agriculture

Review

A Systematic Review on Automatic Insect Detection Using Deep Learning

Ana Cláudia Teixeira [1,2,*], José Ribeiro [1], Raul Morais [1,3], Joaquim J. Sousa [1,2] and António Cunha [1,2]

[1] Engineering Department, School of Science and Technology, UTAD—University of Trás-os-Montes e Alto Douro, 5000-801 Vila Real, Portugal
[2] Institute for Systems and Computer Engineering, Technology and Science (INESC-TEC), 4200-465 Porto, Portugal
[3] Centre for the Research and Technology of Agro-Environmental and Biological Sciences, University of Trás-os-Montes e Alto Douro, 5000-801 Vila Real, Portugal
* Correspondence: ana.c.teixeira@inesctec.pt

Abstract: Globally, insect pests are the primary reason for reduced crop yield and quality. Although pesticides are commonly used to control and eliminate these pests, they can have adverse effects on the environment, human health, and natural resources. As an alternative, integrated pest management has been devised to enhance insect pest control, decrease the excessive use of pesticides, and enhance the output and quality of crops. With the improvements in artificial intelligence technologies, several applications have emerged in the agricultural context, including automatic detection, monitoring, and identification of insects. The purpose of this article is to outline the leading techniques for the automated detection of insects, highlighting the most successful approaches and methodologies while also drawing attention to the remaining challenges and gaps in this area. The aim is to furnish the reader with an overview of the major developments in this field. This study analysed 92 studies published between 2016 and 2022 on the automatic detection of insects in traps using deep learning techniques. The search was conducted on six electronic databases, and 36 articles met the inclusion criteria. The inclusion criteria were studies that applied deep learning techniques for insect classification, counting, and detection, written in English. The selection process involved analysing the title, keywords, and abstract of each study, resulting in the exclusion of 33 articles. The remaining 36 articles included 12 for the classification task and 24 for the detection task. Two main approaches—standard and adaptable—for insect detection were identified, with various architectures and detectors. The accuracy of the classification was found to be most influenced by dataset size, while detection was significantly affected by the number of classes and dataset size. The study also highlights two challenges and recommendations, namely, dataset characteristics (such as unbalanced classes and incomplete annotation) and methodologies (such as the limitations of algorithms for small objects and the lack of information about small insects). To overcome these challenges, further research is recommended to improve insect pest management practices. This research should focus on addressing the limitations and challenges identified in this article to ensure more effective insect pest management.

Keywords: insect detection; insect classification; smart pest monitoring; deep learning; insects traps

check for updates

Citation: Teixeira, A.C.; Ribeiro, J.; Morais, R.; Sousa, J.J.; Cunha, A. A Systematic Review on Automatic Insect Detection Using Deep Learning. *Agriculture* **2023**, *13*, 713. https://doi.org/10.3390/agriculture13030713

Academic Editor: Yanbo Huang

Received: 2 February 2023
Revised: 23 February 2023
Accepted: 17 March 2023
Published: 19 March 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Insect pests cause between 20% and 40% of the world's agricultural production losses every year [1], making agricultural practices dependent on pesticides. Applying these chemical components has become the most profitable solution for crop protection with the appearance of intensive agriculture [2]. There has been an increase in resistant pests, the poisoning of organisms, air pollution, water pollution, poisoning, and other health problems due to the chemical properties of pesticides and their continued use over decades [3].

Insect monitoring is necessary for the early detection of pests to avoid the excessive use of pesticides [4]. Integrated pest management (IPM) systems that can reduce the overuse of pesticides started to be developed in recent decades by the research community, monitoring plagues and applying precise amounts when needed [5,6]. The main objective of insect monitoring is to provide farmers with a decision-making tool, contributing to the optimisation of their crops, increasing environmental sustainability, and improving the quality and yield of production [7]. One form of monitoring is detecting and counting insects that are attracted to traps distributed along the agricultural fields where the insects will be captured. A typical monitoring approach is made by specialists, who recognise and manually count insects caught in traps [4,8]. However, this task is very time consuming, susceptible to errors, and sometimes subjective—each trap may contain dozens of insects of different species [9].

Smart pest monitoring (SPM) has emerged with rapid advances in fields such as artificial intelligence (AI) and the Internet of things (IoT), allowing automatic data acquisition, remote transmission, data processing, and decision making [5,10]. AI algorithms improve data processing and propose hypotheses for increasingly accurate decision-making. AI is a general field that encompasses machine learning (ML) and deep learning (DL) [11]. ML is a type of AI that uses algorithms and statistical models to allow a system to improve its performance of a specific task over time. In other words, ML allows a system to learn from data without being explicitly programmed [11]. DL is a specific type of ML that involves the use of neural networks, which are algorithms inspired by the brain's structure. These algorithms are made up of many layers of interconnected nodes and can learn complex patterns in data. DL has been particularly successful for computational vision tasks suited for image classification, segmentation, detection, and other tasks related to image recognition [12]. Several AI techniques for insect automatic detection and counting have been developed and published with data-driven methods; e.g., DL. However, automatic detection and counting is still an open problem, and several challenges remain [4].

This study aimed to perform a literature review of DL methods for insect classification and detection. The review includes papers submitted until 5 February 2022. For this review, 36 studies were chosen according to predefined criteria. These studies were carefully examined, and their methodologies, results, and database sources were thoroughly analysed. Through this analysis, the most successful methods were identified, and the study also highlighted open challenges and potential solutions. The focus of this research is to address the challenges identified and propose solutions to improve insect pest management practices, with the ultimate goal of achieving better and more effective results.

Regarding the novelty of this article, the following can be listed:

- The integration of deep learning techniques for automatic insect detection in traps;
- A systematic review and analysis of recent research on deep learning methods for insect detection;
- An investigation of the effectiveness of deep learning in addressing the challenges of traditional insect detection methods;
- A comparison of deep learning methods for insect classification and detection;
- The identification of key research gaps and opportunities for future work in this area.

The previous novelties highlight the following needs that this work can help overcome:

- Insect infestations can cause significant crop losses and economic damage in agricultural production;
- Traditional methods of insect detection and control can be time-consuming, labour-intensive, and potentially harmful to the environment and human health;
- Deep learning techniques have the potential to improve the efficiency and effectiveness of insect detection, leading to more sustainable and profitable farming practices;
- A systematic review of recent research on deep learning methods for insect detection can provide valuable insights and guidance for future research and development in this field;

- The results of this study can help inform and improve the use of deep learning techniques for insect detection in practical applications.

This paper is structured as follows: Section 2 provides a background for the theme of automatic image acquisition and insect detection and classification evolution. Section 3 describes the research questions, the inclusion criteria, the research strategy, and the study characteristics. Section 4 presents the main findings in terms of methodologies developed for this specific purpose. DL-based applications for insect detection, classification, and detection are summarised, and the main detected challenges and gaps are provided. In Section 5, we discuss and summarise the results found. Finally, in Section 6, the conclusion and recommendations for the future are presented.

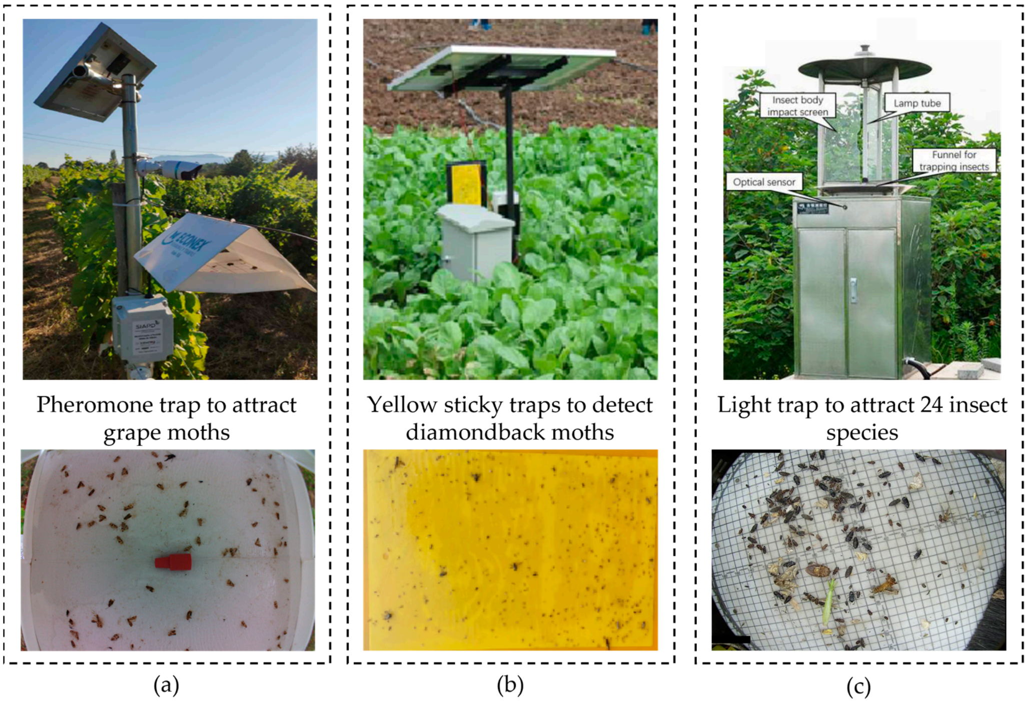
2. Theoretical Background

Pest control seeks to follow a diversified pest reduction strategy combined with other forms of control and the use of chemical components. A possible way to deal with some crop pests is by installing traps to attract insects [13]. Insect traps are essential elements of SPM. These can be sex pheromone traps, yellow sticky traps, and light traps [13]. The type of trap is chosen according to the kind of plantation or the pest to be monitored [14]. Traps are frequently observed by qualified personnel to determine the number of insects that have been trapped in each trap. There is a need to travel regularly to each location to carry out this task, making this work expensive [8]. On the other hand, traps can control large areas and not interfere with crop quality as chemical compounds do. The main advantages of traps are their practical and reliable response for pest monitoring, the identification of the right time to intervene with pesticides, the identification and quantification of pests, and the reduction of costs and harmful effects on human beings, the environment, and natural resources [15]. Therefore, traps yield information about the timing of the appearance and activity of certain pests and auxiliaries, allowing treatments to be carried out at the right time [16]. Monitoring insects through remote sensing is possible with the emergence of more sophisticated technologies, being an asset for agricultural activity and enabling real-time monitoring [17]. Image acquisition devices are installed in fields to monitor traps, and insect detection and classification techniques are used.

Several authors have proposed different SPM systems. The possibility of implementing these mechanisms and acquiring high-resolution images allows the remote control of pests, reduces the need for human resources, and allows decision-making at a distance. The resolution of the acquired images has a great influence on the methods applied in intelligent image processing [18].

Preti et al. [7] reviewed the evolution of insect pest detection in terms of methodology and equipment used. They observed that the first equipment used to collect images in traps were optical sensors directly implemented in traps in 1985. With the integration of IoT, big data, AI, and other modern information technologies, it has been possible to develop and adapt various devices for pest monitoring. As shown in Figure 1, several IoT devices are installed at strategic points on the agricultural plot to collect images from the traps; the images are captured and stored on a server and later processed through digital image processing techniques and by DL [4,19].

Ramalingam et al. [10] proposed a real-time remote monitoring system for insect traps based on IoT and DL. Saranya et al. [20] developed a methodology using image processing and a passive infrared sensor to detect the presence of insects by the heat radiated by their bodies. Image processing is used to capture images of the pest to confirm its presence in the field. Rustia and Lin [21] developed an image monitoring system connected via Wi-Fi, where each trap was equipped with a sensor and camera placed 80 mm away. Every 10 min, an image was collected and sent to a remote server for processing. In the processing, several insect detection and recognition algorithms were used.


Pheromone trap to attract grape moths

Yellow sticky traps to detect diamondback moths

Light trap to attract 24 insect species

Figure 1. Devices installed in the field to collect images of traps and the respective images collected. (**a**) Pheromone trap in a vineyard to attract grape moths provided by [22]; (**b**) yellow sticky traps installed to detect diamondback moths adapted from [23]; (**c**) light trap to attract 24 major pest classes specified by the Chinese Ministry of Agriculture. Images adapted from the dataset Pest24 [24].

With pest monitoring through sensing, methods began to be developed for detecting and identifying pests based on image processing and ML techniques (summarised in Table 1). There are many different approaches to insect detection using ML, and different algorithms may be better suited to different tasks [25,26].

Table 1. Study analysis based on image processing and ML techniques.

Paper	Year	Task	Method	Disadvantages
[27]	2008	Counting whiteflies	Low pass filter, binarisation, and other image processing operations	Methods developed for the resolution of only the proposed task. May be adaptable to other scenarios
[28]	2015	Detection of whiteflies, aphids, and thrips	Identification with a watershed algorithm to segment insects from the background	
[29]	2015	Counting whiteflies	A k-means grouping is applied in each image converted into a colour space	
[30]	2016	Classification of 24 insect species	Multiple task sparse representation and multiple kernel learning techniques	
[21]	2017	Classification of Thysanoptera	Support vector machine and other image processing operations	
[31]	2017	Classification of pests in pomegranate	Support vector machine and other image processing operations	

Qiao et al. [27] proposed a simple image processing system to automatically estimate the number of whiteflies on sticky traps. Initially, the noise was eliminated with a low pass filter; then, the images were converted to grayscale and transformed into binary images. The authors used ten different threshold levels to determine the optimal image level. The pixels with a value greater than that of the defined threshold were white, and the smallest one was black; thus, it was possible to detect the whiteflies. The method proved to be very effective for adult whiteflies. However, it only worked for whiteflies on sticky traps.

Xia et al. [28] developed an automatic method for whitefly, aphid, and thrip identification in greenhouses. The method starts by using the watershed algorithm to segment insects from the background. With the Mahalanobis distance, the insect's colouring characteristics were extracted to identify the species of different insects. Comparing the proposed identification and the manual identification performed by experts, correlations of 93.4%, 92.5%, and 94.5% were obtained, respectively, for whiteflies, aphids, and thrips.

Rustia and Lin [21] proposed an IoT-based remote monitoring system for pests on yellow sticky traps and developed image processing and ML algorithms. The images were divided into four regions and equalised using a histogram based on the brightness adjustment obtained from reference images. A k-means grouping is applied in each image converted into a colour space. The insects and the background are black or white in the image obtained. In the end, the insects can be classified and counted. The method effectively acquired accurate and automatic pest counts, obtaining an average accuracy of 98%. Classifying pests in corn, soybean, wheat, and canola is difficult due to the similarity between insect species; Xie et al. [29] proposed an insect recognition system using multiple task sparse representation and multiple kernel learning techniques. It was shown that their method performs well in classifying insect species, outperforming other methods. Ebrahimi et al. [31] and More and Nighot [30] implemented an approach based on the support vector machine for classifying and identifying pests. Most of these techniques showed good performance; however, they are only recommended for particular situations and are not adaptable to other scenarios because these techniques cannot make intelligent decisions.

DL can learn and make decisions using algorithms inspired by the human brain, making it possible to adapt to more complex environments [19,32]. In recent years, DL has started to be applied in the field of agriculture as well. For example, DL algorithms could be used to analyse images of crops to identify pests or diseases or to monitor the growth and health of plants. This information could be used to optimise irrigation or fertilisation or to take other actions to improve crop yields. DL could also be used in other areas of agriculture, such as in analysing data from field sensors. In other words, several DL applications have emerged to solve challenges in the agricultural context. Automatic recognition of pest images has become one of the leading research points in DL [24].

The object detection task can be associated with two important concepts: (1) object classification; and (2) detection, as shown in Figure 2. Classification is the assignment of a class to the principal object in the image. Object detection consists of the object localisation and classification of multiple objects in an image [33]. This technique uses rectangular bounding boxes to locate and classify the categories of the objects [34]. Object detection is an important area of computer vision. It is crucial in many applications, such as video, medical images, vehicles, pedestrians, and face detection.

There are two significant groups of detectors: one-stage detectors and two-stage detectors. One-stage detectors solve the detection task by directly predicting object categories and regression object locations [33], such as You Only Look Once (YOLO) [35] and the Single Shot Multi-Box Detector (SSD) [36]. This method does not require the region proposal process, so the detection is faster; however, the precision is generally lower than that of the two-stage object detector architecture. Two-stage detectors initially extract the regions of interest from the input image and then classify and redefine the location of the object through the first proposed regions; examples are Region-based Convolutional Neural Networks (R-CNN) [37], Fast R-CNN [38], Faster R-CNN [39], Mask R-CNN [40] and Cascade R-CNN [41]. The most significant advantage is the high precision, and the disadvantage is the high detection time [34].

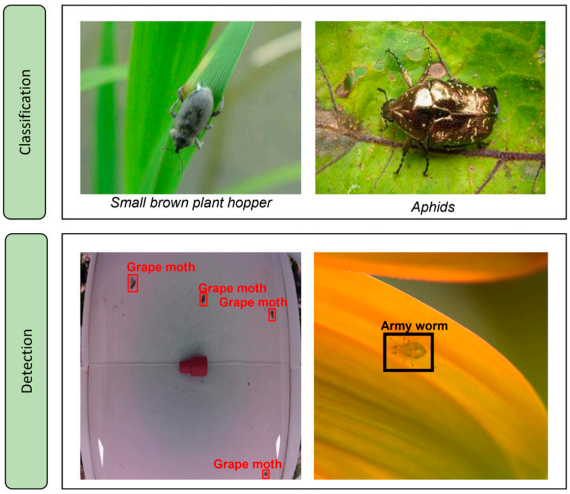


Figure 2. Examples of insect classification and detection tasks. The classification example of of small brown plant hopper and aphids on plant images. The first example of a detection task is the detection of grape moths on pheromone trap, image provided by [22]; the second example is the detection of army worm on plants images. Images with small brown plant hopper, aphids and army worm were adapted from the public dataset IP102 [42].

3. Materials and Methods

3.1. Research Questions

In this study, three essential research questions were considered, which are the following:

- (RQ1) What are the methods that obtain better mean average precision (mAP) for the task of insect detection?
- (RQ2) What dataset variables have the most significant influence on detection?
- (RQ3) What are the main challenges of and recommendations for automatically detecting insects?

3.2. Inclusion Criteria

The study of methods of automatic detection of insects in traps was carried out considering the following criteria: (1) studies that apply DL techniques for insect classification; (2) studies that apply DL methods for automatic insect counting; (3) studies that apply DL methods for insect detection; (4) studies published between 2016 and 2022; and (5) studies written in English.

3.3. Search Strategy

This systematic review consisted of studies that met the inclusion criteria in the following electronic databases: IEEE Xplore, Scopus, MDPI, ScienceDirect, SpringerLink, and PubMed. The search terms used were “automatic detection of insects”, “insect traps”, “classification”, and “DL”. The studies were analysed to identify the various DL methods of automatic insect detection. The search was conducted on 5 February 2022.

3.4. Selection of the Papers and Extraction of Study Characteristics

Ninety-two studies collected in these databases were identified. After analysing all the studies, the selection was made for inclusion in the research, as shown in Figure 3. Of the ninety-two articles initially identified, two were duplicates. After screening, considering the title, keywords, and abstract, thirty-three articles were discarded because they did not cover insect detection and classification. Then, a complete study was carried out considering the inclusion criteria; consequently, twenty-one articles were excluded. Thus, the remaining thirty-six articles were analysed and included in this survey. Of the selected searches, twelve were for the classification task and twenty-four were for the detection task.

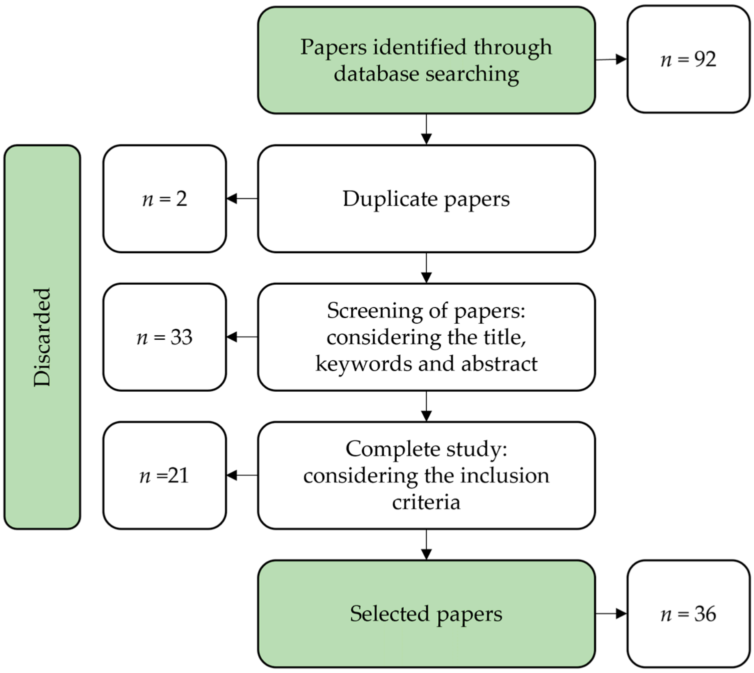

Papers identified through database searching
n = 92
Duplicate papers
n = 2
Screening of papers: considering the title, keywords and abstract
n = 33
Complete study: considering the inclusion criteria
n =21
Discarded
Selected papers
n = 36

Figure 3. Flow diagram of the selection of the papers.

4. Results

The articles selected were divided into three topics: (1) the classification of insects with DL; (2) the detection of insects with DL; and (3) the challenges and recommendations found. For the first topic, the studies that described pest classification were briefly analysed, allowing the identification of the methodologies and architectures, the size of the dataset, and the results obtained. For the second topic, we analysed the papers that solved the detection task. Then, the detailed analysis of eight studies, considered interesting and promising, was performed. Finally, for the third topic, challenges and recommendations were presented.

For better organisation, the studies were separated into three tables. Table 2 summarises the studies focusing on the classification task, and Tables 3 and 4 focus on detection tasks. The tables show the data collected from each selected article: image scenario, the number of classes, dataset size, methods, architectures, and results. Through the detection tables, it is also possible to analyse the average inference times per image obtained in the test dataset of the studies that provided this information. To assess classification and detection performance, the results relied on accuracy and mAP as the respective metrics. These metrics were chosen based on their widespread use in evaluating classification and detection and were consistently employed across all the reviewed studies. This approach facilitated a more meaningful comparison of results across the studies.

Table 2. Study analysis of classification with image scenario on plants.

Paper	Year	Number of Classes	Dataset Size	Methods	Results (Accuracy)
[43]	2017	10	550	ResNet101	98.7%
[44]	2019	40 24 40	4263 1397 4500	CNN proposed by authors	96.8% 97.5% 95.9%
[45]	2020	10	5629	GoogLeNet—fine-tuning	94.6%
[46]	2020	2	5000	Resnet50—fine-tuning	93.8%

Table 2. *Cont.*

Paper	Year	Number of Classes	Dataset Size	Methods	Results (Accuracy)
[47]	2020	10	859	DenseNet169—transfer learning	88.8%
[48]	2020	8	1426	VGG16—fine-tuning	97.1%
[49]	2020	20	4909	CPAFNet: created by authors	92.6%
[50]	2020	15	100	ResNet34	97.8%
[26]	2021	24	1387	CNN proposed by authors	90.0%
[51]	2021	5	500	Faster R-CNN	99.0%
[52]	2021	10	3549	Resnet50—fine-tuning	95.0%
[53]	2021	1	700	YOLOv3	95.3%

Table 3. Study analysis of detection with standard detectors.

Paper	Year	Image Scenario	Number of Classes	Dataset Size	Method	Results (mAP)	Inference Time(s)
[54]	2018	In traps	7	10,000	YOLO	92.5%	0.167
[55]	2018	In traps	3	1350	Faster R-CNN	87.4%	n.a.
[56]	2018	In traps	6	2183	RetinaNet	74.6%	0.448
[32]	2019	On plants	12	3022	SSD	77.1%	0.100
[24]	2020	In traps	24	25,378	YOLOv3	58.8%	n.a.
[57]	2020	In traps	8	1716	R-FCN	83.4%	0.124
[10]	2020	In traps	14	1000	Faster R-CNN	88.8%	0.032
[58]	2020	On plants	1	687	YOLOv3	90.0%	n.a.
[25]	2020	On plants	1	4600	Faster R-CNN	94.6%	0.360
[59]	2021	In traps	1	50	Faster R-CNN	85.6%	0.078
[60]	2021	On plants	14	49,700	Cascade R-CNN	70.8%	n.a.
[61]	2022	In traps	1	4134	YOLOv5	94.7%	n.a.
[62]	2022	On plants	3	4541	Faster R-CNN	92.7%	0.016

n.a. = information not available.

Table 4. Study analysis of detection with combined/adapted methodologies.

Paper	Year	Image Scenario	Number of Classes	Dataset Size	Method	Results (mAP)	Inference Time(s)
[63]	2016	In traps	1	177	CNN with sliding window	93.1%	n.a.
[64]	2019	In traps	16	88,670	PestNet: created by authors	75.5%	0.441
[65]	2019	In traps	3	662	Segmentation + CNN	92.4%	0.145
[66]	2019	On plants	4	4400	Multi-scale CNN + RPN	81.4%	n.a.
[67]	2019	On plants	1	85	CNN + RPN	88.5%	n.a.
[68]	2020	On plants	1	2300	Segmentation + CNN	92.0%	n.a.
[69]	2021	In traps	16	88,600	Modified Faster R-CNN	83.6%	n.a.
[70]	2021	In traps	21	24,412	S-RPN	78.7%	0.045
[6]	2021	In traps	4	5173	YOLOv3 + CNN	91.0%	2.380
[71]	2021	In traps	2	1400	Modified Faster R-CNN	95.2%	n.a.
[72]	2022	In traps	24	28,000	Modified YOLOv4	71.6%	0.013

n.a. = information not available.

4.1. Classification of Insects with DL

Convolutional Neural Networks (CNN) are neural networks that follow a feed-forward pattern, where all layers connect, following the path from the input to the output of the network. CNNs are inspired by biological processes, more specifically by the organisation of an animal's visual cortex [73]. This type of neural network is often applied in image recognition and video processing, thus becoming the "state of the art" in object classification and detection problems. The disadvantage of CNNs is the need for much labelled data for feature extraction [74]. There are some CNN architectures available that are widely used.

Classifying insects is essential in many contexts and for the important premise of IPM in agriculture [4]. More than 1.02 million insect species have been described [75], making insect identification difficult and complex. Some of the applications include the classification of pests, diseases, and invasive species [76]. In Table 2, the data collected in each article on the classification of insect pests using DL are summarised. The selected studies were published between 2017 and 2021. It appears that all studies provide a solution for classification in field images on plants.

To identify the most harmful cotton pests under field conditions, Alves et al. [50] presented a real dataset containing cotton field images, with 15 classes and 100 images. All images were resized to 224 × 224; as the dataset was small, they applied data augmentation and used CNN with ResNet34 to classify major pests automatically; the method was trained on GPU NVIDIA GTX 1060 and obtained a final accuracy of 97.8%. Cheng et al. [43] proposed the use of a CNN with ResNet101 to achieve pest identification with the complex background of agricultural land. The dataset contained different angles and pest poses. All these images were mirrored before being fed into the system to fully utilise CNN to double the total amount of data. For 10 classes in 550 images of agricultural pests in the complex background, an overall accuracy of 98.7% was reached.

Kasinathan et al. [26] used a public dataset with 1387 images (rescaled to the size of 227 × 227 pixels) and 24 classes in a highly complex background. First, image data augmentation techniques were applied, such as rotation, flipping, and cropping operators, and second, they applied CNN with the architecture proposed by them. The CNN model proposed contains five convolutional layers, three max-pooling layers, a flatten layer, a fully connected layer, and a softmax output layer. The authors' methodology was able to reach 90.0% accuracy. With the purpose of classifying insect species in three publicly available insect datasets, Thenmozhi and Srinivasulu [44] proposed an efficient deep CNN model. The CNN architecture was constituted of six convolutional layers, five max-polling layer, one fully connected layer, and the output layer with softmax. Data augmentation techniques were also applied to avoid network overfitting. Deep learning models were implemented using the Matlab2018a framework, utilising NVIDIA Quadro K2200 GPU. The highest classification accuracies of 96.8%, 97.5%, and 95.9% were achieved in the proposed CNN model for insect dataset 1 (40 classes), insect dataset 2 (24 classes) and insect dataset 3 (40 classes), respectively. All images were resized to 227 × 227 pixels. Wang et al. [49] proposed a new model called CPANet that includes four convolution layers, six max-pooling layers, three inception modules, one average pooling layer, one fully convolution layer, and an output layer with softmax. The dataset used contains 20 classes in 4909 images. Before training, the data was enhanced by image processing methods such as inversion, rotation, scaling, and Gaussian noise addition. The authors used standard architectures such as VGG, InceptionV3, and ResNet50 and compared them with the proposed model. All experiments were trained, validated, and tested using GPU Nvidia GTX 1080Ti. Their approach achieved the best accuracy, reaching 92.6%.

The success of DL depends in part on the amount of data. Sometimes the available data are scarce and private, or the costs associated with their acquisition or annotation are very high. In these situations, it is common to use transfer learning [77]. Transfer learning consists of using the knowledge learned for a task in each domain to improve the learning of another domain in another task [46]; i.e., a network is pre-trained on a large dataset, such as ImageNet [78] or MS COCO [79], and then applied to the dataset that we intend to train [77]. If the source dataset is large and complete, the learned features can be useful for the problem we want to solve [11].

There are two ways to use a pre-trained network: (1) fixed feature extraction and (2) fine-tuning [11]. Fixed feature extraction consists of removing the fully connected layers; that is, the convolutional layers of the pre-trained network are froze n and a new classifier is added. Considering the extracted resources, the classifier is trained from scratch [80]. Fine-tuning consists of replacing and training the classifier that was added to the pre-trained network, and tuning part of the pre-trained network kernels through backpropagation [46]. Normally, the initial layers do not change, as they contain more generic resources, while the later layers become more specific to our dataset, so they are adjusted by backpropagation [77].

To recognise ten types of pests present in rice plantations, Malathi and Gopinath [52] used fine-tuning and fixed feature extraction with several standard architectures. The dataset consists of 3549 images (resized to 227 × 227 pixels) of 10 pests that affect rice plantations. The ResNet50 fine-tuning model reached a better accuracy (of 95.0%) than the other models. Still, for the classification of diseases and pests in rice plants, Rahman et al. [48] were able to reach an accuracy of 97.1%. These techniques were used in the dataset with eight different species of pests and contained 1426 images. All the images were been resized to the default image size of each architecture before working with that architecture. Everton Castelão Tetila et al. [46] analysed the performance of InceptionV3, Resnet50, VGG16, VGG19, and Xception for different fine-tuning and fixed feature extraction strategies on a dataset composed of 5000 images and 2 classes, captured under field conditions. They trained all experiments on GPU NVIDIA GTX1070 and showed that architectures trained with fine-tuning have higher accuracy, reaching an accuracy of 93.8% for Resnet50 fine-tuning. Li et al. [45] presented a method to classify 10 common pest species; a fine-tuning GoogLeNet model was proposed to deal with the complex backgrounds presented. The approach was conducted on four Titan X 12 GB GPUs and made it possible to reach an accuracy of 94.6%.

Pattnaik et al. [47] applied transfer learning with the different pre-trained models for pest classification in tomato plants. The dataset was composed of 859 images categorised into 10 classes. The best performance was obtained using the DenseNet169 model (88.8% accuracy).

Chen et al. [53] and Karar et al. [51] used YOLOv3 and Faster R-CNN, respectively, but only for the classification. To classify *T. papillosa* in the orchard, Chen et al. [53] applied YOLOv3 only as a classifier on a dataset composed of 700 images of *T. papillosa*. The input image resolution was 416 × 416 pixels. Data augmentation and the parameters were adjusted to improve the model's learning rates. Their methodology was trained on GPU NVIDIA RTX 2070 and reached an accuracy of 95.3%. Karar et al. [51] tested several detectors, such as Faster R-CNN and SSD, in a dataset with 500 images (with size of 224 × 224 pixels), for classifying aphids, cicadellidae, flax budworms, flea beetles, and red spiders. All detectors were trained, validated, and tested on GPU NVIDIA GTX1080. The Faster R-CNN with the InceptionV2 architecture presented an overall accuracy of 99.0% for all pests tested.

Regarding the results obtained, the approach presented by Pattnaik et al. [47] achieved the lower accuracy, and the one presented by Karar et al. [51] revealed the most performant (99%) accuracy. However, as the authors used different databases, direct comparisons may be unfair. Therefore, we analysed the impact of the number of classes and the size of the dataset on the results obtained. Comparing the best and worst results, the Karar et al. [51] method was applied to a database with 5 classes and 500 images. The Pattnaik et al. [47] method was used in a database with 10 classes and 859 images. In other words, the dataset used by Pattnaik et al. [47] has a greater variability of insect species, with the highest number of classes, which makes it more difficult to classify. These findings suggest that the number of classes is significant, but it can be difficult to determine the correct number of classes, especially when the classes are not well separated and are unbalanced.

4.2. Detection of Insects with DL

The detection of insect pests is an essential task in SPM and can provide farmers with a helpful decision-making tool [7]. Effective detection of insect pests improves the accuracy of applied amounts of pesticide, which can have a significant economic and environmental impact [5].

Twenty-four studies that solve the issue of detecting insect pests with DL were selected. The selected studies were published between 2016 and 2022. About 66.7% of the research covers the detection of insect pests in traps, and 33.3% covers the detection directly in plants. Several different methodologies were proposed that can be divided into two groups: (1) standard detectors; and (2) combined/adapted methodologies. Standard detectors refer to architectures previously proposed by other authors, such as YOLO, Faster R-CNN, SSD, and others. The combined/adapted methodologies include modified architectures, adapted architectures, and a combination of several different methods.

4.2.1. Standard Detectors

Table 3 summarises the data collected in studies that used standard detectors. Chen et al. [21], Wang, Q. et al. [24], Yun et al. [61], and Zhong et al. [54] showed in their experiments that the YOLO, YOLOv3, and YOLOv5 architectures were the ones with the best performances. Butera et al. [62], He et al. [25], Hong et al. [59], Nieuwenhuizen et al. [55], and Ramalingam et al. [10] applied the Faster R-CNN architecture to different datasets and showed that this was the one that showed the best performance. He et al. [32] and Wang et al. [60] proposed an approach with SSD and Cascade R-CNN, respectively. Shi et al. [57] and Sun et al. [56] proposed a methodology often used to solve the small detection task, considering the challenges of the detection of small insects, like R-FCN and RetinaNet, respectively. Since the focus of this study is the use of DL to detect insects, five studies using standard detectors were selected. These studies are analysed in detail below.

He et al. [25] proposed a method for detecting the brown rice leafhopper. The algorithm consists of two layers based on faster R-CNN. The first layer seeks to identify the target of the image; that is, it aims to identify the plant. The second layer aims to detect the brown planthopper tested with Faster R-CNN with the VGG16 and ZF networks, showing that the VGG16 network showed the best results. The dataset contained 4600 images in the rice plantation in a natural environment. The training, validation, and test were set up with Nvidia GeForce GTX 1060 GPU. The proposed model obtained an average precision (AP) of 94.6% with an inference time per image of 0.36 s. Compared to the detection results using only one Faster R-CNN network and the application of the two networks, the detection with two layers showed better results. YOLOv3 was also tested and compared with their proposal. The results showed that the overall performance of their model was better than the YOLOv3 algorithm.

Wang, Q. et al. [24] provide a standardised dataset on traps for multiple agricultural pest targets. This database, called Pest24, consists of 25,378 high-resolution images with 24 major pest classes specified by the Chinese Ministry of Agriculture. They applied several state-of-the-art object detection methods, Faster R-CNN, SSD, YOLOv3, and Cascade R-CNN. For each technique, they initially used the default settings of their hyperparameters, and all experiments were trained on a Linux server with Nvidia Titan X (Pascal) GPU and 128 GB memory. Then, they tried different hyperparameter values for the YOLOv3 method, which showed the best results. The k-means clustering algorithm was used to optimise the parameter’s scaling range. The backbone of this method was Darknet-53. YOLOv3 obtained a mAP of 58.8%, proving to be the model that worked the best in detecting the twenty-four species of insects. Given the size of the dataset and the high number of classes, the authors considered adherence to objects, pest similarity, pest density, relative scale, and colour discrepancy as essential factors in the detection task. The relative scale is the factor that exerts the most significant influence on the AP of detection, and the colour discrepancy has the least significant impact.

Nieuwenhuizen et al. [55] presented a methodology to detect and count whiteflies, macrolophus bugs, and nesidiocoris bugs in sticky traps. The dataset contained 1350 images of high resolutions captured under controlled light conditions in two different greenhouses. The Faster R-CNN method with inception Resnetv2 obtained an 87.4% mAP. The model was trained in Nvidia 1080Ti GPU. The counting task results obtained were compared with those obtained by traditional counting; the correlation was greater than 0.95. However, they state that the quality of the data and annotations present in the images influenced the classification results.

Hong et al. [59] developed algorithms that detect and count *Matsucoccus thunbergianae* from pheromone trap images. The authors collected 50 images in the laboratory. The resolution of the images is 6000 × 4000 pixels, and the insect's average size is only 60 × 60 pixels. The images were cropped, with two different dimensions, 12 × 8 cropping and 6 × 4 cropping, to solve the problem of dataset dimension and of the scale of the insect to the image. In the cropped image, the insect had a larger size relative to the image size than in the uncropped image, so it was also possible to increase the number of images in the dataset. To compare and verify which architecture had the best performance, they trained a the Faster R-CNN with Resnet101, EfficientDet D4, Retinanet50, and SSD Mobilenetv2 architectures for the two chopped databases. The dataset with a 12 × 8 crop had better AP because object size relative to image size increased. Quadro RTX-6000 GPU was used for the training, validation, and testing. The model that obtained the best results was Faster R-CNN, with an AP of 85.6% for an IoU of 0.5 and an inference time per image of 0.078 s. The model that had the shortest inference time was SSD, but the detection results were not as good as those obtained by Faster R-CNN.

Shi et al. [57] proposed an architecture based on the R-FCN method to detect eight species of insects that may be present in stored grains. The dataset used is constituted by dataset1 and dataset2; dataset1 was raised in a laboratory environment (in traps) and had 1716 images, and dataset 2 has 784 images and was created to simulate the actual situation (in grains). The authors proposed R-FCN, an architecture like Faster R-CNN. There is only the replacement of the fully connected layers after RoI pooling, with a set of position-sensitive score maps to perform average voting. The backbone of this method was DenseNet. It used training techniques on various scales and applied the soft-NMS algorithm [81]. Faster R-CNN and YOLO were applied to compare results with their proposed method to compare results with their proposed approach. All experiments were done on two NVIDIA TITAN XP GPUs. The model with the bests results was their proposed one based on R-FCN, with which they obtained an mAP of 83.4% and an inference time of 0.124 s.

4.2.2. Combined/Adapted Methodologies

Table 4 summarises the data collected in each study that used combined or adapted methodologies. Li, W. et al. [71], Liu et al. [69] and Tang et al. [72] proposed some modifications to the original architectures, the first and second authors proposing modiciations to Faster R-CNN and the third author proposing modifications to YOLOv4. Liu et al. [64] proposed a new approach called PestNet inspired by Faster R-CNN. R. Li et al. [67] and Li, W. et al. [66] developed an approach with CNN and Region Proposal Network (RPN), the first author used a multi-scale model, training the images with different resolutions. Wang et al. [70] applied RPN with balanced sampling, the objective was to extract more detailed characteristics of the small insects. Rustia et al. [6] used YOLOv3 to spot all insects present in the image and then applied successive CNN classifiers to filter the insects detected initially. Ding and Taylor [63] ran through all the images with a sliding window and classified the insects found in each position. Martins et al. [65] and Tetila et al. [68] performed the segmentation of all insects in traps, and then in each segmented location, proceeded to classify the insect with CNN.

We selected three surveys that used modified architectures, adapted architectures, and a combination of different methods from the selected studies and made a more detailed analysis of each.

Liu et al. [64] developed a new method called PestNet. This consists of three main parts. The first stage consists of a CNN with channel-spatial attention; the objective is to extract and enhance image resources. The second stage comprises an RPN to provide the region proposals considering the resources extracted in the first stage. In the third step, the fully connected layers are replaced by the position-sensitive score map for classification and bounding-box regression. The dataset used consists of 88,670 images in traps of 16 different species. The authors experimented with the proposed methodology with different CNN architectures, such as VGG, ResNet50 and ResNet101. They compared it with other state-of-the-art methods such as Faster R-CNN and SSD. Their experiments are trained on a GeForce GTX TITAN X GPU and obtained better results with the ResNet101 backbone with a mAP of 75.5% and an inference time of 0.441 s, which surpasses the last generation methods.

To detect two species of the fruit fly, Martins et al. [65] proposed a method in which they initially applied a two-step segmentation method to segment areas with insects, species under study, or others. Generated bounding boxes for each segmented region; trained several CNNs to identify the one that obtained the greatest precision and proceeded to identify and classify each bounding box. The dataset used contained 662 sticky trap images; it got 22,479 bounding boxes after the initial segmentation. The network that obtained the best results for the insect classification task was the ResNet18, with a mAP of 92.4% and an inference time per image of 0.145 s utilizing a Nvidia Tesla T4 GPU.

W. Li et al. [71] developed a method based on Faster R-CNN, called 'TPest-RCNN', to automatically detect whitefly and thrips on the sticky trap in greenhouse conditions. The dataset contained 1400 images. The algorithm proposed has two significant differences from Faster R-CNN, the improved anchor size, and the RoIPooling design which was adjusted to focus on small objects and thus was able to obtain exact locations. The backbone network used is the VGG16. The anchor size present by the R-CNN is larger than the insect dimensions, so they adapted the anchor dimensions to the insect dimensions to solve this problem. RoIPooling has been replaced by a method the authors call RoIAlign, inspired by the Mask R-CNN architecture. RoIPooling can produce a deviation between the final and initial position of the bounding box, which may represent the wrong detection. To solve this, RoIAlign divides the proposed region into 4 × 4 pool sections. Four sampling areas are defined for each section, the centre point of each sampling area representing the sampling location. The pixel values of these points were calculated using the bilinear interpolation method. Finally, max pooling is applied for each compartment. The methodology applied by the authors was trained on NVIDIA Tesla K80 and obtained a mAP of 95.2%. The proposed model surpassed the Faster R-CNN architecture.

Regarding the results obtained, ten surveys had above 90.0%, seven had above 80.0%, and below 90.0%, six had results above 70.0%, and below 80.0%, only one study obtained results less than 70.0%. Li, W. et al. [71] was the study that had the best result with 95.2% and the study Wang, Q. et al. [24] was the one that obtained the lowest result with 58.8%. Analysing the impact of the number of classes and the size of the dataset; First, it is possible to verify that the increase in the number of classes is proportional to the mAP obtained. The example is the method of Li, W. et al. [71] that got 95.2% and used a dataset with 2 classes and the method Wang, Q. et al. [24] got 58.8% with a dataset that contained 24 classes. It's worth noting that the dataset used in [24] was unbalanced and had high similarity between species, posing challenges for DL algorithms to learn from such data.

4.2.3. Challenges and Recommendations in Insect Detection

Despite much research being developed to detect insect pests using DL, some challenges remain unsolved and affect the results obtained. We can divide the challenges into two significant groups, (1) datasets; and (2) methods of insect detection.

1. Datasets

Insects are the most biodiverse group of animals [82]. They can present some challenges related to your physical characteristics, such as size, the similarity between species, the different positions that can have in images and the different morphological characteristics of the same insect. As we know, insects are living beings of reduced dimensions. An image can have a high resolution, being represented by a large set of pixels, or it can be represented by a set of smaller pixels, having a lower resolution. Wang, Q. et al. [24] show that the relative scale, that is, the size of the insects in proportion to the image, is the factor that exerts the most significant influence on the detection task. As shown in Figure 4, the trap includes dozens of insects that are represented with little pixels. So, regular replacement of traps and increase of resolution of the image is encouraged.

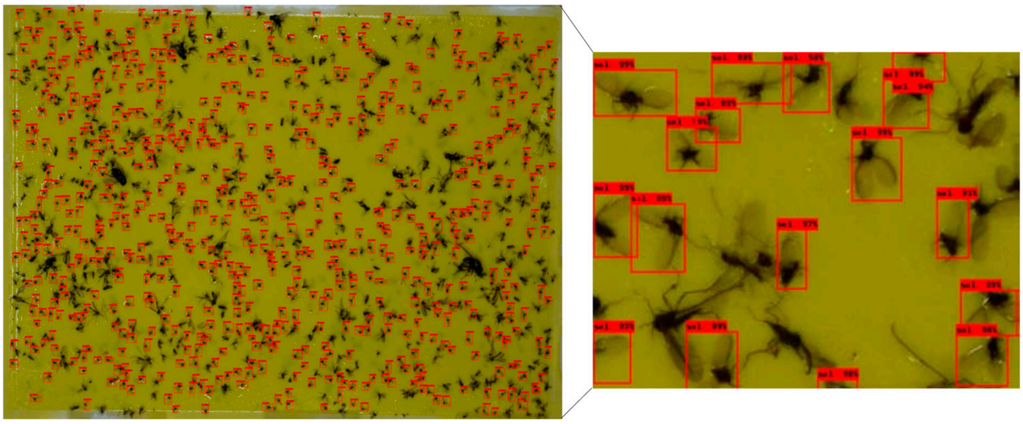


Figure 4. Example of the high amount of *M. thunbergianae* in sticky traps. The original image has a resolution of 6000 × 4000 pixels, and the size of the ground truth bounding box of *M. thunbergianae* was 60 × 60 pixels on average. Images adapted from [59].

Given the incredible biodiversity of insects, there are very similar species; and at the time of image capture, insects of the same species may be in different positions and throughout different life stages that may have different morphological characteristics. These characteristics can generate significant challenges in the task of insect detection. For example, in Figure 5, the similarity between the three species, *Armyworm*, *Bollworm* and *Yellow tiger*, are three different species with identical morphological characteristics. Additionally, two examples of different positions of the same insect in the same image can be observed. Additionally, examples of the same grape moth in different lighting conditions could be confused with another insect.

Some images collected in the field associated with SPM systems may present some challenges related to the background, lighting, and the appearance of shadows [66,83]. This challenge can be solved by carefully choosing the hour when an image is captured, choosing strategic points for the placement of SPM devices, and avoiding areas with trees and shadows. The dataset can be based on plants or traps. In plants, detection or classification can be more difficult because, in traps, the background is uniform, and in plants, the background contains different aspects that can interfere with the performance of the model. Another characteristic of these systems is the acquisition of very similar images since the collection of the image is continuous.

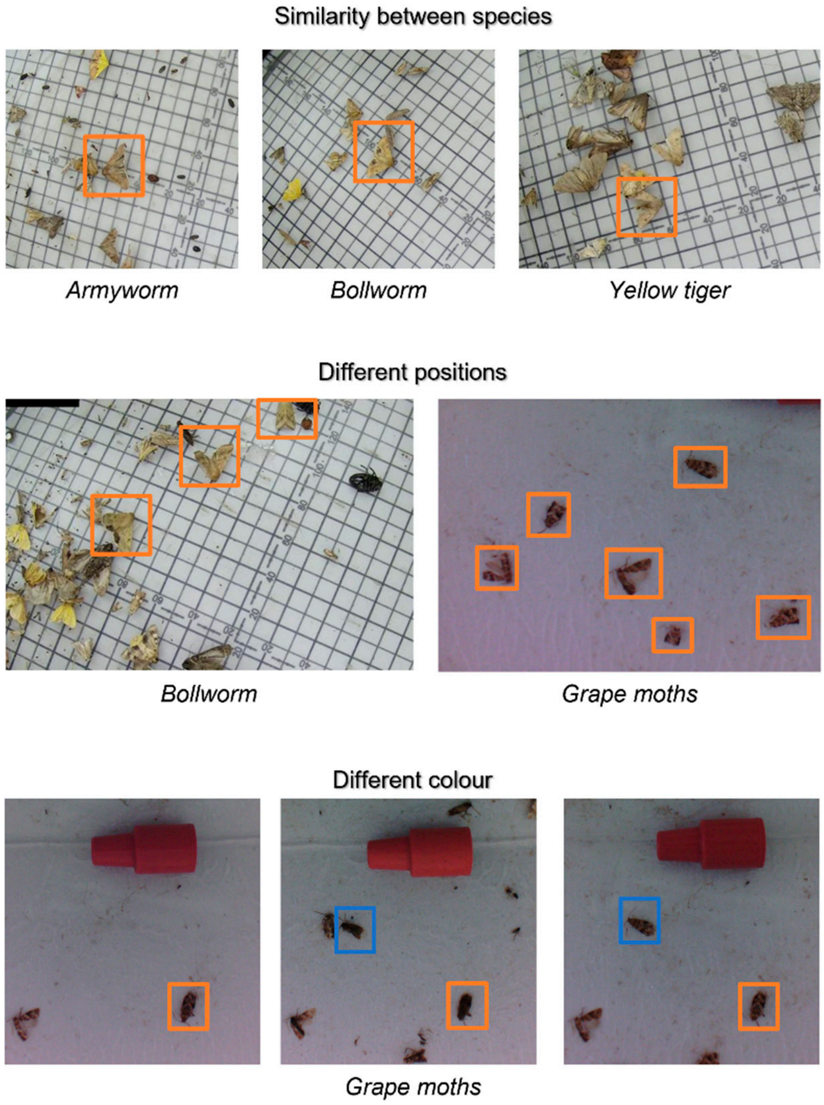


Figure 5. Examples of similarity between species, different positions of the same insect, and the different colours of the same insect. Images from the dataset Pest24 adopted from [24] and provided by [22].

In order to achieve human-level results, DL methodologies require large datasets for training models [4]. However, there is a shortage of public databases that are diverse, labelled, and of sufficient size for insect classification. Furthermore, the class distribution of insects is often unbalanced [32]. To address this challenge, it is important to encourage the collection and publication of images, as well as the development of semi-supervised methodologies [84]. Semi-supervised learning is particularly useful when all data cannot be labelled, as an effective semi-supervised model can outperform a supervised model [85].

The composition of datasets in certain ecosystems is often unbalanced due to an uneven distribution of insects, with some classes having a lack of data and others having a greater amount of data [86]. This can negatively affect the learning of classification and detection models, as the samples with greater representation may lead to the model being biased towards the majority class within the ecosystem, resulting in poor generalisation [87]. This issue of unbalanced datasets, which is commonly encountered in real-world applications, can have a significant impact on the performance of deep learning algorithms for classification and detection [87]. The conventional methods typically used for learning these models are not well-suited to imbalanced datasets, and as a result, existing classifiers tend to exhibit bias towards the majority class due to the unequal class distribution [86].

The use of synthetic data in the classes with the smallest number of images or the implementation of focal loss in the models can be very useful in solving this challenge but the effectiveness of these methods has not yet been thoroughly studied [88,89].

2. Methods of insect detection

Insect detection is a challenging task in computer vision and raises many challenges. Small objects occupy areas less than or equal to 32 × 32 pixels. While many methods used for detection give good results for medium and large objects, their performance is not so good when used to detect small objects [90].

High-resolution images are often resized so that objects of smaller pixel representations end up losing useful information to reduce the computational cost. Most of the algorithms used in the object detection task are based on CNNs. After the convolutional layers, clustering layers are applied to reduce the sampling of feature maps, thus reducing the dimensions of the image and the feature map. Due to this feature of CNN, and as small objects are represented by a few pixels, their features extracted in the initial layers are eliminated [90].

Li et al. [71] proposed the replacement of RoIPooling with RoIAlign. Part of the pixels can be removed through pooling, causing incorrect detections or even the non-detection of insects. This method showed promising results, with an mAP of 95.2%. The study in this research obtained the best results in the detection task. The Faster R-CNN architecture proposed by Ren et al. [40] has a predefined anchor that becomes too large for insect detection, affecting detection results. Thus, several authors proposed anchor optimisation; that is, they adjusted the dimensions of the anchor to the size of the insect to be detected. To solve the problem of insect size, Hong et al. [59] cut the images into two different sizes, 12 × 8 and 6 × 4, and this methodology achieved an mAP of 85.63% with Faster R-CNN. This methodology proved to be significantly better than Faster R-CNN without image cropping. A more extensive set of pixels represents the insect in the cropped image, facilitating the detection task.

There is an increased need to deal with scale problems in the task of detecting small objects. One way to address this challenge is to scale input images to many different scales and use multiple detectors for each different scale. Tong et al. [90] have identified seven methodologies that address this challenge: image pyramids in resources, a single resource map, a pyramidal resource hierarchy, integrated resources, a resource pyramid network, resource merging and resource pyramid generation, and a multi-scale merging module. As is known, DL methodologies perform better on large datasets. The same applies to small objects, which can also be improved by increasing the number of samples. Data augmentation is intended to produce additional data, through transformations, including inverting, cropping, rotation, scaling, and other techniques. The context in which the object is detected can play an important role in the performance of the methodologies. CNNs learn hierarchical representational contextual information; however, in the detection of smaller objects, there is still the possibility of enhancing the contextual information based on learning [90]. There are three different context-based methods, the local context, the global context, and the context interactives, and examples of architecture include CoupleNet [91], R-FCN++ [92], and Context-SVM [93], respectively. The methodologies used learn contextual information that can help or impact performance. The Generative Adversarial Networks (GAN) [94] model is constituted by a generating network and a discriminator. The generator learns the characteristics of the true data and generates a new sample. The discriminator compares the generated data with the real one. GANs can benefit the detection of small objects, as the generator improves the samples of small objects by increasing resolution, and the discriminator competes with the generator.

Some authors have already proposed methods that can contribute to the detection of small objects considering these crucial aspects. Shi et al. [57] proposed an architecture based on R-FCN, a multi-scale feature learning architecture. Tang et al. [72] used various data augmentation methods to increase the diversity of training samples; the resource extraction network obtains resource maps of different scales, and the feature fusion network performs feature fusion based on multi-scale feature maps. Wang et al. [70] developed an adaptive approach to learning features from different levels of the feature pyramid.

5. Discussion

Insects pose challenges in classification and detection. As we have seen, these tasks are essential in the SPM, so there is a great interest on the part of the scientific community in developing this challenge. This review analysed several methods for detecting and classifying insects using DL techniques. In general, the researched methods can be divided into two significant types of approaches: (1) standard and (2) adaptable. We consider standard approaches that implement methodologies proposed by other authors, such as the VGG, ResNet, AlexNet, Inception, and GoogLeNet architectures, and such as Faster R-CNN and YOLO detectors. Several studies have opted for adaptable approaches. It was possible to verify that there has been a growing trend, since 2019, of the development of new methodologies adapted to small objects, with a focus on insects.

The methods that present better performance in classification are the Faster R-CNN detector as a classifier with an accuracy of 99.0%, and the ResNet101 and the ResNet34 architectures with 98.7% and 97.8% accuracy, respectively. However, as the studies use different databases, it became difficult to determine the method with the best performance, so we calculated the average number of images per class. We analysed the impact of the number of classes, the size of the dataset, and the average number of images per class on the accuracy obtained by all the classification studies. Of the analysed variables, the one that showed the greatest influence on the results was the size of the dataset, although only 4.3% of the variability in the results was explained by the number of images in the dataset. The scatter plot indicates a negative relationship between the two variables with a 0.207 correlation value between the variables. Thus, we can verify that, in the analysed studies of classification, there is no significant relationship between the number of classes, the size of the dataset, and the average number of images per class in the results obtained.

In the detection, the methodologies are distributed; 54% use standard detectors in detection, and the remaining 46% apply combined/adapted methodologies. The methods that show the best detection performance are the modified Faster R-CNN, YOLOv5, and Faster R-CNN architectures, with mAP values of 95.20%, 94.70%, and 94.64%, respectively. The authors of the modified Faster R-CNN architecture proposed several modifications to the Faster R-CNN standard, such as replacing RoIPooling with RoIAlign and applying anchor optimisation. YOLOv5 is a recent architecture that has shown good results, which can be explained using a path aggregation network with a pyramid resource network, which improves the propagation of low-level features in the model and increases the accuracy of object location, even for small objects. We analysed the impact of the number of classes, the size of the dataset, and the average number of images per class on the results obtained by all the detection studies. Analysing all the variables individually, we can conclude that 63.8% of the variability of the results can be explained by the number of classes, where the scatterplot indicates a strong negative relationship between the two variables having a correlation value between variables of 0.799. Another variable that has a significant effect is the size of the dataset, with an impact of 21.1% of variability in mAP. The scatterplot indicates a negative relationship between the two variables, with a correlation value of 0.460. The remaining average number of images per class showed that it does not influence the result; i.e., the result is independent of them. This is because we must consider that the classes of the datasets are not balanced and that datasets with more classes tend to have a lower result since the classes are of very similar species, which generates confusion in the learning of the models, thus obtaining inferior results.

There are two main challenges and recommendations in insect detection using DL. First, dataset characteristics influence the performance of DL methods. Additionally, the attributes of DL methods can be adapted or improved to solve the task with better performance.

Challenge 1—dataset images:

- Insects are frequently poorly visible in datasets images.

There are many different insects, so it is necessary to represent them in the images to differentiate them accurately. A careful setup will improve data acquisition and pre-processing stages, ensuring adequate image resolution to represent insect characteristics. Attention should be given to the number of insets in the image; e.g., traps with overlayer insects will result in complex insect recognition.

- Images captured in the field using SPM systems.

Creating datasets that comprise field-captured images using SPM systems can prove to be a daunting undertaking due to various factors, such as the presence of shadows, background interference, and lighting inconsistencies. To overcome these obstacles, a range of strategies can be implemented, including careful selection of the best time of day for image capture, the strategic placement of SPM devices, and avoiding shadowy and treed areas. Several pre-processing techniques can also be employed to tackle these challenges, including shadow- or glare-free image reconstruction and the creation of an illumination invariant shadow-free image for shadow edge detection. Alternatively, simpler image processing algorithms like morphological reconstruction can also be utilised. Additionally, datasets can be based on plants or traps, each of which poses unique challenges. In particular, detecting and classifying plants can be more complex due to diverse background aspects that can impact the model's effectiveness.

- Insect classes are unbalanced in datasets images.

DL algorithms have issues learning from unbalanced datasets, significantly impacting the performance of methods of insect classification and detection. Data augmentation, where synthetic data can be used to make the dataset more balanced, and the implementation of focal loss in model training, are suggestions to help model performance and can be very useful to solve this challenge.

- Complete annotated insect datasets.

Annotation costs are frequently unbearable, preventing the annotation of the entire dataset and limiting the success of supervised DL algorithms. Using semi-supervised and domain-knowledge algorithms is recommended. Semi-supervised detection can effectively leverage unlabelled data to improve model performance.

Challenge 2—methodologies:

Small insects are represented by few pixels, so they lack information about their appearance necessary to distinguish the background or differentiate them from other classes. Furthermore, the DL algorithms developed for object detection are limited because most were developed for the detection of medium and large objects.

Recommendations:

- Multi-scale resource learning.

Resizing the input images to different scales and thus enabling learning at different scales. For this, the use of methodologies with image pyramids in resources, a single resource map, a pyramidal resource hierarchy, integrated resources, a resource pyramid network, resource merging and resource pyramid generation, and a multi-scale merging module is recommended.

- Context-based detection

Object context can play an important role in the performance of methodologies. The context in which the object is detected can help improve object detection performance, especially for detecting small objects.

- GAN based detection

The use of GAN can improve the performance of insect detection because, during training, the discriminator generates the bounding boxes and is made of classification, which is backpropagated to the generator, thus improving detection precision.

Over a period of time, insect traps can get excessively crowded with insects, which can have an adverse effect on the quality of the collected data. To maintain the integrity of the study and ensure that the data remains relevant and reliable, it is crucial to replace these traps. Although deep learning detection methods can be used to detect insects, they are limited when it comes to insect overlay. Therefore, substitution is particularly important when using traps to monitor insect populations or study insect behaviour. However, the process of replacing traps can be time-consuming and labour-intensive, which presents a challenge. To address this issue, a possible solution would be to automatically replace the trap once it reaches its maximum capacity of insects. This would eliminate the need for manual labour and ensure that the data collected is of high quality and accuracy. This approach can be achieved by integrating sensors or image recognition technology into the trap, which can detect the number of insects in the trap and signal the need for replacement. The information collected by the sensor or image recognition system can be sent to a central control unit or a cloud-based system, which can then trigger the replacement of the trap by deploying a new one. This approach can be especially useful for large-scale monitoring programs and precision agriculture, where a large number of traps are deployed in multiple locations. By implementing an automated trap replacement system, the overall efficiency of the monitoring process can be improved, and the accuracy of the data can be maintained, resulting in better decision-making and increased productivity. This will undoubtedly be one of the main challenges from the industry's point of view, and can be overcome by the collaboration between research groups and farmers.

6. Conclusions

Several DL applications have emerged for insect classification and detection in recent years, and several methodologies have been developed to improve these tasks' results. This article carried out a systematic review on automatic insect detection using DL, where thirty-six articles were analysed considering the inclusion criteria to answer three research questions:

- (RQ1) What are the methods that obtain better mAP for the task of insect detection?

The method that obtained the best result for detection was the modified Faster R-CNN architecture, where the replacement of RoIPooling with RoIAlign was proposed. However, YOLOv5 also showed high performance, and its use is recommended.

- (RQ2) What dataset variables have the most significant influence on detection?

The number of classes in the dataset is the factor that most strongly influences insect detection methods. Datasets with many classes tend to negatively influence the result since there is usually a lot of similarity between classes, and the number of images per class is unbalanced.

- (RQ3) What are the main challenges of and recommendations for automatically detecting insects?

Two key open challenges were identified that were related to automatic insect detection using DL: those associated with datasets images and methodologies. For the challenges associated with dataset images, we recommend improving data acquisition, data augmentation, focal loss, and semi-supervised and domain-knowledge algorithms, while for the methodologies, we recommend multi-scale resource learning, context-based detection, and GAN-based detection.

Incorporating advanced insect detection methods is a game-changer in the agriculture industry, and enables the vast improvement of the efficiency, quality, and sustainability of production. By leveraging these technologies, farmers can benefit from early insect detection and swift intervention, significantly reducing crop losses and optimising output. In addition to boosting productivity, these advancements can also minimise the need for harmful chemicals, decreasing chemical contamination and promoting a healthier and more sustainable farming environment.

The benefits of implementing precise insect detection methods extend beyond crop health and productivity. As farmers utilise these technologies to target specific areas of infestation, the use of pesticides and other harmful chemicals that have detrimental impacts on the environment and human health can be reduced. This, in turn, will lead to a more sustainable and eco-friendly agricultural industry, ensuring a better future for both the industry and the planet.

In conclusion, the use of advanced insect detection methods is a vital aspect of modern agriculture, with the potential to revolutionise the industry. By improving crop health, productivity, and environmental sustainability, these advancements can ensure a more profitable, efficient, and eco-friendly future for the agricultural industry.

Author Contributions: Conceptualisation, A.C.T. and J.R.; methodology, A.C.T., J.R. and A.C.; formal analysis, A.C.T. and A.C.; resources, A.C.T., R.M., J.J.S. and A.C.; data curation, A.C.T., R.M. and J.J.S.; writing—original draft preparation, A.C.T. and J.R.; writing—review and editing, A.C.T., R.M., J.J.S. and A.C.; supervision, R.M., J.J.S. and A.C.; project administration, R.M., J.J.S. and A.C.; funding acquisition, R.M., J.J.S. and A.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by National Funds by FCT (Portuguese Foundation for Science and Technology), under the projects UIDB/04033/2020, UIDP/04033/2020, and LA/P/0063/2020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	artificial intelligence
AP	average precision
CNN	Convolutional Neural Networks
DL	deep learning
IoT	Internet of things
IPM	integrated pest management
mAP	mean average precision
R-CNN	Region-Based Convolutional Neural Networks
RPN	Region Proposal Network
SPM	smart pest monitoring
SSD	Single Shot Multi-Box Detector
YOLO	You Only Look Once

References

1. Organization of the United Nations. *The State of Food and Agriculture*; Organization of the United Nations: Rome, Italy, 2014.
2. Pereira, V.J.; da Cunha, J.P.A.R.; de Morais, T.P.; Ribeiro-Oliveira, J.P.; de Morais, J.B. Physical-chemical properties of pesticides: Concepts, applications, and interactions with the environment. *Biosci. J.* **2016**, *32*, 627–641. [CrossRef]
3. Sarker, I.H. Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Comput. Sci.* **2021**, *2*, 420. [CrossRef]
4. Li, W.; Zheng, T.; Yang, Z.; Li, M.; Sun, C.; Yang, X. Classification and detection of insects from field images using deep learning for smart pest management: A systematic review. *Ecol. Inform.* **2021**, *66*, 101460. [CrossRef]
5. Lima, M.C.F.; de Almeida Leandro, M.E.D.; Valero, C.; Coronel, L.C.P.; Bazzo, C.O.G. Automatic detection and monitoring of insect pests—A review. *Agriculture* **2020**, *10*, 161. [CrossRef]
6. Rustia, D.J.A.; Chao, J.J.; Chiu, L.Y.; Wu, Y.F.; Chung, J.Y.; Hsu, J.C.; Lin, T.T. Automatic greenhouse insect pest detection and recognition based on a cascaded deep learning classification method. *J. Appl. Entomol.* **2021**, *145*, 206–222. [CrossRef]
7. Preti, M.; Verheggen, F.; Angeli, S. Insect pest monitoring with camera-equipped traps: Strengths and limitations. *J. Pest. Sci.* **2021**, *94*, 203–217. [CrossRef]

8. Henderson, P.A.; Southwood, T. *Ecological Methods*, 3rd ed.; Oxford: Oxford, UK, 2000; ISBN 978-0-632-05477-0.
9. Nanni, L.; Manfè, A.; Maguolo, G.; Lumini, A.; Brahnam, S. High performing ensemble of convolutional neural networks for insect pest image detection. *Ecol. Inform.* **2022**, *67*, 101515. [CrossRef]
10. Ramalingam, B.; Mohan, R.E.; Pookkuttath, S.; Gómez, B.F.; Sairam Borusu, C.S.C.; Teng, T.W.; Tamilselvam, Y.K. Remote insects trap monitoring system using deep learning framework and Iot. *Sensors* **2020**, *20*, 5280. [CrossRef]
11. Chollet, F. *Deep Learning with Python*; Manning Publications: Shelter Island, NY, USA, 2017.
12. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
13. Abbas, M.; Ramzan, M.; Hussain, N.; Ghaffar, A.; Hussain, K.; Abbas, S.; Raza, A. Role of light traps in attracting, killing and biodiversity studies of insect pests in Thal. *Pak. J. Agric. Res.* **2019**, *32*, 684–690. [CrossRef]
14. Trematerra, P.; Colacci, M. Recent advances in management by pheromones of *Thaumetopoea* Moths in urban parks and woodland recreational areas. *Insects* **2019**, *10*, 395. [CrossRef]
15. Gilbert, A.J.; Bingham, R.R.; Nicolas, M.A.; Clark, R.A. *Insect Trapping Guide*, 13th ed.; Gilbert, A.J., Hoffman, K.M., Cannon, C.J., Cook, C.H., Chan, J.K., Eds.; CDFA: Sacramento, CA, USA, 2013.
16. Mendes, J.; Peres, E.; Neves Dos Santos, F.; Silva, N.; Silva, R.; Sousa, J.J.; Cortez, I.; Morais, R. VineInspector: The vineyard assistant. *Agriculture* **2022**, *12*, 730. [CrossRef]
17. Ennouri, K.; Smaoui, S.; Gharbi, Y.; Cheffi, M.; ben Braiek, O.; Ennouri, M.; Triki, M.A. Usage of artificial intelligence and remote sensing as efficient devices to increase agricultural system yields. *J. Food Qual.* **2021**, *2021*, 6242288. [CrossRef]
18. Martineau, M.; Conte, D.; Raveaux, R.; Arnault, I.; Munier, D.; Venturini, G. A Survey on Image-Based Insect Classification. *Pattern Recognit.* **2017**, *65*, 273–284. [CrossRef]
19. Gutierrez, A.; Ansuategi, A.; Susperregi, L.; Tubío, C.; Rankić, I.; Lenža, L. A benchmarking of learning strategies for pest detection and identification on tomato plants for autonomous scouting robots using internal databases. *J. Sens.* **2019**, *2019*, 5219471. [CrossRef]
20. Saranya, K.; Dharini, P.; Monisha, S. Iot based pest controlling system for smart agriculture. In Proceedings of the 2019 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 15–16 November 2019; ISBN 9781728112619.
21. Rustia, D.J.A.; Lin, T.T. An IoT-based wireless imaging and sensor node system for remote greenhouse pest monitoring. *Chem. Eng. Trans.* **2017**, *58*, 601–606. [CrossRef]
22. Morais, R.; Silva, N.; Mendes, J.; Adão, T.; Pádua, L.; López-Riquelme, J.A.; Pavón-Pulido, N.; Sousa, J.J.; Peres, E. MySense: A comprehensive data management environment to improve precision agriculture practices. *Comput. Electron. Agric.* **2019**, *162*, 882–894. [CrossRef]
23. Guo, Q.; Wang, C.; Xiao, D.; Huang, Q. An enhanced insect pest counter based on saliency map and improved non-maximum suppression. *Insects* **2021**, *12*, 705. [CrossRef] [PubMed]
24. Wang, Q.J.; Zhang, S.Y.; Dong, S.F.; Zhang, G.C.; Yang, J.; Li, R.; Wang, H.Q. Pest24: A large-scale very small object data set of agricultural pests for multi-target detection. *Comput. Electron. Agric.* **2020**, *175*, 105715. [CrossRef]
25. He, Y.; Zhou, Z.; Tian, L.; Liu, Y.; Luo, X. Brown rice planthopper (*Nilaparvata lugens stal*) detection based on deep learning. *Precis. Agric.* **2020**, *21*, 1385–1402. [CrossRef]
26. Kasinathan, T.; Singaraju, D.; Uyyala, S.R. Insect classification and detection in field crops using modern machine learning techniques. *Inf. Process. Agric.* **2021**, *8*, 446–457. [CrossRef]
27. Qiao, M.; Lim, J.; Ji, C.W.; Chung, B.K.; Kim, H.Y.; Uhm, K.B.; Myung, C.S.; Cho, J.; Chon, T.S. Density estimation of *Bemisia tabaci* (Hemiptera: Aleyrodidae) in a greenhouse using sticky traps in conjunction with an image processing system. *J. Asia Pac. Entomol.* **2008**, *11*, 25–29. [CrossRef]
28. Xia, C.; Chon, T.S.; Ren, Z.; Lee, J.M. Automatic identification and counting of small size pests in greenhouse conditions with low computational cost. *Ecol. Inform.* **2015**, *29*, 139–146. [CrossRef]
29. Xie, C.; Zhang, J.; Li, R.; Li, J.; Hong, P.; Xia, J.; Chen, P. Automatic classification for field crop insects via multiple-task sparse representation and multiple-kernel learning. *Comput. Electron. Agric.* **2015**, *119*, 123–132. [CrossRef]
30. More, S.; Nighot, M. AgroSearch: A web based search tool for pomegranate diseases and pests detection using image processing. In *Proceedings of the ACM International Conference Proceeding Series*; Association for Computing Machinery: Laval, France, 2016; Volume 04-05-March-2016.
31. Ebrahimi, M.A.; Khoshtaghaza, M.H.; Minaei, S.; Jamshidi, B. Vision-based pest detection based on SVM classification method. *Comput. Electron. Agric.* **2017**, *137*, 52–58. [CrossRef]
32. He, Y.; Zeng, H.; Fan, Y.; Ji, S.; Wu, J. Application of deep learning in integrated pest management: A real-time system for detection and diagnosis of oilseed rape pests. *Mob. Inf. Syst.* **2019**, *2019*, 4570808. [CrossRef]
33. Chen, Q.; Wang, P.; Cheng, A.; Wang, W.; Zhang, Y.; Cheng, J. Robust one-stage object detection with location-aware classifiers. *Pattern Recognit.* **2020**, *105*, 107334. [CrossRef]
34. Xiao, Y.; Tian, Z.; Yu, J.; Zhang, Y.; Liu, S.; Du, S.; Lan, X. A review of object detection based on deep learning. *Multimed. Tools Appl.* **2020**, *79*, 23729–23791. [CrossRef]
35. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. 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, NV, USA, 27–30 June 2016; pp. 779–788.

36. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016, Proceedings, Part I 14*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37. [CrossRef]
37. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 142–158. [CrossRef]
38. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
39. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* 2015. [CrossRef]
40. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *42*, 386–397. [CrossRef] [PubMed]
41. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
42. Wu, X.; Zhan, C.; Lai, Y.K.; Cheng, M.M.; Yang, J. IP102: A Large-Scale Benchmark Dataset for Insect Pest Recognition. In Proceedings of the Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Long Beach, CA, USA, 15–19 June 2019; pp. 8779–8788.
43. Cheng, X.; Zhang, Y.; Chen, Y.; Wu, Y.; Yue, Y. Pest identification via deep residual learning in complex background. *Comput. Electron. Agric.* **2017**, *141*, 351–356. [CrossRef]
44. Thenmozhi, K.; Srinivasulu Reddy, U. Crop pest classification based on deep convolutional neural network and transfer learning. *Comput. Electron. Agric.* **2019**, *164*, 104906. [CrossRef]
45. Li, Y.; Wang, H.; Dang, L.M.; Sadeghi-Niaraki, A.; Moon, H. Crop pest recognition in natural scenes using convolutional neural networks. *Comput. Electron. Agric.* **2020**, *169*, 105174. [CrossRef]
46. Tetila, E.C.; Machado, B.B.; Astolfi, G.; de Souza Belete, N.A.; Amorim, W.P.; Roel, A.R.; Pistori, H. Detection and classification of soybean pests using deep learning with UAV images. *Comput. Electron. Agric.* **2020**, *179*, 105836. [CrossRef]
47. Pattnaik, G.; Shrivastava, V.K.; Parvathi, K. Transfer learning-based framework for classification of pest in tomato plants. *Appl. Artif. Intell.* **2020**, *34*, 981–993. [CrossRef]
48. Rahman, C.R.; Arko, P.S.; Ali, M.E.; Iqbal Khan, M.A.; Apon, S.H.; Nowrin, F.; Wasif, A. Identification and recognition of rice diseases and pests using convolutional neural networks. *Biosyst. Eng.* **2020**, *194*, 112–120. [CrossRef]
49. Wang, J.; Li, Y.; Feng, H.; Ren, L.; Du, X.; Wu, J. Common pests image recognition based on deep convolutional neural network. *Comput. Electron. Agric.* **2020**, *179*, 105834. [CrossRef]
50. Alves, A.N.; Souza, W.S.R.; Borges, D.L. Cotton pests classification in field-based images using deep residual networks. *Comput. Electron. Agric.* **2020**, *174*, 105488. [CrossRef]
51. Karar, M.E.; Alsunaydi, F.; Albusaymi, S.; Alotaibi, S. A new mobile application of agricultural pests recognition using deep learning in cloud computing system. *Alex. Eng. J.* **2021**, *60*, 4423–4432. [CrossRef]
52. Malathi, V.; Gopinath, M.P. Classification of pest detection in paddy crop based on transfer learning approach. *Acta Agric. Scand. B Soil Plant Sci.* **2021**, *71*, 552–559. [CrossRef]
53. Chen, C.J.; Huang, Y.Y.; Li, Y.S.; Chen, Y.C.; Chang, C.Y.; Huang, Y.M. Identification of fruit tree pests with deep learning on embedded drone to achieve accurate pesticide spraying. *IEEE Access* **2021**, *9*, 21986–21997. [CrossRef]
54. Zhong, Y.; Gao, J.; Lei, Q.; Zhou, Y. A vision-based counting and recognition system for flying insects in intelligent agriculture. *Sensors* **2018**, *18*, 1489. [CrossRef]
55. Nieuwenhuizen, A.; Hemming, J.; Suh, H. Detection and Classification of Insects on Stick-Traps in a Tomato Crop Using Faster R-CNN. In Proceedings of the Netherlands Conference on Computer Vision, Eindhoven, The Netherlands, 26–27 September 2018; pp. 1–4.
56. Sun, Y.; Liu, X.; Yuan, M.; Ren, L.; Wang, J.; Chen, Z. Automatic in-trap pest detection using deep learning for pheromone-based *Dendroctonus valens* monitoring. *Biosyst. Eng.* **2018**, *176*, 140–150. [CrossRef]
57. Shi, Z.; Dang, H.; Liu, Z.; Zhou, X. Detection and identification of stored-grain insects using deep learning: A more effective neural network. *IEEE Access* **2020**, *8*, 163703–163714. [CrossRef]
58. Chen, C.J.; Huang, Y.Y.; Li, Y.S.; Chang, C.Y.; Huang, Y.M. An AIoT based smart agricultural system for pests detection. *IEEE Access* **2020**, *8*, 180750–180761. [CrossRef]
59. Hong, S.J.; Nam, I.; Kim, S.Y.; Kim, E.; Lee, C.H.; Ahn, S.; Park, I.K.; Kim, G. Automatic pest counting from pheromone trap images using deep learning object detectors for *Matsucoccus thunbergianae* monitoring. *Insects* **2021**, *12*, 342. [CrossRef] [PubMed]
60. Wang, R.; Liu, L.; Xie, C.; Yang, P.; Li, R.; Zhou, M. Agripest: A Large-scale domain-specific benchmark dataset for practical agricultural pest detection in the wild. *Sensors* **2021**, *21*, 1601. [CrossRef]
61. Yun, W.; Kumar, J.P.; Lee, S.; Kim, D.S.; Cho, B.K. Deep learning-based system development for black pine bast scale detection. *Sci. Rep.* **2022**, *12*, 606. [CrossRef]
62. Butera, L.; Ferrante, A.; Jermini, M.; Prevostini, M.; Alippi, C. Precise agriculture: Effective deep learning strategies to detect pest insects. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 246–258. [CrossRef]
63. Ding, W.; Taylor, G. Automatic moth detection from trap images for pest management. *Comput. Electron. Agric.* **2016**, *123*, 17–28. [CrossRef]

64. Liu, L.; Wang, R.; Xie, C.; Yang, P.; Wang, F.; Sudirman, S.; Liu, W. PestNet: An end-to-end deep learning approach for large-scale multi-class pest detection and classification. *IEEE Access* **2019**, *7*, 45301–45312. [CrossRef]
65. Martins, V.A.M.; Freitas, L.C.; de Aguiar, M.S.; de Brisolara, L.B.; Ferreira, P.R. Deep learning applied to the identification of fruit fly in intelligent traps. In Proceedings of the Brazilian Symposium on Computing System Engineering, SBESC, Natal, Brazil, 19–22 November 2019; Volume 2019-November.
66. Li, R.; Jia, X.; Hu, M.; Zhou, M.; Li, D.; Liu, W.; Wang, R.; Zhang, J.; Xie, C.; Liu, L.; et al. An effective data augmentation strategy for CNN-based pest localization and recognition in the field. *IEEE Access* **2019**, *7*, 160274–160283. [CrossRef]
67. Li, W.; Chen, P.; Wang, B.; Xie, C. Automatic localization and count of agricultural crop pests based on an improved deep learning pipeline. *Sci. Rep.* **2019**, *9*, 7024. [CrossRef]
68. Tetila, E.C.; MacHado, B.B.; Menezes, G.V.; de Souza Belete, N.A.; Astolfi, G.; Pistori, H. A deep-learning approach for automatic counting of soybean insect pests. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 1837–1841. [CrossRef]
69. Liu, L.; Xie, C.; Wang, R.; Yang, P.; Sudirman, S.; Zhang, J.; Li, R.; Wang, F. Deep learning based automatic multiclass wild pest monitoring approach using hybrid global and local activated features. *IEEE Trans. Industr. Inform.* **2021**, *17*, 7589–7598. [CrossRef]
70. Wang, R.; Jiao, L.; Xie, C.; Chen, P.; Du, J.; Li, R. S-RPN: Sampling-balanced region proposal network for small crop pest detection. *Comput. Electron. Agric.* **2021**, *187*, 106290. [CrossRef]
71. Li, W.; Wang, D.; Li, M.; Gao, Y.; Wu, J.; Yang, X. Field detection of tiny pests from sticky trap images using deep learning in agricultural greenhouse. *Comput. Electron. Agric.* **2021**, *183*, 106048. [CrossRef]
72. Tang, Z.; Chen, Z.; Qi, F.; Zhang, L.; Chen, S. 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), Auckland, New Zeeland, 7–10 December 2011; pp. 1348–1353.
73. Beltrão, F. Aplicação de Redes Neurais Artificais Profundas na Deteção de Placas Pare. Bachelor's Thesis, Universidade Tecnológica Federal do Paraná, Curitiba, Brazil, 2019.
74. Rodrigues, D.A. Deep Learning e Redes Neurais Convolucionais: Reconhecimento Automático de Caracteres em Placas de Licenciamento Automotivo. Bachelor's Thesis, Universidade Tecnológica Federal do Paraná, Curitiba, Brazil, 2018.
75. Zhang, Z.-Q. Animal biodiversity: An introduction to higher-level classification and taxonomic richness. *Zootaxa* **2011**, *3148*, 7–12. [CrossRef]
76. Valan, M.; Makonyi, K.; Maki, A.; Vondráček, D.; Ronquist, F. Automated taxonomic identification of insects with expert-level accuracy using effective feature transfer from convolutional networks. *Syst. Biol.* **2019**, *68*, 876–895. [CrossRef]
77. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. Convolutional neural networks: An overview and application in radiology. *Insights Imaging* **2018**, *9*, 611–629. [CrossRef]
78. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–15 June 2009; pp. 248–255.
79. Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014, Proceedings, Part V 13*; Springer: Berlin/Heidelberg, Germany, 2014.
80. Pan, S.J.; Yang, Q. A Survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]
81. Bodla, N.; Singh, B.; Chellappa, R.; Davis, L.S. Soft-NMS-improving object detection with one line of code. *arXiv* **2017**, arXiv:1406.2661. [CrossRef]
82. Stork, N.E. Biodiversity: World of insects. *Nature* **2007**, *448*, 657–658. [CrossRef]
83. Wosner, O.; Farjon, G.; Bar-Hillel, A. Object detection in agricultural contexts: A multiple resolution benchmark and comparison to human. *Comput. Electron. Agric.* **2021**, *189*, 106404. [CrossRef]
84. Ouali, Y.; Hudelot, C.; Tami, M. An overview of deep semi-supervised learning. *arXiv* **2020**, arXiv:2006.05278.
85. Sohn, K.; Zhang, Z.; Li, C.-L.; Zhang, H.; Lee, C.-Y.; Pfister, T. A simple semi-supervised learning framework for object detection. *arXiv* **2020**, arXiv:2005.04757.
86. Amarathunga, D.C.; Grundy, J.; Parry, H.; Dorin, A. Methods of insect image capture and classification: A systematic literature review. *Smart Agric. Technol.* **2021**, *1*, 100023. [CrossRef]
87. Ganganwar, V. An overview of classification algorithms for imbalanced datasets. *Int. J. Emerg. Technol. Adv. Eng.* **2012**, *2*, 42–47.
88. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 1–48. [CrossRef]
89. Cui, Y.; Jia, M.; Lin, T.-Y.; Song, Y.; Belongie, S. Class-balanced loss based on effective number of samples. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–19 June 2019.
90. Tong, K.; Wu, Y.; Zhou, F. Recent advances in small object detection based on deep learning: A review. *Image Vis. Comput.* **2020**, *97*, 103910. [CrossRef]
91. Zhu, Y.; Zhao, C.; Wang, J.; Zhao, X.; Wu, Y.; Lu, H. CoupleNet: Coupling global structure with local parts for object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2017; Volume 2017-October, pp. 4146–4154.
92. Li, Z.; Chen, Y.; Yu, G.; Deng, Y. R-FCN++: Towards accurate region-based fully convolutional networks for object detection. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, AAAI, New Orleans, LA, USA, 2–7 February 2018; pp. 7073–7080.

93. Song, Z.; Chen, Q.; Huang, Z.; Hua, Y.; Yan, S. Contextualizing object detection and classification. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011; IEEE Computer Society: Washington, DC, USA, 2011; pp. 1585–1592.
94. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.