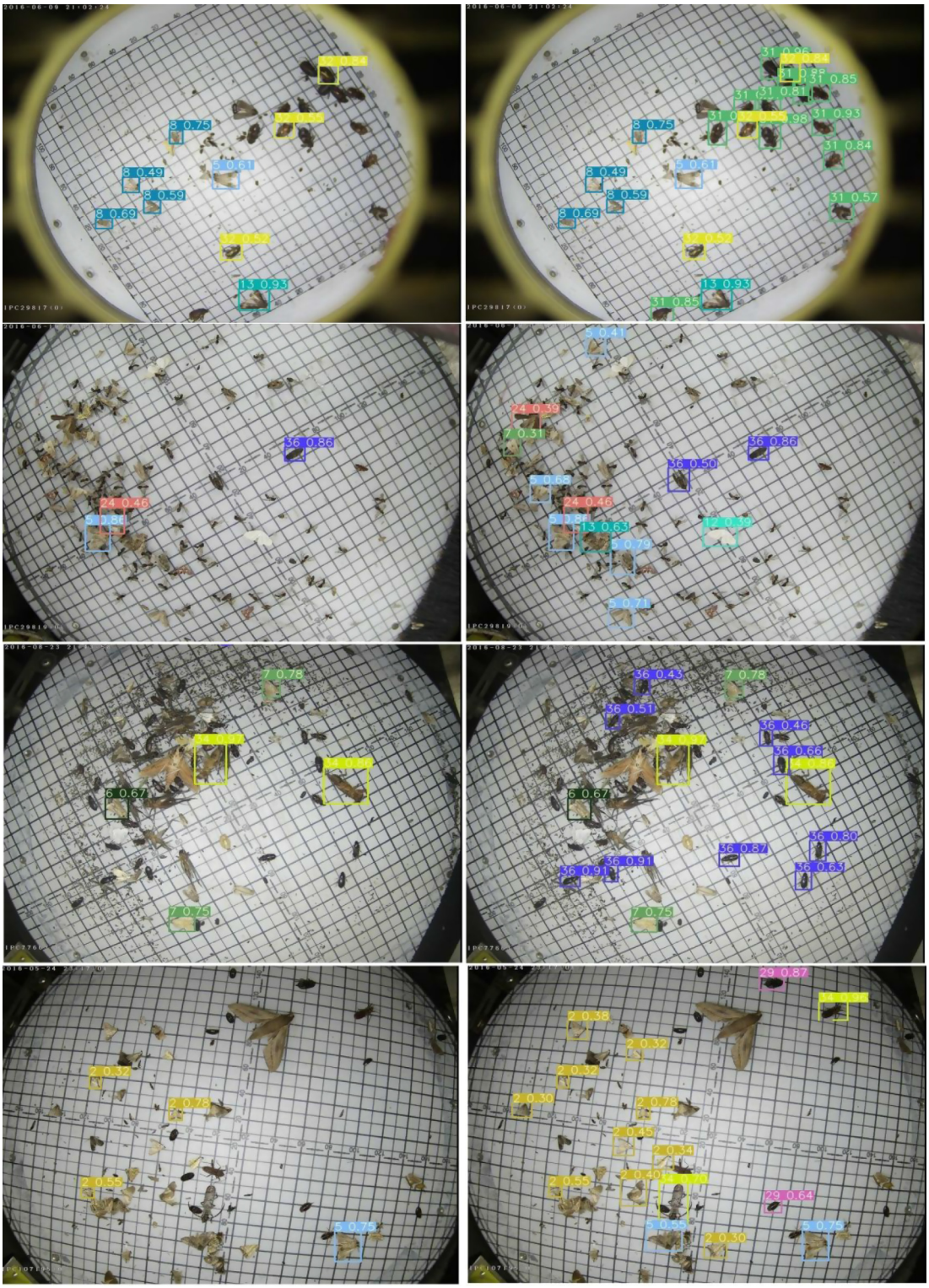

Fig. 8. Bounding box detection on images; Left: default YOLOv3, Right: YOLOv3 AFF (our method). On top of bounding boxes: insect index on the left, detection accuracy the right.

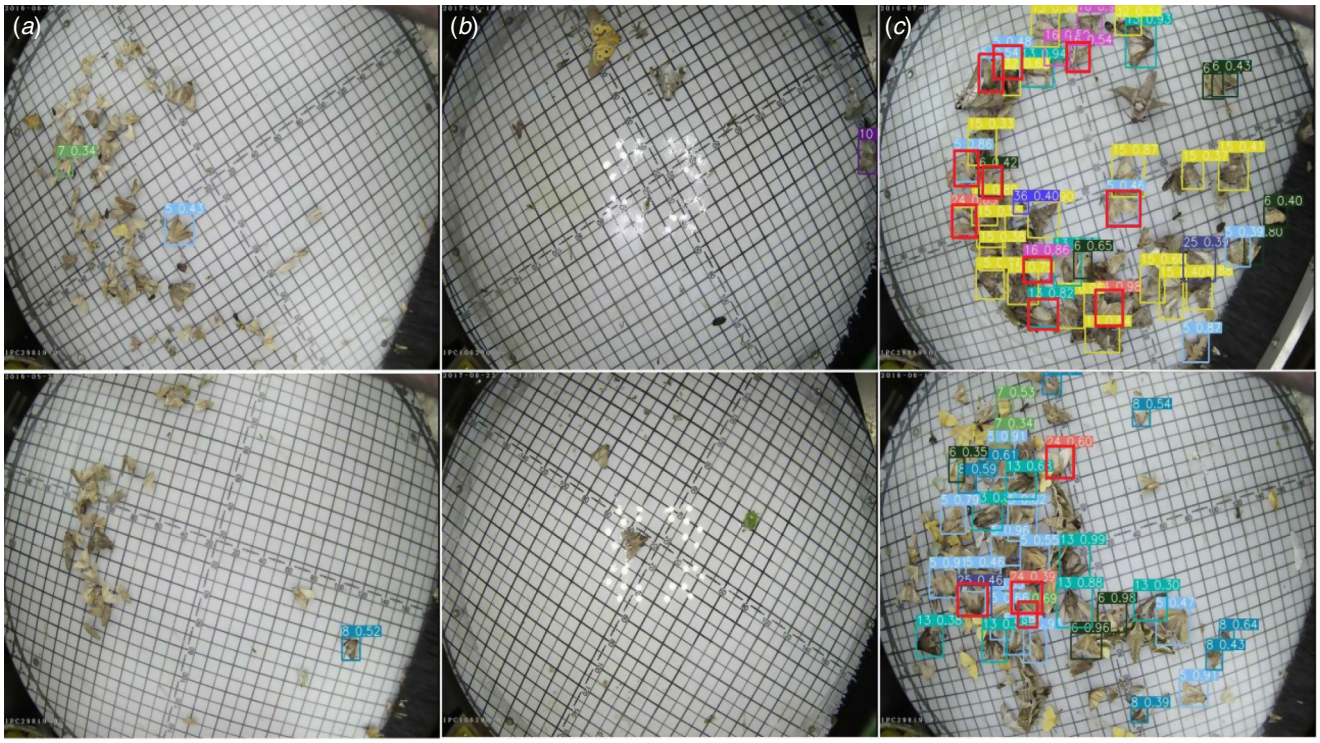

Fig. 9. Examples of wrong detection: (*a*) missing detection, (*b*) complex background and illumination, (*c*) false detection (red squares).

multi-scale images with small insects. However, as shown in Fig. 9, false and missing detection occurs in some insect images, especially with similar or overlapped insects.

Conclusion

This paper presents an improved YOLOv3 object detector to detect insects from small-object images in the Pest24 dataset. We improved the multi-scale feature detection of YOLOv3 by integrating an adaptive feature fusion module. Experimental results show a significant improvement in insect detection accuracy compared to other state-of-art object detectors: SSD, Faster-RCNN, Cascade-RCNN, RetinaNet, Fast-RCNN, and YOLOv3. Additionally, the integrated adaptive feature fusion module has minimal impact on the processing speed (fps) compared to default YOLOv3. The proposed YOLOv3-AFF demonstrates promising results for small insects and multi-class detection while optimising processing speed. Further, it is applicable for real-time automatic insect detection from images on a large multi-scale agricultural pest dataset. However, the model's performance was poorer for images containing overlapping small insects. Also, colour and shape similarity between insects caused some mis-detection, e.g. the species of *Mamestra brassicae* and *Scotogramma trifolli* Rottemberg have similar wing shapes and colour, which confuses the network during the feature extraction stage. Another challenge is the scale difference between insects on the same images, e.g. *Gryllotalpa orientalis* is 28 times larger than *Nilaparvata lugens*. This size difference increases the imbalance between instances in the dataset, which decreases the insect detection accuracy.

In future, we plan to address the detection of overlapped insects, insects with high similarity, and scale invariant. In addition, more research is required on the feature extraction strategy to increase the accuracy of detection in challenging conditions.

References

Australian Department of Agriculture, Water and the Environment (2021) Plant pests and diseases @ONLINE. Available at https://www.awe.gov.au/biosecurity-trade/pests-diseases-weeds/plant

Bradshaw CJA, Leroy B, Bellard C, Roiz D, Albert C, Fournier A, Barbet-Massin M, Salles J-M, Simard F, Courchamp F (2016) Massive yet grossly underestimated global costs of invasive insects. *Nature Communications* **7**, 12986. doi:10.1038/ncomms12986

Dangles O, Mesías V, Crespo-Perez V, Silvain J-F (2009) Crop damage increases with pest species diversity: evidence from potato tuber moths in the tropical Andes. *Journal of Applied Ecology* **46**, 1115–1121. doi:10.1111/j.1365-2664.2009.01703.x

Gao T, Packer B, Koller D (2011) A segmentation-aware object detection model with occlusion handling. In 'Proceedings of the IEEE conference on computer vision and pattern recognition'. pp. 1361–1368. (IEEE)

Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In 'Proceedings of the IEEE conference on computer vision and pattern recognition'. pp. 580–587. (IEEE)

Guarnieri A, Maini S, Molari G, Rondelli V (2011) Automatic trap for moth detection in integrated pest management. *Bulletin of Insectology* **64**, 247–251.

Hasan ASMM, Sohel F, Diepeveen D, Laga H, Jones MGK (2021) A survey of deep learning techniques for weed detection from images. *Computers and Electronics in Agriculture* **184**, 106067. doi:10.1016/j.compag.2021.106067

He K, Gkioxari G, Dollár P, Girshick R (2017) Mask R-CNN. In 'Proceedings of the IEEE international conference on computer vision'. pp. 2961–2969. (IEEE)

Hinterstoisser S, Cagniart C, Ilic S, Sturm P, Navab N, Fua P, Lepetit V (2012) Gradient response maps for real-time detection of textureless objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**, 876–888. doi:10.1109/TPAMI.2011.206

Hogenhout SA, Oshima K, Ammar E-D, Kakizawa S, Kingdom HN, Namba S (2008) Phytoplasmas: bacteria that manipulate plants and insects. *Molecular Plant Pathology* **9**, 403–423. doi:10.1111/j.1364-3703.2008.00472.x

Kang S-H, Cho J-H, Lee S-H (2014) Identification of butterfly based on their shapes when viewed from different angles using an artificial neural network. *Journal of Asia-Pacific Entomology* **17**, 143–149. doi:10.1016/j.aspen.2013.12.004

Kaya Y, Kayci L, Tekin R (2013) A computer vision system for the automatic identification of butterfly species *via* gabor-filter-based texture features and extreme learning machine: GF + ELM. *TEM Journal* **2**, 13–20.

Kaya Y, Kayci L, Uyar M (2015) Automatic identification of butterfly species based on local binary patterns and artificial neural network. *Applied Soft Computing* **28**, 132–137. doi:10.1016/j.asoc.2014.11.046

Lai K, Bo L, Ren X, Fox D (2011) A large-scale hierarchical multi-view RGB-D object dataset. In 'Proceedings of the IEEE international conference on robotics and automation'. pp. 1817–1824. (IEEE)

Li K, Zhu J, Li N (2021) Insect detection and counting based on YOLOv3 model. In 'Proceedings of the 2021 IEEE 4th international conference on electronics technology (ICET)'. pp. 1229–1233. (IEEE)

Liu J, Wang X (2021) Plant diseases and pests detection based on deep learning: a review. *Plant Methods* **17**, 22. doi:10.1186/s13007-021-00722-9

Lin T-Y, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In 'Proceedings of the IEEE international conference on computer vision'. pp. 2980–2988. (IEEE)

Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) SSD: single shot multibox detector. In 'European conference on computer vision'. pp. 21–37. (Springer)

Liu B, Hu Z, Zhao Y, Bai Y, Wang Y (2019*a*) Recognition of pyralidae insects using intelligent monitoring autonomous robot vehicle in natural farm scene. *arXiv:1903.10827*. doi:10.48550/arXiv.1903.10827

Liu S, Huang D, Wang Y (2019*b*) Learning spatial fusion for single-shot object detection. *arXiv:1911.09516*. doi:10.48550/arXiv.1911.09516

Liu Y, Sun P, Wergeles N, Shang Y (2021) A survey and performance evaluation of deep learning methods for small object detection. *Expert Systems with Applications* **172**, 114602. doi:10.1016/j.eswa.2021.114602

Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* **60**, 91–110. doi:10.1023/B:VISI.0000029664.99615.94

Pimentel D (2009) Pesticides and pest control. In 'Integrated pest management: innovation-development process'. (Eds R Peshin, AK Dhawan) pp. 83–87. (Springer)

Plantinga H, Dyer CR (1990) Visibility, occlusion, and the aspect graph. *International Journal of Computer Vision* **5**, 137–160. doi:10.1007/BF00054919

Redmon J, Farhadi A (2018) YOLOv3: an incremental improvement. *arXiv:1804.02767*. doi:10.48550/arXiv.1804.02767

Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In 'Proceedings of the IEEE conference on computer vision and pattern recognition'. pp. 779–788. (IEEE)

Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. In 'Proceedings of the advances in neural information processing systems 28'. pp. 91–99. (Curran Associates)

Savary S, Willocquet L, Pethybridge SJ, Esker P, McRoberts N, Nelson A (2019) The global burden of pathogens and pests on major food crops. *Nature Ecology & Evolution* **3**, 430–439. doi:10.1038/s41559-018-0793-y

Shammi S, Sohel F, Diepeveen D, Zander S, Jones MG (2022) A survey of image-based computational learning techniques for frost detection in plants. *Information Processing in Agriculture*. [In press] doi:10.1016/j.inpa.2022.02.003

Shen Y, Zhou H, Li J, Jian F, Jayas DS (2018) Detection of stored-grain insects using deep learning. *Computers and Electronics in Agriculture* **145**, 319–325. doi:10.1016/j.compag.2017.11.039

Silveira M, Monteiro A (2009) Automatic recognition and measurement of butterfly eyespot patterns. *Biosystems* **95**, 130–136. doi:10.1016/j.biosystems.2008.09.004

Strauss SY, Zangerl AR (2002) Plant-insect interactions in terrestrial ecosystems. In 'Plant-animal interactions: an evolutionary approach'. (Eds CM Herrera, O Pellmyr) pp. 77–106. (Blackwell Publishing)

Tang Z, Chen Z, Qi F, Zhang L, Chen S (2021) Pest-YOLO: deep image mining and multi-feature fusion for real-time agriculture pest detection. In 'Proceedings of the 2021 IEEE international conference on data mining (ICDM)'. pp. 1348–1353. (IEEE)

Thenmozhi K, Srinivasulu Reddy U (2019) Crop pest classification based on deep convolutional neural network and transfer learning. *Computers and Electronics in Agriculture* **164**, 104906. doi:10.1016/j.compag.2019.104906

Toshev A, Taskar B, Daniilidis K (2010) Object detection *via* boundary structure segmentation. In 'Proceedings of the 2010 IEEE computer society conference on computer vision and pattern recognition'. pp. 950–957. (IEEE)

Wang J, Lin C, Ji L, Liang A (2012) A new automatic identification system of insect images at the order level. *Knowledge-Based Systems* **33**, 102–110. doi:10.1016/j.knosys.2012.03.014

Wang G, Wang K, Lin L (2019) Adaptively connected neural networks. In 'Proceedings of the IEEE/CVF conference on computer vision and pattern recognition'. pp. 1781–1790. (IEEE)

Wang Q-J, Zhang S-Y, Dong S-F, Zhang G-C, Yang J, Li R, Wang H-Q (2020) Pest24: a large-scale very small object data set of agricultural pests for multi-target detection. *Computers and Electronics in Agriculture* **175**, 105585. doi:10.1016/j.compag.2020.105585

Wang R, Liu L, Xie C, Yang P, Li R, Zhou M (2021) AgriPest: a large-scale domain-specific benchmark dataset for practical agricultural pest detection in the wild. *Sensors* **21**, 1601. doi:10.3390/s21051601

Wen C, Wu D, Hu H, Pan W (2015*a*) Pose estimation-dependent identification method for field moth images using deep learning architecture. *Biosystems Engineering* **136**, 117–128. doi:10.1016/j.biosystemseng.2015.06.002

Wu X, Zhan C, Lai YK, Cheng MM, Yang J (2019) Ip102: A large-scale benchmark dataset for insect pest recognition. In 'Proceedings of the IEEE/CVF conference on computer vision and pattern recognition'. pp. 8787–8796. (IEEE)

Xia D, Chen P, Wang B, Zhang J, Xie C (2018) Insect detection and classification based on an improved convolutional neural network. *Sensors* **18**, 4169. doi:10.3390/s18124169

Zhang Z, He T, Zhang H, Zhang Z, Xie J, Li M (2019) Bag of freebies for training object detection neural networks. *arXiv:1902.04103*. doi:10.48550/arXiv.1902.04103

Zhao Z-Q, Zheng P, Xu S-T, Wu X (2019) Object detection with deep learning: a review. *IEEE Transactions on Neural Networks and Learning Systems* **30**, 3212–3232. doi:10.1109/TNNLS.2018.2876865

Zheng WS, Gong S, Xiang T (2012) Quantifying and transferring contextual information in object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**, 762–777. doi:10.1109/TPAMI.2011.164

Data availability. We thankfully acknowledge that we received the data set from the authors of Wang *et al.* (2020) , who are the original contributors of the dataset. Data related to our experimental models and analysis will be available upon request and as per the guidelines of the journal.

Conflicts of interest. The authors declare no conflicts of interest.

Declaration of funding. This work was supported by Murdoch University Digital Agriculture Connectivity PhD scholarship to Abderraouf Amrani.

Author affiliations
[A]Information Technology, Murdoch University, Murdoch, WA 6150, Australia.
[B]Centre for Crop and Food Innovation, Food Futures Institute, Murdoch University, Murdoch, WA 6150, Australia.
[C]Department of Primary Industries and Regional Development, Western Australia, South Perth, WA 6151, Australia.