Gesunde Pflanzen (2023) 75:37–48
https://doi.org/10.1007/s10343-022-00794-0

ORIGINAL ARTICLE / ORIGINALBEITRAG

Deep Learning Object Detection for Image Analysis of Cherry Fruit Fly (*Rhagoletis cerasi L.*) on Yellow Sticky Traps

Christian Salamut[1] · Iris Kohnert[1] · Niels Landwehr[1] · Michael Pflanz[2] · Michael Schirrmann[2] · Mohammad Zare[2]

Received: 5 June 2022 / Accepted: 10 November 2022 / Published online: 12 December 2022

Abstract
Insect populations appear with a high spatial, temporal and type-specific diversity in orchards. One of the many monitoring tools for pest management is the manual assessment of sticky traps. However, this type of assessment is laborious and time-consuming so that only a few locations can be controlled in an orchard. The aim of this study is to test state-of-the art object detection algorithms from deep learning to automatically detect cherry fruit flies (*Rhagoletis cerasi*), a common insect pest in cherry plantations, within images from yellow sticky traps. An image annotation database was built with images taken from yellow sticky traps with more than 1600 annotated cherry fruit flies. For better handling in the computational algorithms, the images were augmented to smaller ones by the known image preparation methods "flipping" and "cropping" before performing the deep learning. Five deep learning image recognition models were tested including Faster Region-based Convolutional Neural Network (R-CNN) with two different methods of pretraining, Single Shot Detector (SSD), RetinaNet, and You Only Look Once version 5 (YOLOv5). R-CNN and RetinaNet models outperformed other ones with a detection average precision of 0.9. The results indicate that deep learning can act as an integral component of an automated system for high-throughput assessment of pest insects in orchards. Therefore, this can reduce the time for repetitive and laborious trap assessment but also increase the observed amount of sticky traps

Keywords Annotation · Cherry fruit fly · Deep learning · Insect detection · Sticky traps

✉ Mohammad Zare
mzare@atb-potsdam.de

[1] Department of Computer Science, University of Hildesheim, Hildesheim, Germany

[2] Leibniz Institute for Agricultural Engineering and Bioeconomy (ATB), Potsdam, Germany

Erkennung der Kirschfruchtfliege (*Rhagoletis cerasi L.*) in Bildern von Gelbtafel-Klebefallen mit Methoden des Deep Learning

Zusammenfassung

Insektenpopulationen treten in Obstanlagen mit einer hohen räumlichen, zeitlichen und artenspezifischen Vielfalt auf. Eine wichtige Methode zur Überwachung von Schadinsekten in Obstanlagen ist der Einsatz von Klebefallen. Die manuelle Auswertung der Klebefallen hinsichtlich des Vorhandenseins des jeweiligen Schadinsekts ist mühsam und zeitaufwändig, sodass nur wenige Standorte in einer Obstanlage kontrolliert werden können. In der vorliegenden Studie wurden Bilderkennungsalgorithmen des maschinellen Lernens getestet, um die Kirschfruchtfliege (Rhagoletis cerasi), eine wichtige Schädlingsinsektenart in Kirschplantagen, in Bildern von gelben Klebefallen automatisch zu erkennen. Eine Datenbank aus Bildaufnahmen von gelben Klebefallen mit mehr als 1600 manuell markierten Kirschfruchtfliegen nebst zusätzlichen Metainformationen wurde erzeugt. Zur Verbesserung der Durchführbarkeit des maschinellen Lernens erfolgte eine Erhöhung der Anzahl der Bilder mittels der üblichen Methoden „Spiegeln“ und „Zuschneiden“. Fünf Deep-Learning-Bilderkennungsmodelle wurden getestet, darunter Faster Region-based Convolutional Neural Network (R-CNN) mit zwei verschiedenen Vortrainingsmethoden, Single Shot Detector (SSD), RetinaNet und You Only Look Once Version 5 (YOLOv5). R-CNN- und RetinaNet-Modelle übertrafen die anderen Modelle mit einer durchschnittlichen Erkennungsgenauigkeit von 0,9. Die Ergebnisse zeigen, dass Deep Learning als integraler Bestandteil eines automatisierten Systems zur Hochdurchsatzbewertung gelber Klebefallen zum Monitoring der Kirschfruchtfliege in Kirschplantagen fungieren kann. Durch die Reduzierung der Auswertungszeit der gelben Klebefallen kann zukünftig deren Anzahl und damit die Stichprobendichte in Kirschplantagen erhöht werden.

Schlüsselwörter Annotation · Kirschfruchtfliege · Deep Learning · Insektenerkennung · Gelbtafeln

Introduction

Agricultural plants are often threatened by pests such as specific insects, making it difficult to produce high-quality food. Different pest control methods are available based on cultivation standards. Once the species of a pest is identified, many methods can be improved by adapting them. In order to keep track of that, one can lay out yellow sticky traps that are subsequently evaluated by experts. Additionally, changing climatic conditions are occurring all over the world, owing mostly to the phenomena of climate change that affect temporal and spatial patterns of precipitation and temperature causing significant effects on crop-pest interactions (Heeb et al. 2019). Losses due to pests are not only economic (40% of the world's food supply is destroyed by pests) but also decreasing food security (IPPC Secretariat 2021). Responding appropriately to usually growing healthy food demands under these potentially threatening situations, necessitates the use of innovative pest detection techniques and technologies more than ever (Saleem et al. 2021; Böckmann et al. 2021). One of the main priorities of any pest management solution is to find suitable methods and models in order to detect insects better (Böckmann et al. 2021). Nowadays, scientific and technological advances, particularly in image processing techniques and computer vision technology have enabled the application of new tools for describing areas affected by insects and also facilitating pest management to increase yields in the context of precision horticulture (Zude-Sasse et al. 2016; Cardim Ferreira Lima et al. 2020). The combination of these techniques and data-driven computing tools such as machine learning (ML), more specifically deep learning (DL) approaches, have led to increased accuracy in translating unstructured image data to practical information for the end-user. In recent years ML data driven methods have been more and more employed in insect detection and monitoring systems (Cardim Ferreira Lima et al. 2020; Jiang et al. 2008) and, more specifically, deep learning methods have been employed in pest detection studies (Wenyong et al. 2021). Wang (2022) indicated that the complexity of data preprocessing in traditional methods of artificial intelligence and machine learning models is high, therefore he proposed an improved deep learning model namely, AlexNet, for detecting crop diseases and insect pests. Thenmozhi et al. (2021) applied convolutional neural network (CNN) for two different insect datasets with 24 classes. The results showed more than 90% accuracy for classification using a CNN model. Kuzuhara et al. (2020) studied the application of region based convolutional neural Networks (R-CNN) and "You Only Look Once" v3 (YOLOv3) for insect pest detection. They concluded that deep learning models need a large dataset to optimize parameters during the training stage and consequently proposed data augmentation methods to overcome the lack of data problem. These studies showed that the deep learning approach is one of the most suitable data driven models for image processing and object detection studies for insect pest detection.

The insect detection image processing of installed sticky traps using DL data driven methods can be entwined with two main challenges. On the one hand, many parameters including different types of insects, appropriate annotation methods, image resolution, etc., should be considered when using trap images for object detection. On the other hand, the selection of DL methods that are suitable for processing images needs to be properly tested and adapted. This can be difficult because of the magnitude of available ML methods for object detection. In this context, the objective of this study focuses on the application of different object detection DL methods, namely, faster R-CNN, single-shot detector (SSD), RetinaNet, and YOLOv5, for insect detection on sticky trap images using two high resolution data sets, i.e. a single-class—cherry fruit fly (*Rhagoletis cerasi*)- and a multi-class data set, taken from insect traps on cherry orchards located in eastern Germany.

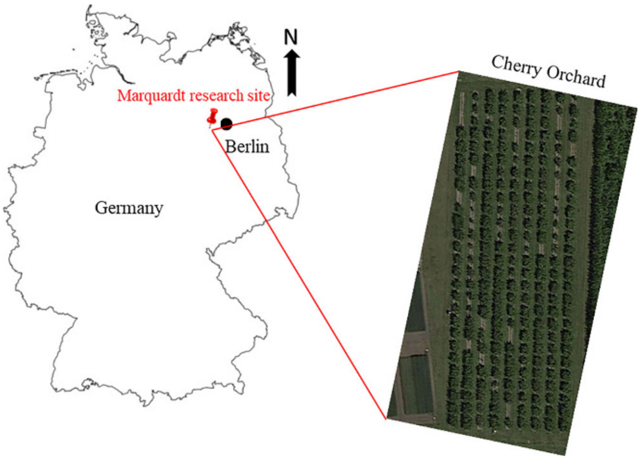


Fig. 1 Study area

Materials and Methods

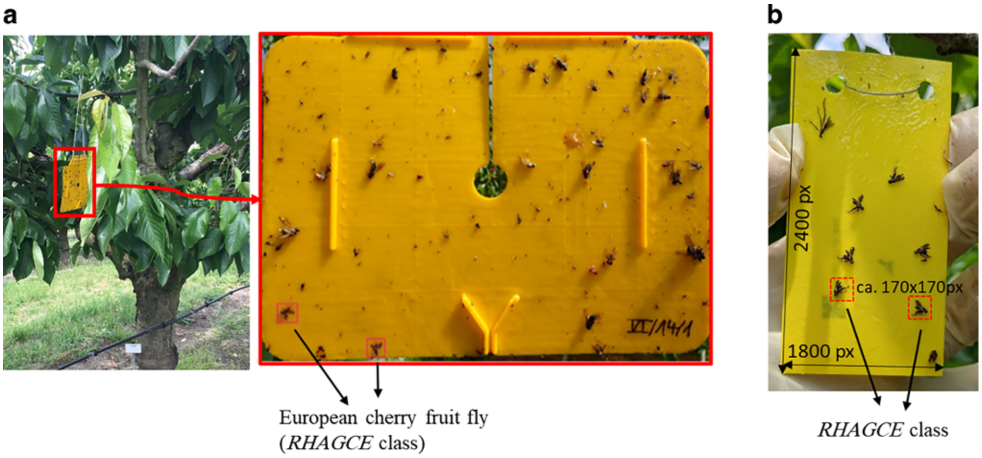
Study Area and Data

The cherry orchard is located in the Leibniz Institute of Agricultural Engineering and Bio-economy (ATB) research site in Marquardt, eastern Germany; the longitude and latitude of the field center is 52°28′01″N 12°57′27″E (Fig. 1). The data consist of two high-resolution sets of images that display insects on hanging yellow sticky traps (Fig. 2) in cherry orchards placed at 0.74 to 2.17 m over the ground in two consecutive years (2020 and 2021). The most serious pest of cherries is the European cherry fruit fly (*Rhagoletis cerasi*), which appears in orchard between mid-May and mid-June (Böhm 1949). The datasets consist of 140 and 850 images for 2020 and 2021, respectively. In 2020, within the images, 47 different classes were detected and labeled, whereas in 2021, within the images, the whole area of each trap was categorized in two classes, namely, *RHAGCE* for European cherry fruit fly and *non-RHAGCE.* Since the target pest is the European cherry fruit fly, all deep learning object detection models in the present study have been trained and validated based on the 2021 labeling method.

Insect Annotation

Many insects find bright yellow to be very attractive and European cherry fruit fly (*Rhagoletis cerasi*) is not an exemption (Lu et al. 2019). As Fig. 2 shows, the images have been labeled in a way that each trapped insect has a class label and a bounding box. For the annotation, the computer vision annotation tool (CVAT) was applied. Using bounding boxes, the CVAT software enabled the insect individuals to be quickly annotated within the images using bounding boxes specified with left-down (*x_min, y_min*), right-up (*x_max, y_max*) points. The images with annotated classes in CVAT were stored as a comma-separated values (csv)

Fig. 2 Example of yellow sticky traps on cherry orchard. **a** Year 2020, **b** Year 2021



and converted to the pattern analysis, statistical modeling and computational (PASCAL) visual object classes (VOC) model format (Everingham et al. 2015). In order to make the data persistent and easier to employ, all information in each individual image was further converted to the Extensible Markup Language (XML) format by PASCAL-VOC (Everingham et al. 2007), which is composed of several attributes, such as filename, size and objects that consists of a label and a bounding box.

Data Augmentation

In order to increase the amount of data, the augmentation techniques including flipping and cropping was used. Flipping is a frequent technique in computer vision that leads to a significant performance improvement (Shorten and Khoshgoftaar 2019). It aids in preventing a model from learning a certain order of pixels used to create an object. By shifting one's point of view, one can gain a more general and fine-tuned "understanding" of an object. Cropping is used to reduce the size of the input. This can boost performance; for example, a batch of a smaller size can be processed faster. While the insects are just a small part of the photo, the insects may occupy a bigger area in the cropped picture. As a result, the model may extract different features than the original sized input. Two methods of cropping were applied. First, grid cropping created patches from a picture of equal size. Second, sliding window cropping created images of a fixed size by moving a window over the image and cutting it out at every step. The latter allows the identical object to appear in a different area of the cropped image (See Table 1). Due to less complexity, i.e., detection of one class only, the single-class problem should be "easier" for the model to learn. Therefore, exploration with grid cropping methods are evaluated on the 2021 data set for feasibility reasons prototypically. The reason is that a small object can be harder to detect in a large image. When cropping the images into sub images, the objects will take up more space when occurring. Also, detecting on smaller images may have a performance boost during train and test time and can be elaborated on in future research.

Table 1 Data augments techniques

Year	Augment tech	Total added images
2020	Flipping[a]	806
2021	Flipping[a]	1203
2021	Grid cropping[b]	50,110
2021	Sliding window cropping[b]	177,325

[a]The resolution of image has not been changed (3248–4032 pixels × 1960–3024 pixels)

[b]The cropped parts has the resolution of (300 pixels × 300 pixels)

Deep Learning Models

As shown in Fig. 3, deep learning (DL) is a subfield of artificial intelligence (AI) and machine learning (ML). The term "deep" in this approach refers to a layered structure in the learning process, and it is not always associated with a deeper understanding of this problem-solving methodology. The sample deep learning model is composed of an input layer, multiple hidden layers and one output layer. Each layer has multiple fully connected nodes. The layer names are self-explanatory, as the input layer handles input data and the output layer delivers output values, for example probabilities for a class in a classification problem. The hidden layers lay between them and do the calculations between the layers. As a result, it is also known as "hierarchical representations learning" and "layered representations learning." (Chollet 2017).

Although the idea of DL was first introduced in 1993, it was not really applied for more than a decade because of the lack of data and high-performance computing hardware as well as popularity of other ML models at the time such as support vector machines (SVM). However, Many labeled datasets have been collected since 2006 and, moreover, significant progress in computers and data training methods have been made in the recent years that removed obstacles to apply DL methods (Fig. 4). In fact, more recently it has become a rather hot topic in image processing, specifically the CNN models (Chollet 2017; Hatt et al. 2019).

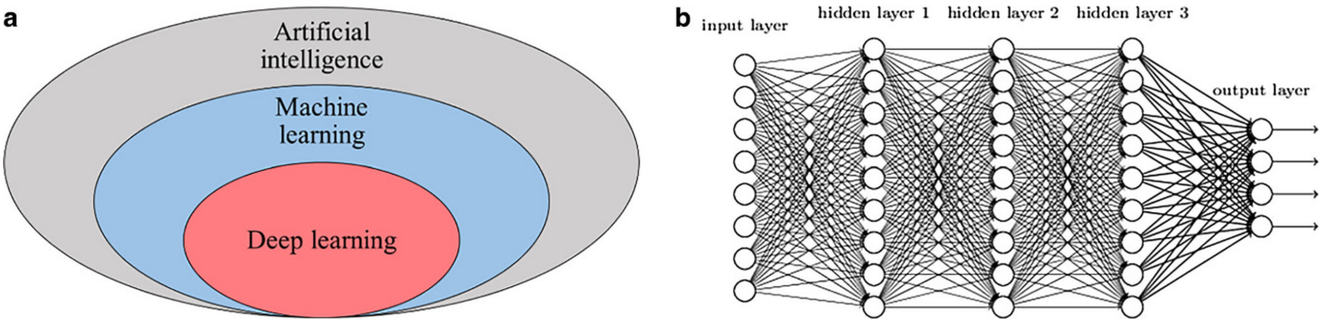


Fig. 3 **a** A visual representation that shows how machine learning (ML) and deep learning (DL) are regarded within the context of artificial intelligence (AI); **b** Architecture of a DL model (Chollet 2017)

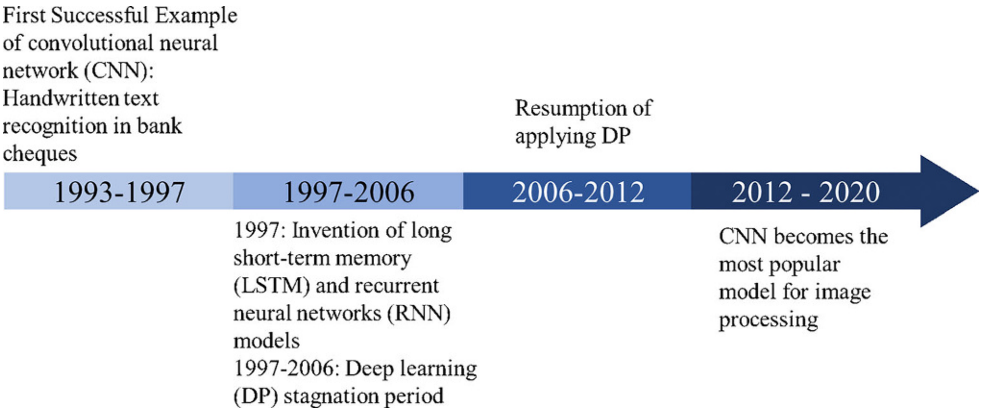

First Successful Example of convolutional neural network (CNN): Handwritten text recognition in bank cheques
Resumption of applying DP
1993-1997
1997-2006
2006-2012
2012 - 2020
1997: Invention of long short-term memory (LSTM) and recurrent neural networks (RNN) models
1997-2006: Deep learning (DP) stagnation period
CNN becomes the most popular model for image processing

Fig. 4 Short history of deep learning models

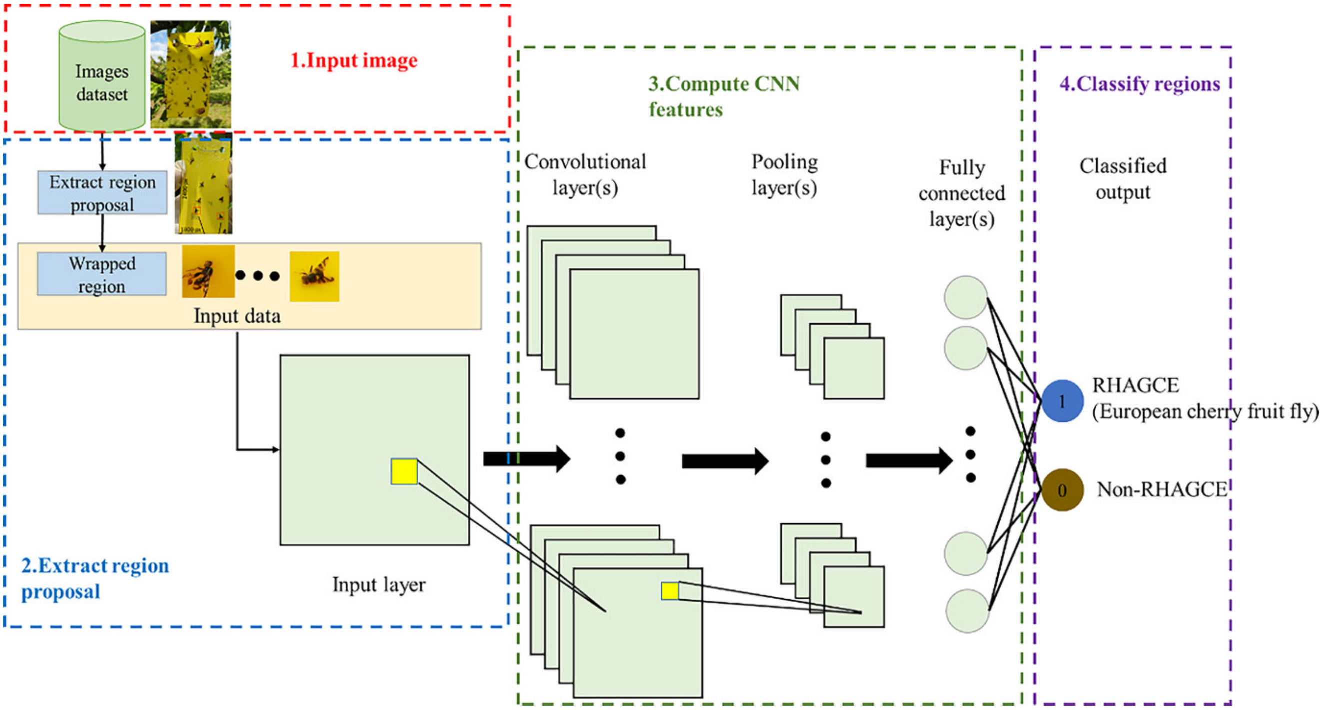
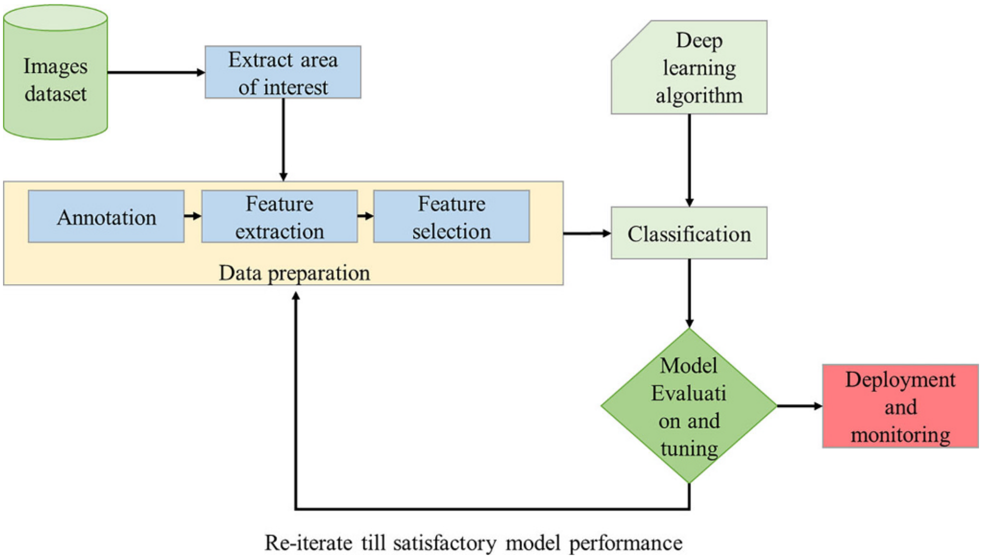


Fig. 5 Region based convolutional neural networks (R-CNN) model for insect detection

In order to apply deep learning to detect cherry fruit flies, five different models were applied in this study: i) Faster-RCNN model pretrained using a residual neural network (ResNet); ii) Faster-RCNN model pretrained using MobileNet; iii) single-shot detector (SSD); iv) RetinaNet (one-stage deep CNN); and v) YOLOv5. Faster R-CNN models are a modified version of R-CNN (Girshick 2015). The model is known as a two-staged approach because it extracts region proposals at the first stage and computes CNN features on latter proposals at second stage.

As shown in Fig. 5, given the input images as three-dimensional matrices (RGB), the model extracts region suggestions for each input image using the selective search technique, which recursively merges comparable neighboring region pairings into bigger ones using a similarity measure. Then, using affine image warping, each proposal is transformed to a fixed size that serves as an input layer for the CNN. Then the CNN generates a feature vector from each area suggestion. Finally, each class is predicted using a separate linear support vector machine (SVM) model. Fast-RCNN is a form of R-CNN that uses a neural network instead of SVM and modifies the feature computation. A further development of both models is called Faster-RCNN, which is a single and unified network for object detection. On the PASCAL VOC 2007 data set, relative to R-CNN, the Faster-RCNN model takes a shorter time (about 0.2 s) for image proposal and detection (Liu et al. 2016). *SSD* is a single-stage object detection model proposed by Liu et al. (2016). Similar to Faster R-CNN, SSD uses an offset prediction of the default boxes and its confi-

Fig. 6 Deep learning workflow used in the present study



dence values but at different scales. *RetinaNet* is a single-stage object detection model based on the focal loss introduced by Lin et al. (2020) that also uses anchor boxes. You Only Look Once (YOLO) proposed by Redmon et al. (2016) is the last single-stage object detection model that divides images into grids at the first step. In a grid, each cell detects objects within itself. The latest version (YOLOv5) has been used in present study. The general workflow of all applied models has been shown in Fig. 6. It includes data preparation and annotation, classification made by a deep learning algorithm and iterative evaluation of the results. In order to give a broad overview of different learning rate behaviors, an extensive grid search is applied for the learning rates [10^{-2}, 10^{-3}, ..., 10^{-7}, 10^{-11} and 10^{-12}] onto each model for the 2020 and 2021 data set. They are used along with a stochastic gradient descent optimizer and a momentum of 0.9 (Ruder 2016). The learning rate hyper-parameters define how much the weights are adjusted during the back-propagation process of the neural network according to the loss of stochastic gradient descent. The training process is combined with a learning rate scheduler that is applied after 15 episodes. It multiplies the learning rate by 0.1 similar to Liu et al. (2016). That should make a network find a more exact solution and fine-tune its parameters in theory.

As shown in Fig. 6, the model evaluation and tuning plays an important role in the DL model. This stage includes the loss function in the training stage and precision/recall metrics in the validation and test stages. The loss functions—also called error function—calculate how far a model output deviates from its ground truth. It is composed of the classification loss (L_{cls}) and the bounding box regression loss (L_{reg}) (Lin et al. 2020).

Backbones and Pre-training

Backbones and pre-training in object detection models have been used for improvement in DL model results. A backbone describes a certain method for feature extraction of images in any CNN model like MobileNet (Howard et al. 2017), ResNet (He et al. 2016) or Visual Geometry Group-16 (VGG-16) network (Krizhevsky et al. 2017). A pre-trained backbone can use its knowledge of extracting certain features out of the data for a new problem and accelerates training, especially when the new data set is similar to the one used during pre-training. A pre-trained backbone network has been applied in this study in order to improve the performance of DL models.

Classification Model Evaluation Metrics

In the case of a classification issue with two output classes (RHAGCE, non-RHAGCE), the prediction model's output is a probability that decides, which class the output is allocated to. There are four possible results of a classification prediction model, namely, true positive (TP), false positive (FP), true negative (TN), and false negative (FN). This list of possible results defines an important metric for the model, called the confusion matrix (Tharwat 2021; Bradley 1997) (Table 2).

Precision and recall are the measures used to evaluate the model based on these four possible outputs. Precision [0, 1] quantifies how accurate the model is when it produces

Table 2 Confusion matrix

Ground truth class/Predicted class	*RHAGCE*	*Non-RHAGCE*
RHAGCE	Count of TPs	Count of FNs
Non-RHAGCE	Count of FPs	Count of TNs

a positive outcome. Recall [0, 1] measures how many correct *TP*s can be produced by the model (Tharwat 2021).

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

Here, TP means that the predicted label correctly estimates the ground truth label, TN denotes that the predicted label correctly estimates the absence of the ground truth label, FP means that the predicted label wrongly estimates a ground truth label, if it is present and FN indicates that the predicted label estimates a ground truth label, if it is absent.

In order to find the degree of overlap between predicted bounding box around the target insect individual and its associated annotated bounding boxes, the object detection methods use the intersection over union (IoU) metric, which is calculated by Eq. 3 (Ren et al. 2015).

$$IoU = \frac{area\ \ (predicted \cap groundtruth)}{area\ \ (predicted \cup groundtruth)} \quad (3)$$

IoU and precision-recall measures are used to compute average precision (AP), which is the most popular evaluation metrics for object detection models. It is defined as the area under the precision-recall curve (AUC-PR) evaluated at α threshold of IoU is equal to $AP@\alpha$. The latter parameter has been calculated in the VOC2007 challenge (Everingham et al. 2015) by the following equation:

$$AP = \frac{1}{11}\sum_{r=0}^{r=1} p_{\text{int}}(r)\,,\ \ r = 0, 0.1, 0.2, ..., 1 \quad (4)$$

$$p_{\text{int}}(r) = p\left(\widetilde{r}\right) \quad , \widetilde{r} : \widetilde{r} \geq r \quad (5)$$

where p_{int} is the interpolated average precision and interpolated average precision (p_{int}). For a class and a set of precision at a certain recall level (r), one sums the maximum precision at each rank level ($\widetilde{r}$), i.e., at every recall rank. This measure approximates the AUC-PR (Everingham et al. 2015).

The main idea of using AP is to reduce the impact of the "wiggles" in the precision-recall curve. This parameter is shown in the form of $AP@\alpha$, which means that the AP precision is defined at α threshold of IoU (Ren et al. 2015). In the present study, *AP@0.5* and *AP@0.75* was calculated for all DL models. If the number of classes (C) exceeded one, the AP will be the average value for all classes or *m*AP:

$$mAP = \frac{1}{C}\sum_{i=1}^{i=C} AP_i\,,\ \ C = 1, 2, ... \quad (6)$$

Since all models run by an efficient computational system, frame per seconds (fps) was used to describe the performance of the model. FPS determines how quickly the object detection DL model processes images and generates the model output.

Results and Discussions

Insect Annotation Results

There are a total number of 4905 annotations for 140 images of the 2020 data set, more than 90% of the annotations belong to the 7 upper classes in terms of occurrence (at least 100 times). By eliminating the first two classes, "NOTHIN" and "BACKGR", the most frequent class was *RHAGCE* or European cherry fruit fly, which was observed in 60% of the sticky trap's images. Regarding the observation results in 2020, it was decided that in the 2021 images only the *RHAGCE* class will be considered as a target pest for cherry orchards (See Table 3). The annotation results clearly showed that the size of both datasets is limited, ergo, data augmentation techniques are needed to increase the performance and results of the DL models by creating additional and diverse instances for training datasets. Table 4 shows the results of data classification after applying augmentation techniques.

Table 3 Class distribution of insects on yellow sticky traps in cherry orchard of the study area

2020 annotation (140 Images—47 classes)			
Scientific name	Class-Name	Insect count	Observed in # images
Indefinable insects	NOTHIN	2644	89
Background of image	BACKGR	885	23
Rhagoletis cerasi	RHAGCE	469	84
Drosophila	1DROSG	183	38
Chrysopidae	1CHASF	151	12
Muscidae	1MUSCF	109	31
Formicidae	1CHSAF	100	19
Other	40-Classes	364	114
2021 annotation (850 Images—1 class)			
Rhagoletis cerasi	RHAGCE	1626	401

Table 4 Class distribution of insects (most frequent classes) after data augmentation

2020 annotation			
Scientific name	Class-Name	Insect count	Observed in # images
Indefinable insects	NOTHIN	16,638	610
Background of image	BACKGR	6048	164
Rhagoletis cerasi	RHAGCE	3424	573
Drosophila	1DROSG	1073	216
Chrysopidae	1CHASF	919	78
Muscidae	1MUSCF	775	215
Formicidae	1CHSAF	393	97
2021 annotation			
Rhagoletis cerasi	RHAGCE	4872	1203

Deep Learning Models Evaluation

For the training, validating and testing of the DL models, the augmented 2021 data set was randomly divided into three sets, wherefore the length of the training and testing data sets are 977 and 108 images, respectively. The validation data consisted of the remaining 118 images. All training, testing and validating images were resized to 1000 × 1000 pixels to have identical image sizes. The training results of all models showed that after 30 episodes (even earlier) the amount of loss function stopped changing, ergo, the calculated training weights after 30 episodes have been applied for validating and testing stages. Fig. 7 shows the loss development graphs of some DL models. Each subgraph shows a different model and its learning rate. The summed loss is the sum of the loss for the bounding box regression and classification, respectively.

Based on the model validation results, the appropriate learning rates for insect detection on the 2021 data set, were chosen to be 10^{-2}, 10^{-3} and 10^{-5} with an AP of 0.9 (Fig. 8).

After finding the appropriate learning rates, the model performance was measured on the test data set, where the goal was to find out the performance on the unseen data. For the learning rates, the results from training were limited to the best values, i.e. (10^{-2}, ..., 10^{-5}). The models that were dropped due to a loss of "NaN" were also ignored during training because there are no weights available for testing. For testing, the weights after training for 30 episodes have been used. The average precision was calculated for the IoU thresholds 0.5 (AP@0.5) and 0.75 (AP@0.75) (Table 5).

Regarding the training, validating and test results, the Faster R-CNN-MobileNet and RetinaNet models with learning rates of 0.01 and 0.001 were suitable DL algorithms for detecting European cherry fruit flies in yellow sticky traps images.

As Fig. 9 shows, the DL model detected European cherry fruit flies on sticky traps after 30 epochs, properly. The impact of the sliding window cropping (SW) data augmentation technique on performance of the DL model has been evaluated by cropping the original size images into the 300 × 300 pixels patches and predictions for all patches

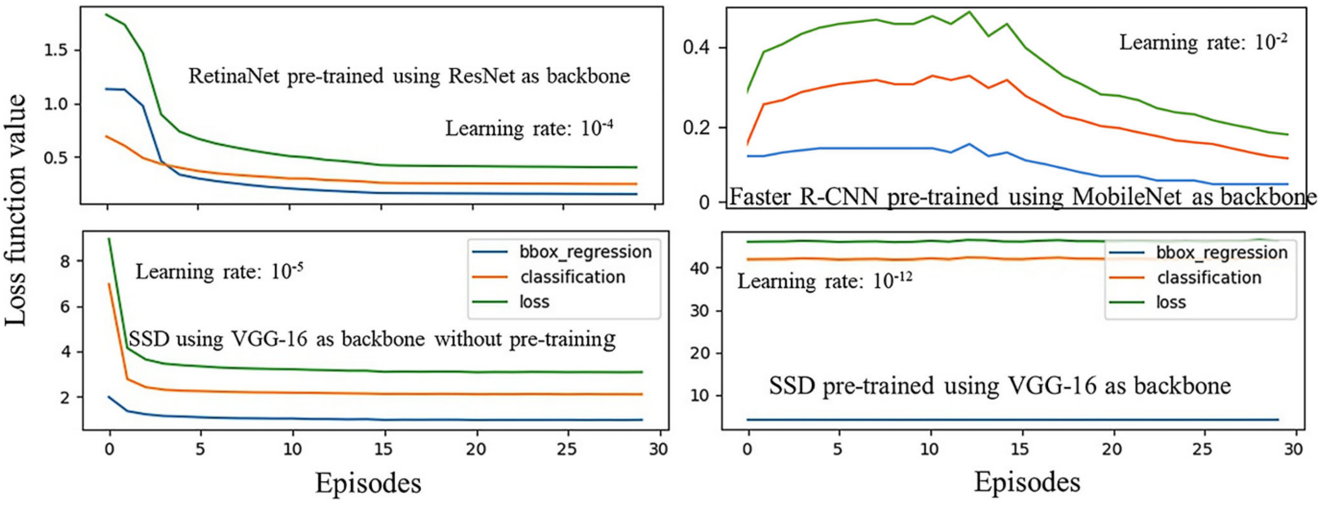


Fig. 7 Loss function graphs during training stage for four different DL model; the loss (*green*) is the summation of loss for the bounding box regression (*blue*) and classification (*orange*)

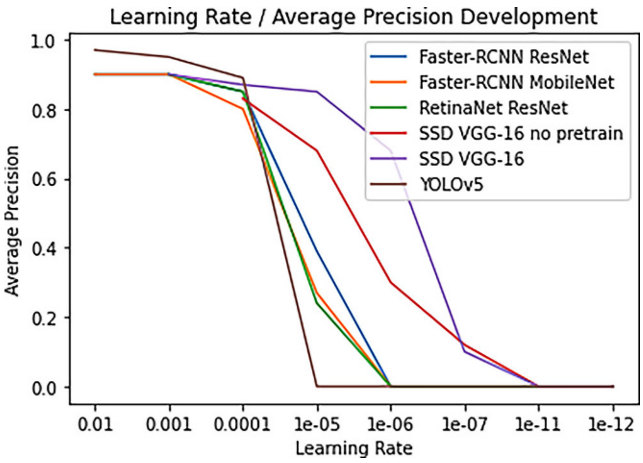


Fig. 8 Grid search learning rate development considering average precision on the validation data set for year 2021

merged and mapped to the original size of the image. In this regard, a RetinaNet was trained with the learning rate of 0.001. The results show a recall and precision values of 1 and 0.9 on the validation data, respectively (see Fig. 10). Although the performance of the model improved, training the SW model is very slow and the performance of the computing system degrades noticeably. Moreover, the trained model on SW cropped data computes 0.34 frame per second (fps) on average, which is significantly smaller than DL models before applying the latter data augmentation method i.e. 11 to 16 fps.

Regarding the results of the 2021 DL models, Faster R-CNN and RetinaNet models have been chosen for multi-class object detection for the 2020 data set. After data augmentation, 732 images splitting as 588/71/73 images for training/validation/test stages were selected. For this data set only classes that occurred over 100 times (except “nothing” and “background”) were considered. In training and validation stages, the Faster R-CNN ResNet model with learning rate, mAP@0.5 and fps of 0.01, 0.5 and 1.35, respectively, outperformed the other models. Moreover, the prediction results for *Rhagoletis cerasi, Drosophila, Muscidae* and *Formicidae* classes are shown in Table 6.

The precision recall curves (Fig. 11) of Faster R-CNN ResNet model with learning rate of 0.01 for multi-class object detection clearly showed that the AP@0.5 for “*RHAGCE*” class is considerably higher than other individual classes. It is related to the fact that the DL models have been already trained on the same class using the 2021 data set.

The single-class data set had an output performance of around 9–16 frames per second (fps), which means that the model takes 111 millisecond (ms) to 62.4 ms on average for computing one input image. For on-site detection, this can be considered as a decent value providing almost real-time detection. However, when considering image-streaming of 30 fps, which is a typical frame rate for videos, the models are not fast enough to be used for what one can consider “live” detection with sufficient performance. Taking a closer look at the loss values over the episodes for the 2021 model, one notices that a smaller loss does not always correspond to better results. This is why it is also very important to consider metrics like the AP or plot the predicted

Table 5 Average precision results on test data 2021 for the intersection over union (IoU) threshold of 0.5 and 0.75 (AP@0.5 and AP@0.75)

DL model	Learning rate (lr)	AP@0.5	AP@0.75	Average fps
Faster R-CNN MobileNet	0.01	0.88	0.69	15.47
Faster R-CNN MobileNet	0.001	0.88	0.59	15.76
RetinaNet ResNet	0.001	0.88	0.6	11.91
SSD VGG-16	0.0001	0.88	0.55	9.33
SSD VGG-16	0.001	0.87	0.57	9.48
Faster R-CNN ResNet	0.001	0.86	0.59	10.99
RetinaNet ResNet	0.0001	0.86	0.42	11.88
Faster R-CNN ResNet	0.0001	0.84	0.53	9.83
SSD VGG-16	0.00001	0.84	0.35	9.33
SSD VGG-16 no pretrain	0.0001	0.83	0.37	9.33
Faster R-CNN MobileNet	0.0001	0.81	0.33	15.63
YOLOV5	0.001	0.76	0.73	15.04
YOLOV5	0.0001	0.76	0.67	14.75
YOLOV5	0.01	0.75	0.75	14.21
SSD VGG-16 no pretrain	0.00001	0.75	0.18	9.24
Faster R-CNN ResNet	0.01	0.51	0.71	10.96
RetinaNet ResNet	0.00001	0.28	0.1	10.34
Faster R-CNN MobileNet	0.00001	0.19	0.09	13.35
YOLOV5	0.00001	0	0	14.87

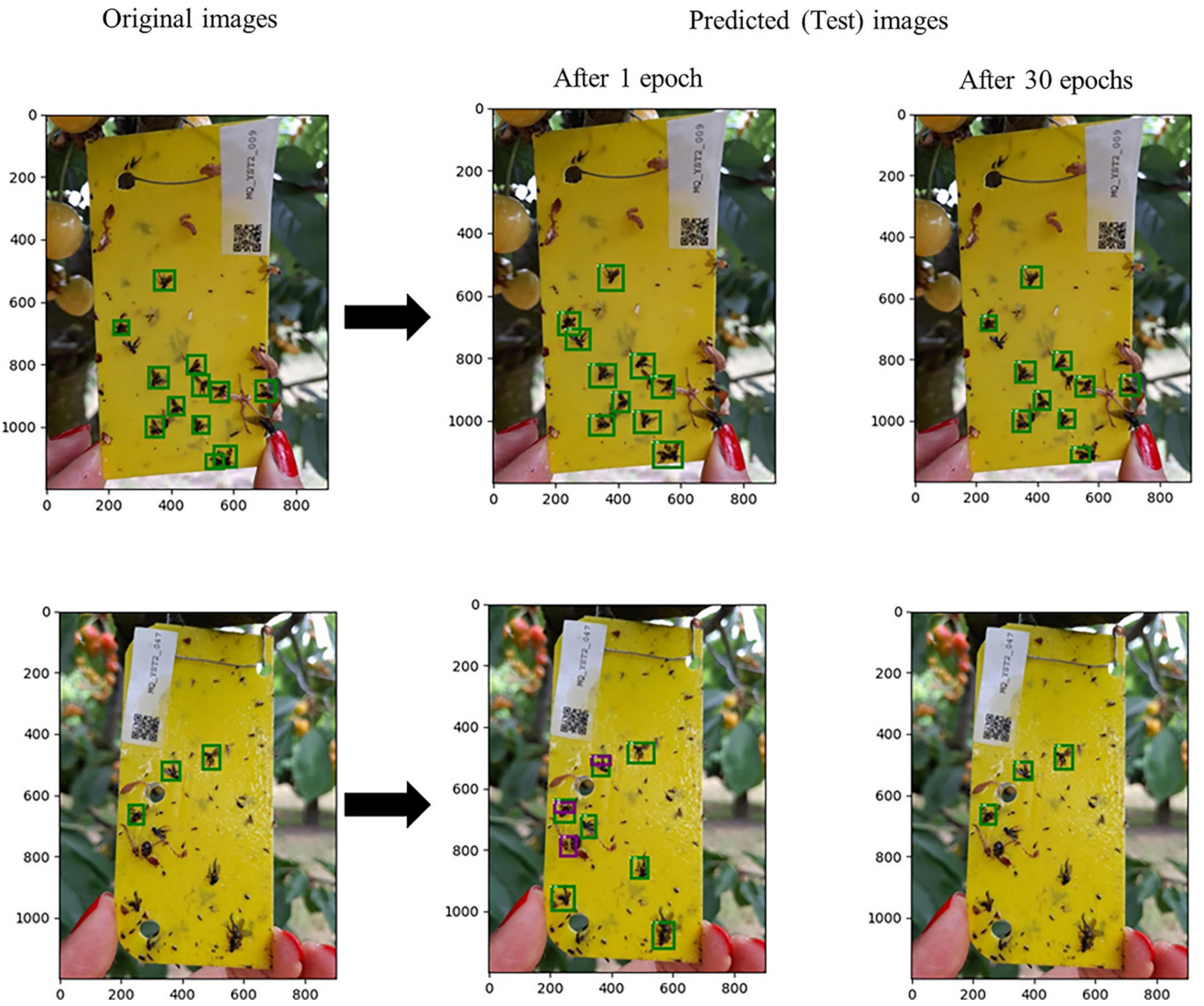


Fig. 9 Results of the Faster R-CNN-MobileNet with learning rate of 0.01 for two test (prediction) images. Ground truth and predictions with confidences over 0.5 are *green* bordered. Predictions that have a confidence value below 0.5 are *purple* bordered

predictions. As Kuzuhara et al. (2020) have already shown, increasing the dataset using augmentation methods could increase the model's performance. By adding more artificial data, the data set will get better balanced. In the present study, five different deep learning models were applied to detect cherry fruit flies on yellow sticky traps images and, consequently, choose the better models for multi-class insect detection in the study area. This is a complementary of Böckmann et al. (2021) study published in the literature. As indicated in Thenmozhi et al. (2021) and Wenyong et al. (2021), DL can be valuable modeling tools to help increase our understanding of the complex insect detection processes in pest management solution linked to the agricultural management practices.

Conclusions

This study shows different methods and approaches for applying computer vision to the problem of insect detection. The best deep learning (DL) model (Faster R-CNN) reach an average precision of around 0.88 for the single class data set. The speed performance for the single-class data set is around 9–16 frames per second during test time. Since the final goal is to develop a system that can automatically detect the type and number of insect pests on sticky traps, the results can be considered as a step towards real-time detection. The multi-class data set was limited to its most occurring classes and achieved a mean average precision of around 0.51. The best class (cherry fruit fly) in the multi-class object detection model achieved the average precision of 0.82. It can be concluded that the imbalance in the data

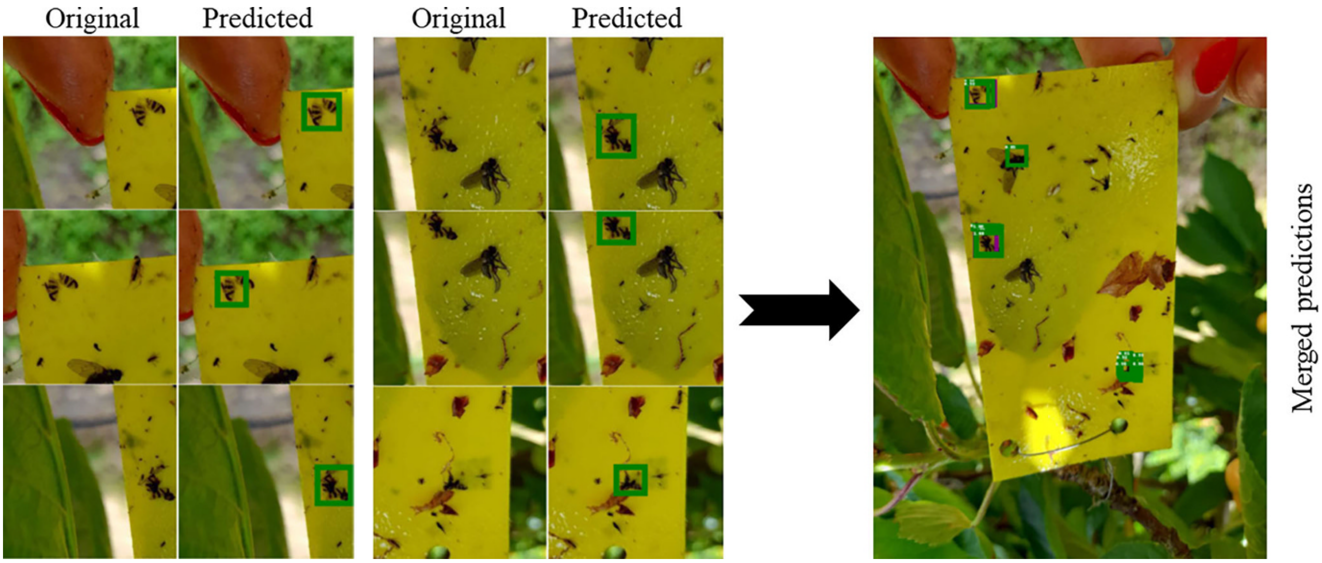


Fig. 10 RetinaNet + Prediction on 300 × 300 pixels cropped with sliding window

Table 6 Results of test (prediction) stage with AP@0.5 for *Rhagoletis cerasi (RHAGCE)*, *Drosophila (1DROSG)*, *Muscidae (1MUSCF)* and *Formicidae (CHSAF)* classes

DL model	Learning rate (lr)	mAP@0.5	Average fps
Faster R-CNN ResNet	0.01	0.51	1.44
Faster R-CNN ResNet	0.0001	0.14	1.72
Faster R-CNN MobileNet	0.001	0.19	1.52
Faster R-CNN MobileNet	0.0001	0.1	1.37
RetinaNet ResNet	0.001	0.42	1.2
RetinaNet ResNet	0.00001	0	1.74

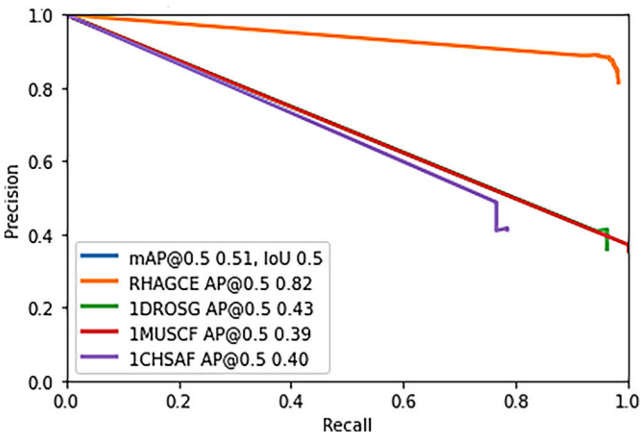


Fig. 11 The precision recall curves of Faster R-CNN ResNet model with learning rate of 0.01 for multi-class object detection with mAP@0.5 of 0.51. (Classes: *Rhagoletis cerasi* (RHAGCE), *Drosophila* (1DROSG), *Muscidae* (1MUSCF) and *Formicidae* (CHSAF))

set causes performance differences between classes. This lack of proportion in datasets can be fixed by adding artificial data for the underrepresented classes in future studies. The main achievement of current study is to successfully translate results of DL models into useful information for horticultural management.

Furthermore, the results show possibilities of improving the model performance by increasing the dataset using sliding window data augmentation for single class. This method is time-consuming and only achieves around 0.34 fps which is far away from real-time. However, this method gives a different view on the problem and suggests tweaks like parallelization and merging of bounding boxes to increase the performance.

Funding This research was funded by the Federal Ministry of Food and Agriculture (BMEL) of Germany with grant agreement number 2818505A18.

Funding Open Access funding enabled and organized by Projekt DEAL.

Conflict of interest C. Salamut, I. Kohnert, N. Landwehr, M. Pflanz, M. Schirrmann and M. Zare declare that they have no competing interests.

References

Böckmann E, Pfaff A, Schirrmann M, Pflanz M (2021) Rapid and low-cost insect detection for analysing species trapped on yellow sticky traps. Sci Rep. https://doi.org/10.1038/s41598-021-89930-w

Böhm H (1949) Untersuchungen über die Lebensweise und Bekämpfung der Kirschfliege (Rhagoletis cerasi L.). Pflanzenschutzberichte 3:177–185

Bradley AP (1997) The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern Recogn 30(7):1145–1159

Cardim Ferreira LM, Damascena de Almeida LME, Valero C, Pereira Coronel LC, Gonçalves Bazzo CO (2020) Automatic detection and monitoring of insect pests—A review. Agriculture 10:161. https://doi.org/10.3390/agriculture10050161

Chollet F (2017) Deep learning with python. Manning Publications

Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A The Pascal visual object classes challenge 2007. http://host.robots.ox.ac.uk:8080/pascal/VOC/voc2007/. Accessed 01 December 2022

Everingham M, Eslami SMA, Van Gool L (2015) The PASCAL visual object classes challenge: a retrospective. Int J Comput Vis 111:98–136. https://doi.org/10.1007/s11263-014-0733-5

Girshick R (2015) Fast R-CNN. 2015 IEEE International Conference on Computer Vision (ICCV), pp 1440–1448 https://doi.org/10.1109/ICCV.2015.169

Hatt M, Parmar C, Qi J, Naqa IE (2019) Machine (deep) learning methods for image processing and radiomics. IEEE Trans Radiat Plasma Med Sci 3(2):104–108. https://doi.org/10.1109/TRPMS.2019.2899538

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

Heeb L, Jenner E, Cock MJW (2019) Climate-smart pest management: building resilience of farms and landscapes to changing pest threats. J Pest Sci 92:951–969. https://doi.org/10.1007/s10340-019-01083-y

Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861

Jiang JA, Tseng CL, Lu FM, Yang EC, Wu ZS, Chen CP, Lin SH, Lin KC, Liao CS (2008) A GSM-based remote wireless automatic monitoring system for field information: a case study for ecological monitoring of the oriental fruit fly, Bactrocera dorsalis (Hendel). Comput Electron Agric 62:243–259

Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. Commun ACM 60(6):84–90. https://doi.org/10.1145/3065386

Kuzuhara H, Takimoto H, Sato Y, Kanagawa A (2020) Insect pest detection and identification method based on deep learning for realizing a pest control system. 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), pp 709–714 https://doi.org/10.23919/SICE48898.2020.9240458

Lin TY, Goyal P, Girshick R, He K, Dollár P (2020) Focal loss for dense object detection. IEEE Trans Pattern Anal Mach Intell 42(2):318–327. https://doi.org/10.1109/TPAMI.2018.2858826

Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Single shot multibox detector. In: Leibe B, Matas J, Sebe N, Welling M (eds) Computer vision—ECCV 2016. Springer, Cham, pp 21–37

Lu CY, Arcega Rustia DJ, Lin TT (2019) Generative adversarial network based image augmentation for insect pest classification enhancement. IFAC-PapersOnLine 52(30):1–5. https://doi.org/10.1016/j.ifacol.2019.12.406

Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 779–788 https://doi.org/10.1109/CVPR.2016.91

Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. Paper presented at the Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal. vol 1

Ruder S (2016) An overview of gradient descent optimization algorithms (cite arxiv:1609.04747Comment: 12 pages, 6 figures)

Saleem MH, Potgieter J, Arif KM (2021) Automation in agriculture by machine and deep learning techniques: a review of recent developments. Precis Agric 22:2053–2091. https://doi.org/10.1007/s11119-021-09806-x

Secretariat IPPC (2021) Scientific review of the impact of climate change on plant pests—A global challenge to prevent and mitigate plant pest risks in agriculture, forestry and ecosystems. FAO, Rome https://doi.org/10.4060/cb4769en (on behalf of the IPPC Secretariat)

Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. J Big Data 6:60. https://doi.org/10.1186/s40537-019-0197-0

Tharwat A (2021) Classification assessment methods. Appl Comput Informatics 17(1):168–192. https://doi.org/10.1016/j.aci.2018.08.003

Thenmozhi K, Dakshayani S, Srinivasulu RU (2021) Insect classification and detection in field crops using modern machine learning techniques. Inf Process Agric. https://doi.org/10.1016/j.inpa.2020.09.006

Wang B (2022) Identification of crop diseases and insect pests based on deep learning. Sci Program. https://doi.org/10.1155/2022/9179998

Wenyong L, Tengfei Z, Zhankui Y, Ming L, Chuanheng S, Xinting Y (2021) Classification and detection of insects from field images using deep learning for smart pest management: a systematic review. Ecol Inform 66:101460. https://doi.org/10.1016/j.ecoinf.2021.101460

Zude-Sasse M, Fountas S, Gemtos TA, Abu-Khalaf N (2016) Applications of precision agriculture in horticultural crops. Eur J Hortic Sci 81:78–90. https://doi.org/10.17660/eJHS.2016/81.2.2

Christian Salamut (Born in 1990) is currently a MSc. student in Computer Sciences at the University of Hamburg, Germany. He graduated from his bachelor at the University of Hildesheim in Applied Computer Science in March 2022. He is specialized in developing deep learning methods and their implementation in Python.

Mohammad Zare PhD. (Born in 1984) is currently working as a research scientist with focus on developing machine/deep learning algorithms for image processing at the Leibniz Institute for Agricultural Engineering and Bioeconomy (ATB), Potsdam, Germany. He graduated from his PhD in July 2017 at the University of Kassel, faculty of Civil and Environmental Engineering, Kassel, Germany.