

Research Article

A Benchmarking of Learning Strategies for Pest Detection and Identification on Tomato Plants for Autonomous Scouting Robots Using Internal Databases

Aitor Gutierrez ¹, Ander Ansuategi,¹ Loreto Susperregi,¹ Carlos Tubío,¹ Ivan Rankić,² and Libor Lenža²

¹Autonomous and Intelligent Systems Unit, IK4-Tekniker, Eibar, Spain

²Department of Chemistry and Biochemistry, Faculty of AgriSciences, Mendel University, Brno, Czech Republic

Correspondence should be addressed to Aitor Gutierrez; aitor.gutierrez@tekniker.es

Received 21 December 2018; Accepted 17 April 2019; Published 5 May 2019

Academic Editor: Eduard Llobet

Copyright © 2019 Aitor Gutierrez et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Greenhouse crop production is growing throughout the world and early pest detection is of particular importance in terms of productivity and reduction of the use of pesticides. Conventional eye observation methods are nonefficient for large crops. Computer vision and recent advances in deep learning can play an important role in increasing the reliability and productivity. This paper presents the development and comparison of two different approaches for vision based automated pest detection and identification, using learning strategies. A solution that combines computer vision and machine learning is compared against a deep learning solution. The main focus of our work is on the selection of the best approach based on pest detection and identification accuracy. The inspection is focused on the most harmful pests on greenhouse tomato and pepper crops, *Bemisia tabaci* and *Trialeurodes vaporariorum*. A dataset with a huge number of infected tomato plants images was created to generate and evaluate machine learning and deep learning models. The results showed that the deep learning technique provides a better solution because (a) it achieves the disease detection and classification in one step, (b) gets better accuracy, (c) can distinguish better between *Bemisia tabaci* and *Trialeurodes vaporariorum*, and (d) allows balancing between speed and accuracy by choosing different models.

1. Introduction

European agriculture is facing numerous challenges such as population growth, climate change, resource shortages, or increased competition. Today's challenge is to produce "more with less." Greenhouse crop production is growing throughout the world, generating 46,377€/ha across Europe. Greenhouses protect crops from adverse weather conditions allowing year-round production. Integrated crop management approaches provide better control over pests and diseases. However, the intensification of greenhouse crop production creates favourable conditions for devastating infestation that can cost a 25% of the potential income.

A pest in agriculture is defined as a population of animals that feed from crop plant tissue (Phytophagous), producing economic damage. Most pests are insects or mites. The development of a pest depends mainly on the local

weather, external insect's pressure, greenhouses design, and crop management practices. Pests can severely damage crops causing important losses: economic (loss of productivity, income, and investments), social (rural areas depopulation), and psychological (commotion and panic). Not only does the presence of pests and diseases represent a risk for the farmer that owns the exploitation, but also it represents a threat for adjacent and sometimes distant holdings. According to the FAO, one of the most important pests in greenhouse or protected cultivation is whitefly, the greenhouse whitefly being predominant (*Trialeurodes vaporariorum*). In addition, *Bemisia tabaci* is the most common pest since 1989, when it was detected for the first time in pepper in the Mediterranean region. Also, the *Tuta absoluta* is one of the most devastating pests in Solanaceae crops. It is a pest coming from South America and it can cause production losses of 80-90%.

Early pest identification is of paramount importance in terms of productivity and reduction of the use of pesticides. Eye observation methods have been used in recent years, but they are not efficient in large crops. The automation of this repetitive inspection task can be done by computer vision system in order to increase the reliability and productivity. Furthermore, endowing robotic systems with pest detection capabilities will allow developing innovative and efficient solutions for Integrated Pest Management (IPM) in crops, with robots that have the ability to navigate inside greenhouses while performing early pest detection and control tasks in an autonomous way.

The requirements for computer vision based pest detection vary from species: their location in the plants and the features at the different stages in their development cycle. There are different aspects and visual characteristics of the different pests during their development from egg stage to adult stage. In the case of the whitefly, normally, the pest is located in the under part of leaves. In contrast, *Tuta absoluta* is detected by the symptoms present in the plants. Some symptoms caused by viruses in tomato are defoliation, silver and bronze leaves, reduced leaf area, curl leaves, etc.

In addition, the integration of an automated pest detection system in a robotic platform implies limits on computation hardware. The inspection solution needs to work in real time in order to modify, if it is necessary, the autonomous robot inspection route on the greenhouse. Algorithms response time is critical in order to inspect more plants due to the robot's batteries duration. Pest detection and identification techniques with faster response time will be executed in real time on the robot hardware. Other algorithms with higher accuracy and higher response time will be executed on external servers, but the results will not be used in real time.

The work presented on this paper is focused on the most harmful pests on greenhouse tomato and pepper crops, the polyphagous. The main polyphagous pests are the sweet potato whitefly, *Bemisia tabaci* (Gennadius), the greenhouse whitefly *Trialeurodes vaporariorum* (Westwood), and leaf miner *Tuta absoluta*.

This paper presents the work performed to develop and compare two different approaches for vision based automated pest detection and identification, using learning strategies. The paper shows the work performed to select the best approach based on pest detection and identification accuracy. The first approach uses computer vision techniques for pest detection and machine learning for pest classification, while deep learning technique is used for both pest detection and classification. The comparison of both approaches is performed using a large number of pictures that are generated and labelled using realistic setups.

From a qualitative point of view, both approaches present advantages and disadvantages. Solutions based on deep learning algorithms are demonstrated to be very effective in image processing, showing high performance and good results in different research and industrial applications. The main disadvantage of the deep learning approach is its "black box" nature, making it difficult to understand the reason why a deep learning based algorithm makes a specific prediction.

On the other hand, computer vision and machine learning algorithms need fewer amounts of data and less time to train them. Deep learning model creation process needs also more amount of computational power. The objective of this paper is to provide objective and quantitative evaluation of the performance of both approaches.

The paper starts with describing the current state of the art of the automatic pest detection and identification in agriculture using computer vision, machine learning, and deep learning strategies in Section 2. It also describes some pest detection approaches using the PlantVillage [1] dataset, a public leaf disease images dataset. Due to the fact that the PlantVillage dataset does not contain images of the diseases that are of interest in this work, an internal dataset has been created which is explained in Sections 3.1 and 3.2. It is composed of pictures that are taken both in manual and automatic way. Once the dataset is generated and labelled, the computer vision approach for pest detection, the machine learning approach for pest classification, and the deep learning approach for pest detection and identification are explained in detail in Sections 3.3 and 3.4. In addition, the experimentation performed is explained in Section 4. Both techniques are compared based on the results, and the main conclusions are extracted in Section 5. Finally, some future work is envisioned in Section 6.

2. Related Work

Automatic pest identification is an active research topic in the last years. In most cases, computer vision, machine learning, or deep learning technologies are selected and used to detect plant diseases, but the comparison of the different possible techniques in the same work is not usually found; instead, normally a single approach is selected. Many works on automatic pest detection and identification are focused on a specific selected technological approach [2, 3], but different technological solutions are not tested.

Computer vision and object recognition made huge advances in last years. Large Scale Visual Recognition Challenge (ILSVRC) [4] based on the ImageNet public dataset has been used as benchmark for different visualization-related problems in computer vision, including object classification and object identification. Previously, the traditional approach for image classification tasks has been based on features detection algorithms, such as DoG, Salient Regions, SURF, SIFT, MSER, etc. [5]. When features are extracted, some learning algorithms are used with these features. The performance of the approaches depends on predefined features. Feature engineering itself is a hard process which needs to be revisited when the problem and the dataset change. This problem happens in all attempts to detect plant diseases using computer vision as they trust in hand-engineered features and image enhancement algorithms. Manual feature extraction issue can be solved using deep learning techniques, since feature extraction process is done automatically.

Latest developments in machine learning and deep learning allow drastically improving the accuracy of object recognition and detection. On the one hand, machine learning methods have been applied as solution for disease

detection [6]. Many of these methods have been applied in agricultural research projects, for example, Artificial Neural Networks (ANNs), Decision Trees, K-means, and K-nearest neighbours. Support Vector Machines (SVMs) is one of these approaches that have been used extensively in disease detection field. On the other hand, deep learning is a new trend in machine learning and it improves the state of the art in different fields, such as the ability to deal directly with images without using manual extracted features. Both deep learning and machine learning can improve computer vision accuracy as it has been demonstrated in several applications. While machine learning makes informed decision due to what it has learned from input data, deep learning can learn and make intelligent decision using structures algorithms in layers. In 2012, deep learning networks achieved a top-5 error of 16.4% for the classification of images into one thousand different categories [7]. Advances in deep convolutional neural networks reduced the error to 3.57% [8–10].

Different approaches to disease detection and classification via machine learning in tomato crops have been analysed. First, using RGB images and different machine learning algorithms (SVM, linear kernel, quadratic kernel (QK), radial basis function (RBF), multilayer perceptron (MLP), and polynomial kernel), tomato yellow leaf curl disease (TYLCD) is detected [11]. This approach obtained accuracy of 90 % in average. Second, using both RGB images and spectral reflectance to detect and quantify tomato leaf miner, SVM (support vector machine) algorithms are used [12]. Third, using also SVM algorithms and thermal and stereo visible light, tomato powdery mildew fungus *Oidium neolycopersici* is detected [13]. Fourth, powdery mildew on tomato crops is identified using self-organizing map (SOM) and Bayesian classifiers and RGB images [14]. They only use 138 pictures in this work, a small number of pictures to get variability in the dataset.

Most of the classifiers in disease detection and classification via machine learning were trained with small datasets, focusing on the extraction of image features to classify the leaves. A large, labelled, and verified dataset of images of diseased and healthy plants is necessary in order to develop an accurate image classifier. Until very recently, it was not available any dataset with these features. To solve this problem, the PlantVillage project has begun collecting and labelling tens of thousands of images of healthy and diseased crop plants. PlantVillage dataset is used as the approach of training deep learning models for different crop disease diagnosis. It is used in most of the latest research projects related with pest detection and deep learning. It contains 18,160 tomato pictures of bacterial spot, early blight, late blight, leaf mold, septoria leaf spot, spider mites, two-spotted spider mite, target spot, tomato yellow leaf curl virus, and healthy leaves.

Unfortunately, it is not possible to use PlantVillage dataset on this work because it does not contain images corresponding to the three diseases this work is targeting. To overcome this gap, the project will generate its own dataset of *Trialeurodes vaporariorum*, *Bemisia tabaci*, and *Tuta absoluta* pictures on an automatic and manual way. Generated pictures will be labelled using an image annotation tool. The

time and effort needed to create and label these datasets is one of the bottlenecks for using deep learning technique. PlantVillage dataset is composed of cropped pictures of leaves with different diseases. The dataset can be used for pest classification, but it cannot be used for pest detection as an insect or egg level. The dataset constructed for this work is labelled at insect and egg level, it is composed of original pictures of plants (not cropped pictures), and it can be used for object detection using deep learning. It also contains pictures with tomato diseases that are not available in the PlantVillage dataset.

Related to deep learning for tomato diseases classification and symptoms visualization, [15] presents a convolutional neural network (CNN) as a learning algorithm that uses images directly. They use deep learning as a method for classifying plant diseases, focusing on images of leaves. The model is using visualization methods to understand and localize regions in the leaf. The model is reaching a 99.18% of accuracy. This work is using the PlantVillage dataset, and it is using cropped pictures of leaves. Disease is detected at leaf level, but not at insect or egg level. In [16], deep learning Convolutional Neural Networks (CNNs) have been used to detect crop leaf disease where the classification of 26 diseases in 14 crop species in 54,306 images of PlantVillage dataset occurs, using two popular CNN architectures (AlexNet and GoogLeNet). The deep learning approach proposed in this work is focused on the detection and identification of the disease using an object detection framework. Other deep learning models are based on Faster RCNN (Region-Based Convolutional Neural Network) and SSD (Single Shot Detector).

3. Approach

The work described in this paper is focused on the comparison of two pest detection and identification techniques. Selected technique will be implemented on a pest detection autonomous scouting robot. As the selected approach for pest detection and identification is based on learning algorithms, it is necessary to generate and label a dataset with infected leaves with *Bemisia tabaci* and *Trialeurodes vaporariorum* insect and eggs. Due to the difficulty to generate *Tuta absoluta* pests at egg stage and insect level, this first approach is focused on *Bemisia tabaci* and *Trialeurodes vaporariorum* pictures. The generated dataset is composed of manually and automatically obtained pictures. Each technological solution contains different steps and algorithms in order to detect any pest on a plant and to classify the detected pest that are explained in the following subsections.

3.1. Dataset Generation. As it has been explained in Section 2, a dataset of infected tomato plants is needed in order to generate machine learning and deep learning models.

The quality of the dataset and its labelling will impact on the accuracy of the generated models. On the one hand, manual pictures are taken in order to have pictures with defined features inside the cultivation chamber. Figure 1 on the left side shows the cultivation chamber that is used to generate controlled tomato pests. Manual pictures are taken



FIGURE 1: Cultivation chamber and the automatic dataset generator system.



FIGURE 2: Automatic dataset generator system components.

in the cultivation chamber. On the other hand, automatic pictures are taken to increase the dataset size and variability inside the greenhouse. Figure 1 on the right side shows the automatic dataset generator system that is installed on the greenhouse. Pictures are taken on an automatic way using a microcontroller and a pan-tilt structure.

3.1.1. Manual Dataset Generator. Completely enclosed boxes have been used for the tomato cultivation on the cultivation chamber. These cultivation chambers allow eliminating both external and internal factors such as the contamination by other pathogens. Chambers have been infected with different selected diseases. Plants cultivated in Mendelu's [17] cultivation chamber are used to take pictures of different pests in a manual way. There are five cultivation boxes with tomato plants planted at different developmental stages and diverse magnitude of infestation pests.

Images are taken using the colour camera AP-3200t-PGE and monochrome camera DataCam 2016R using a standard display system connected to a PC. The selection of the captured area is performed by the worker based on his knowledge and instructions of the leader of the experiment. Different types of lenses and lighting systems are used when necessary. The focus and the shutter speed are manual, and the settings values are determined by the operator's experience. 13,047 pictures were taken in a manual way, 6,016 using monochrome camera DataCam 2016R and 7,031 using the colour camera AP-3200t-PGE.

3.1.2. Automatic Dataset Generator. The following section contains the description of the automatic dataset generator system that is installed on the greenhouse. A huge number of images are needed in order to generate models for pest detection and identification. The variability between different

images is crucial when the objective is to create an accurate model for pest detection and identification in pictures. The automatic dataset generator system will complete the dataset that is generated manually by the previous approach. The proposed solution is to take pictures every minute on the greenhouse. As a result, there are pictures with distinct angles, directions, illumination, and localization.

Automatic dataset generator is composed of two microcontroller, two cameras, two tripods, two USB flash drives, two artificial illumination systems, one pan-and-tilt structure, one tilt structure, one power source, one IP65 box, and one portable Wi-Fi 4G router (Figure 2). Both the camera and the movement structure are controlled by the microcontroller. The GigE UI-5240CP colour camera is selected as the automatic dataset generator camera. The Raspberry Pi 3 was selected to be the microcontroller due to its functionality and the price. The microcontroller is programmed to take pictures both with and without artificial lighting in different directions and angles. The interval of time between two pictures can be easily modified in the configuration file of the microcontroller. Due to the cost of sending every picture to an external server, pictures are stored on a local USB storage. Using a portable Wi-Fi 4G router with a SIM card, a local Wi-Fi network is available in order to control that the system is working correctly. Therefore, the microcontroller is accessible from other computers to update any script remotely in case it is needed.

The generated dataset contains pictures of different phases of the plant growth. It also contains pictures of both healthy and infected leaves. Distinct phases of infected leaves are in the dataset. Pictures need to be labelled with its specific disease in order to use this dataset to generate machine learning and deep learning models. Initially, the automatic dataset generator system is placed in a random location of the

TABLE 1: Labels summary.

DISEASE	TAGS
<i>Trialeurodes vaporariorum</i> eggs	25,313
<i>Bemisia tabaci</i> eggs	9,559
<i>Trialeurodes vaporariorum</i> insect	13,405
<i>Bemisia tabaci</i> insect	6,466
TOTAL	54,743

greenhouse. Then, the system is moved near the place where a disease is located.

The first phase of the generation of the dataset started the 13th of April 2018 and finished the 6th of July 2018, once all the tomato plants were removed from the greenhouse. The second phase of the generation of the dataset started the 19th of September 2018 and finished at the end of 2018.

There were different issues on the first phase of the dataset generation that are already solved in the second phase. Pictures without a correct illumination, not correctly focused, with some type of obstacle and not representative were removed from the dataset. 100,593 pictures were collected using the first microcontroller and 75,741 pictures were collected using the second microcontroller. There were many invalid pictures on this set of pictures, and finally 18,050 valid pictures were obtained from the first microcontroller and 19,692 from the second microcontroller.

3.2. Dataset Labelling. The quality of the labelling will affect the accuracy of the generated model. There are different open source and commercial solutions in order to label pictures in a faster and semiautomatic way. LabelImg [18], an open source project released under the MIT license, is currently used as the image labelling tool. It is a graphical image annotation tool developed in Python and uses QT as the graphical interface. Generated annotations are saved as XML in Pascal VOC or YOLO format.

Image labeling is a manual and time-consuming work. Due to the necessary time to label all the pictures by experts (Mendel University), a semiautomatic algorithm was developed to sort all the pictures. Pictures were labelled considering the order of the generated list. The order was established measuring the image quality, the variability, and a random selection. Image Quality Assessment (IQA) algorithm was developed to obtain the image quality score between 0 and 100. Generated dataset was divided by pictures of every day. The variability of the order of the list for labeling was obtained selecting both pictures with the higher quality score and some other pictures randomly every week. Every week the order of the list was updated with new pictures and list changes.

4,331 pictures have been labelled using image labelling tool (Figure 3), but each image contains a different number of insects and/or eggs. Table 1 shows the number of cropped images per disease. Cropped pictures for each insect and egg are used in order to generate machine learning models. 54,743 different cropped pictures are generated from the original 4,331 labelled pictures.



FIGURE 3: Image labelling tool.

3.3. Computer Vision and Machine Learning Approach (1). A well-known computer vision library used in different industrial projects is selected as the computer vision and machine learning software (HALCON) [19] in order to detect and classify pest on pictures.

Figure 4 represents the computer vision and machine learning approach step by step. On the one hand, computer vision algorithms aim at detecting possible pests on pictures. They are working together with machine learning algorithms. Computer vision output is used as machine learning input for pest classification. On the other hand, machine learning algorithms objective is pest classification. They get all possible detected diseases regions and decide whether there is a disease or not in the specific region. If a disease is detected, machine learning algorithms classify it.

3.3.1. Computer Vision for Pest Detection. The pest detection flow using computer vision can be divided into three different steps (Figure 5). Depending on the image peculiarities and features, different functions are executed at different steps. The first step is image preprocessing. If the image quality level is not good enough, a new image will be demanded. The second step is background subtraction. Finally, feature extraction algorithms are applied to select different regions with possible pests. Machine learning algorithms will decide whether it is a pest or not.

Table 2 describes the different developed functions for each step.

3.3.2. Machine Learning for Pest Classification. Two machine learning algorithms are tested in order to select the one with best accuracy: K-nearest neighbour (KNN) and Multilayer Perceptron (MLP).

One of the main differences between machine learning and deep learning approach to be considered during the system implementation is that machine learning algorithms require complex feature engineering work while, in deep learning, features between different categories are extracted in an automatic way.

Table 3 describes the features that have been selected and extracted for each object (insects and eggs) in order to generate the first version of the machine learning model.

Figure 6 describes machine learning model creation flow. Previously labelled pictures using image annotation tool and

TABLE 2: Computer vision functions for pest detection.

IMAGE PREPROCESSING FUNCTIONS	
Check image quality	Checks the image quality level in order to determine whether the image is processed or a new one is requested. Different functions or filters are applied depending on the image quality level.
Emphasize image	Enhances the contrast of the image.
Gauss filter	Smoothens an image using discrete Gauss functions.
Illuminate	Very dark parts of the image are illuminated more strongly, and very light ones are darkened.
Image enhancement	Modifies the image to improve its visual appearance. Sharpening and magnifying algorithms will accentuate pictures features.
Image restoration	Removes blur and noises from images.
BACKGROUND SUBSTRACTION FUNCTIONS	
Decompose RGB	Converts a three-channel image into three one-channel images with the same definition domain.
RGB to HSV	Transforms an image from the RGB colour space to an HSV (Hue, Saturation, and Value). HSV is defined in a way that is similar to how humans perceive colours.
Reduce image domain	Reduces the definition domain of the given image to the indicated region. It subtracts a region to a specific image.
Region segmentation	Segments images into regions of the same intensity.
Threshold image	Segments an image using a local threshold. It selects those regions in which the pixels fulfill a threshold condition.
Automatic threshold	Segments an image using thresholds determined from its histogram.
Edge detection	Detects edges using filters such as Deriche, Lanser, Shen, Canny, and Sobel.
FEATURE EXTRACTION FUNCTIONS	
Get region features	Gets different features related to colour, texture, and shape.
Connected regions	Determines the connected components of the input regions.
Select specific shape	Chooses regions according to shape feature values such as area, width, and circularity.
Count and crop regions	Counts and crops the possible regions with pests. It generates the input for the machine learning algorithm for pest classification.

TABLE 3: Machine learning model features.

FEATURE	DEFINITION
Area	Area of the object.
Circularity	Shape factor for the circularity of an object. It calculates the similarity of the object with a circle.
Compactness	Compactness of the object.
Content Length	Total length of the object.
Convexity	Shape factor for the convexity of an object. The shape factor is one if the object is convex. If there are holes, the shape factor is smaller than one.
Rectangularity	It is considered as the shape factor for the rectangularity of an object.
Elliptic axis	Calculates the main and the secondary radius of the equivalent ellipse.
Phi orientation	The orientation of the equivalent ellipse.
Anisometry	The relationship between the main and the secondary radius of the equivalent ellipse.
Bulkiness	The relationship between the anisometry and the area of the object.
Structure factor	The relationship between the anisometry and the bulkiness.
Smallest circle	Determines the smallest surrounding circle of an object. It is the circle with the smallest area of all circles containing the object.
Inner circle	Calculates the largest inner circle of an object.
Inner rectangle	Determines the largest rectangle that fits into an object.
Roundness	Calculates the distance between the contour and the centre of the area.
Sides	The number of polygon sides.
Diameter	The maximum distance between two points of the object.
Orientation	Determines the orientation of the object.
Smallest rectangle	Calculates the rectangle with the smallest area of all rectangles containing the object.

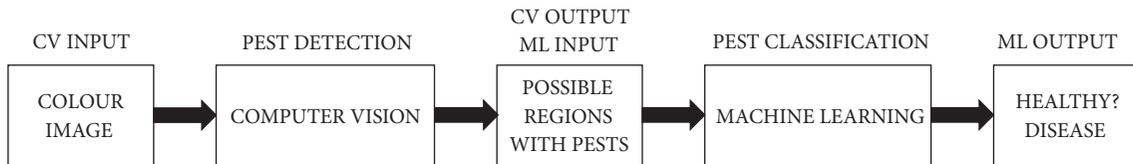


FIGURE 4: Computer vision and machine learning approach flow.

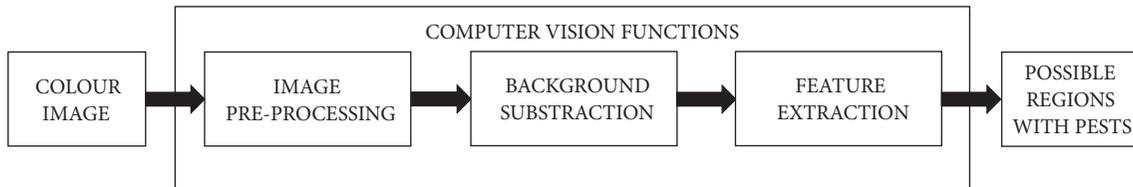


FIGURE 5: Computer vision for pest detection flow.

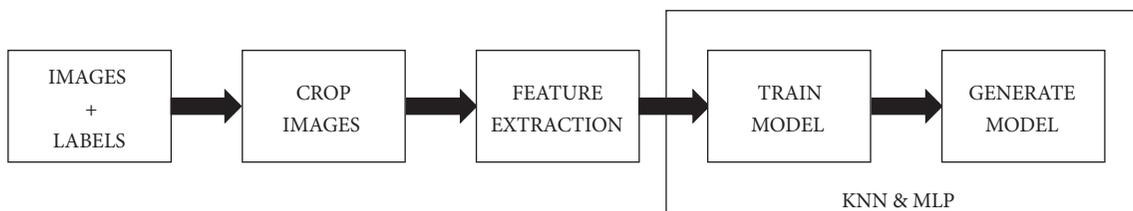


FIGURE 6: Machine learning model creation.

their labels in XML format are used as input. Images are cropped as an insect and egg level because each picture can contain more than one element. As it is represented in Table 1, labelled 4,331 pictures are converted into 54,743 pictures of different insects and eggs of *Trialeurodes vaporariorum* and *Bemisia tabaci*. Then, image background is subtracted, and different features are extracted from each insect or egg. Some of the extracted features are described in Table 3. Each feature generates a value and this number is added to the feature vector of the model. Generated vector of features and its category are added to the model. The model is considered trained when all the features of all the pictures of the training dataset are added to the model. Four different types of machine learning classification algorithms are generated.

3.4. Deep Learning Based Approach (2). TensorFlow [20] library is selected for pest detection and classification using deep learning. It is considered as the first stable framework and one of the most popular ones focused on deep learning. In particular, TensorFlow object detection library is used for the deep learning based approach [21]. This section focuses on how the data (images and annotations) is adapted for the deep learning library and the deep learning model creation flow. Figure 7 shows the deep learning flow composed of the data manipulation and model creation. First, input data is cleaned to remove not useful content, and the dataset is augmented in order to increase the variability. After that, deep learning model creation is explained in four different steps.

3.4.1. Data Manipulation. Several data manipulation steps need to be performed. First, images and annotations need to be cleaned up. Not useful pictures and annotations are removed. Second, data augmentation techniques are applied to the dataset. Minor alterations to the dataset by generating modified images can improve model accuracy. Finally, modified annotation files and real and modified images need to be converted to a specific format.

Data cleaning is the first step on data manipulation task. Data cleaning task needs to remove all unnecessary pictures and annotations from the dataset. A proper data cleaning work will improve generated model quality. It is usually a manual work, but some steps can be processed automatically. Images without annotations, annotations without images, and duplicate or irrelevant images will be removed.

Data augmentation is important to be able to detect objects at different scales [22]. Minor alterations such as rotations, viewpoint, or size to the dataset can improve model accuracy. Synthetically modified images will be new images for the neural network. Both the greenhouse's pictures and the cultivation chamber's pictures are taken in a limited set of conditions. However, the inspection task will have different conditions such as scale, brightness, illumination, and orientation. Training this type of situation can be achieved using synthetically modified images.

Data augmentation techniques can be divided into two types depending on the execution time. The first option is to execute desired transformations beforehand. This is known as offline augmentation. Using offline augmentation, the dataset

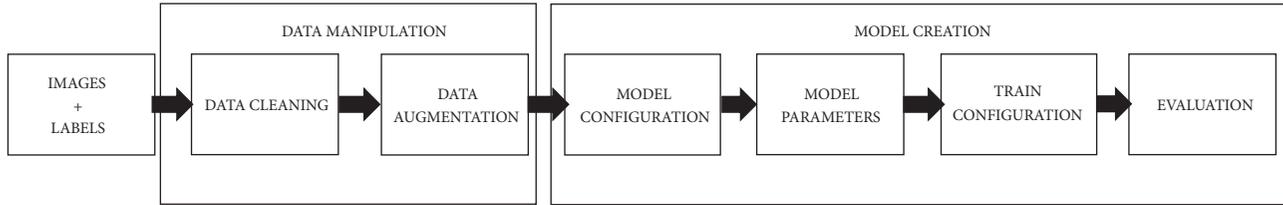


FIGURE 7: Deep learning based approach flow.

will be increased before training the model, by a factor equal to the number of transformations. Offline augmentation is normally used in small datasets. The second option is called online augmentation. Transformations are performed at the time of training the model. Some basic augmentation techniques such as crop, rotation, Gaussian noise, scale, and flip are applied.

3.4.2. Model Creation. TensorFlow pipeline configuration can be divided into four different steps (Figure 7). Firstly, the model is configured. There are different parameters defined on this step such as the number of the classes of the model, feature extractor characteristics, the meta-architecture, and the loss function characteristics. The best parameter selection will depend on the application. SSD [23] and Faster RCNN [24] are two object detection architectures in TensorFlow library. On the one hand, The SSD architecture was released by Google in 2016. It is an object detection model using a single deep neural network combining regional proposals and feature extraction. A group of boxes with distinct scales and ratios are used and applied to the feature maps. One step is only necessary to extract the feature extraction for the bounding box because the feature maps are figured out passing an image through an image classification network. Each object category got a score in every default bounding box. Boxes adjustment offsets are calculated for each box, trying to find the correct ground truth.

On the other hand, Faster RCNN deep learning architecture was developed by Microsoft based on RCNN. It uses Selective Search in order to extract different region proposals. Extracted proposals are sent to a classification network in order to use SVM to classify each region in any of the categories. According to [21], Faster RCNN architecture accuracy is better if the speed is not a problem. Faster RCNN requires at least 100 ms to analyse each picture. The correct choice of the feature extractors on Faster RCNN has a big impact on the accuracy. The feature extractor choice is not critical in SSD. SSD has the best accuracy trade-off within the fastest detectors, but it works worse for small objects compared with Faster RCNN.

Then, the configuration of the trainer step is defined. It decides the elements and parameters that should be used to train the model. The model parameter initialisation, the input preprocessing, and the SGD (Stochastic Gradient Descent) parameters are configured in this section. The learning rate and the batch size configuration are the most important configuration values in this step. Setting the learning rates

and the batch size is important to reduce overfitting. Training an object detector model from scratch takes too much time. Generated pest detection models use the weights from another model checkpoint in order to speed up the training process. The path to another object detection checkpoint is defined also in this step.

After that, the input train configuration step is defined. This step defines what dataset the model should be trained on. TensorFlow offers a set of detection models pretrained on different datasets such as the COCO dataset, the Kitti dataset, the Open Images dataset, and the AVA v2.1 dataset. It is also necessary to define the path where the label map is defined. Label map contains the information between all the defined categories and its unique identifier.

The final step is focused on configuring the evaluator. It is necessary to define the metrics that will be used for evaluation. The number of batches used for an evaluation cycle, the size of the evaluation dataset, and the selected metric to run during evaluation are defined.

It is possible to train the model both in a local PC and on the cloud. The training process will be faster if the model is trained using GPU rather than CPU. It is possible to use a computer on the cloud with a faster GPU to decrease the training time. It is mandatory to have the TensorFlow library installed and its dependencies; a set of pictures with its labels and the object detection pipeline needs to be generated in order to be able to train the model. While the model is training, it is continuously generating different checkpoints. When the training process is finished, it is necessary to select the checkpoint number in order to export and generate the model. It is common to select the biggest checkpoint number because it is the latest one. When a new model is exported, a frozen graph, the model checkpoint, the checkpoint files, the pipeline configuration file, and the exported model are generated

4. Experiments

The main purpose of this experiment is to analyse the results of pest detection and identification on plants using the combination of computer vision and machine learning, versus the deep learning technique. First, the combination of computer vision and machine learning approach will be validated using k-fold cross validation technique. Second, the deep learning approach will be validated using Average Precision (AP) which is based on the Intersection over Union (IoU) technique. Finally, the comparison between both techniques will be validated using custom metrics.

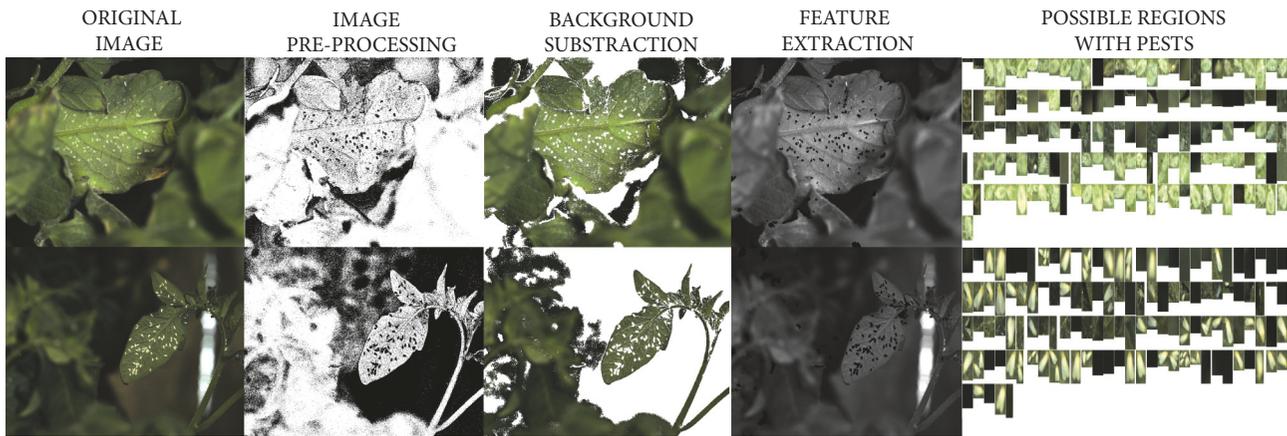


FIGURE 8: Computer vision pest detection example.

Pictures of unhealthy plants are available on the generated dataset. As it has been explained in Section 3.1, manual pictures were taken in lab conditions, and automatic pictures were taken in the greenhouse, with different perspectives and ambient light, including at night. 4331 original labelled pictures are converted into 54743 pictures of different insect and eggs of *Trialeurodes vaporariorum* and *Bemisia tabaci*.

4.1. Computer Vision and Machine Learning Approach Evaluation. The computer vision and machine learning approach evaluation is divided into two different experiments. On the one hand, the computer vision experiment will be focused on the extraction of the regions with possible pests. On the other hand, the machine learning experiment will be focused on the evaluation and the comparison of the obtained results between the different classification models.

First, computer vision experiment and results are described. The 4,331 labelled pictures are going to be used to test the computer vision algorithm. The objective of the test is to measure the number of possible regions with pests that the algorithms will extract. Due to the image labelling work, we know that from 4,331 original pictures, 54,743 pictures of different insect and eggs are extracted. The computer vision algorithm in the experiment has extracted 667,299 different pictures with possible insects and eggs. It is 12 more times than the exact number of tags. The machine learning model will classify all the cropped pictures, but it will need more time than expected for this work due to the big number of cropped pictures. On average, 154 pictures are extracted by the computer vision algorithm, when they were 12 on the generated dataset. It is possible to reduce this number of pictures, but the objective is to ensure that every possible pest on the image is going to be classified by the machine learning algorithm. If the computer vision algorithm is adjusted to reduce the number of possible pictures with pests, some insects or eggs will not be selected to inspect. As the early pest detection is very important in order to reduce the damage of any disease, we prefer more pictures to analyse and try to detect any pest on early stages.

Figure 8 shows the computer vision pest detection flow in two different pictures. The pipeline is based on the description shown in Figure 5. The flow is divided into three different steps. The first step is focused on the image preprocessing. The second step is focused on background removal. As last step, feature extraction algorithms are executed. Computer vision output is a set of pictures with possible pests and is used as the machine learning algorithm input.

Second, machine learning experiment and results are described. k-fold cross validation is used to compare the performance of the four different machine learning models on our generated dataset. The performances of K-nearest neighbour (KNN) and Multilayer Perceptron (MLP) models are measured using k-fold cross validation.

The following steps are performed to evaluate each machine learning model:

- (1) The original training dataset is divided into ten different folds or subsets. Each fold contains around 4,331 images.
- (2) For each k value:
 - (a) One fold is kept as the validation set, and the rest of the folds are kept as the training set.
 - (b) The machine learning model is trained using the training set, and the accuracy is calculated using the validation set.
- (3) The accuracy of the machine learning model is calculated by averaging the accuracies in all the cases of the cross validation.

The training dataset is divided into ten different folds. Each column represents one validation test as shown in Table 4. Every validation test is using the 10% of the images for evaluation. Each row represents one machine learning model. The accuracy of each machine learning model on each fold is measured generating an array of real categories and predicted categories. Using both arrays the confusion matrix can be generated. The average of the accuracies will be used to

TABLE 4: Machine learning accuracy results.

	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	AVR
KNN	66.11%	66.16%	66.25%	66.45%	66.83%	65.94%	66.92%	67.23%	66.36%	66.50%	66.47%
MLP	80.35%	81.12%	80.40%	80.35%	82.34%	80.93%	81.62%	81.71%	81.18%	81.27%	81.12%

choose the best machine learning model for pest detection and identification.

MLP is the machine learning pest classification model with the best accuracy. Figure 9 shows an example image using the machine learning MLP model classification to classify it inside the *Bemisia Tabaci* insect category. All the insects on the original image actually correspond to *Bemisia Tabaci* category, but not all of them are classified correctly. There are also some cropped images without an insect that are categorized into this category.

4.2. Deep Learning Approach Evaluation. 4,331 pictures are labelled using image labelling tool. 90% of labelled pictures are used for training purposes and the 10% are used for evaluating the model accuracy. The disease detection model is developed with this 90% of the pictures. We use the rest of the pictures (10%) to get the real accuracy of the pest detection and identification approach.

Determining if a disease exists in the image and the location of the insect or egg are the two different tasks to evaluate in the deep learning object detection and identification approach. This approach combines both the object classification and the localization tasks. For the object classification task, the Average Precision (AP) metric is commonly used to measure the deep learning model accuracy. The AP metric is based on the precision and the recall metrics [25]. Precision metric measures how accurate the deep learning object detection predictions are and the recall metric measures how good all the true positives are. It can be defined as the average of the maximum precisions at different recall values.

$$\begin{aligned}
 TP &= \text{True Positive}, \\
 TN &= \text{True Negative}, \\
 FP &= \text{False Positive}, \\
 FN &= \text{False Negative} \\
 \text{Precision} &= \frac{TP}{TP + FP}, \\
 \text{Recall} &= \frac{TP}{TP + FN}, \\
 \text{IoU} &= \frac{\text{Area of overlap}}{\text{Area of union}}
 \end{aligned} \tag{1}$$

On the other hand, for object localization task, the Intersection over Union (IoU) metric is used. The IoU metric will determine the accuracy of the predicted location of the object on the image. This metric is used in different object detection challenges such as the Pascal VOC [25]. It can be defined as how good the deep learning object detection prediction

is against the ground truth. Object detections results can be considered or discarded depending on an IoU threshold. An IoU value greater than 0.5 is usually considered a good prediction [25].

Table 5 describes different deep learning models that have been trained for pest detection and identification. Six different models have been trained, and each model is identified with a unique name. Each row contains the model architecture, the aspect ratio or size that the network is going to apply to the input image, the model that is reused to train the pest detection and identification model instead of starting from scratch, a variable explaining if any data augmentation technique is applied, and the number of steps on the training process.

Table 6 includes pest detection and identification results of each model (Table 5) that was previously trained. The Average Precision (AP) evaluation metric, which is based on the Intersection over Union (IoU), is used to evaluate the performance of the disease detection and identification algorithm using deep learning. The evaluation of the model is using a 0.5 IoU threshold [25] and a 0.65 confidence threshold.

Figure 10 shows the application of the six different deep learning pest detection and identification models for two different images. The first picture is full of eggs of *Trialeurodes vaporariorum*. Each model detects a different number of eggs on the first picture. Some models also classify incorrectly some eggs as *Bemisia tabaci* (blue colour labels). The second picture contains few insects, and they are detected by deep learning models.

4.3. Computer Vision and Machine Learning vs. Deep Learning Approach. Once computer vision, machine learning, and deep learning experiments have been presented and measured with their most common metrics, this section describes an experiment to check and compare the accuracy and the speed of the generated pest detection and identification models. The final goal of the work is to build a global positioning system guided autonomous robot for pest control in greenhouses. Since the robot is composed of limited batteries, the speed is another critical factor. The robot needs to inspect the largest amount of plants. Pictures will be analysed both in real time and offline (on the server side), but real-time information will be crucial in making decisions about the robot routes. The inspection solution needs to know if a disease is present in an image. If a disease is detected, the algorithm also needs to determine which type of disease is. It is not so important to detect all existing insects and eggs on a tomato leaf, but to determine if a disease is present or not in a picture.

Considering the final goal of the project, an experiment has been developed in order to determine which technique



FIGURE 9: Machine learning image classification.

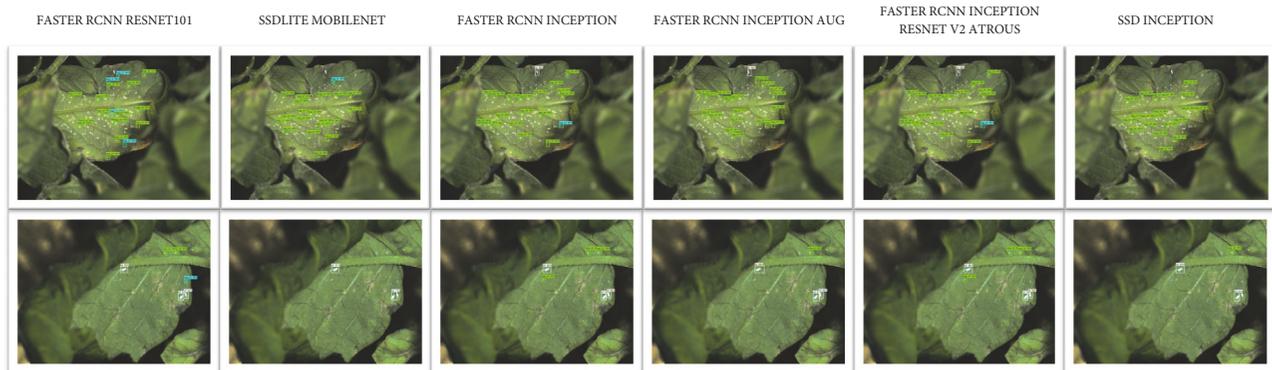


FIGURE 10: Deep learning for pest detection and identification.

TABLE 5: Tested deep learning models.

Id	Architecture	Aspect ratio resizer	Fine tune model	Data augmentation	Number of steps
M1	Faster RCNN	600x1,024	FASTER RCNN RESNET101 COCO	YES	335,736
M2	SSD	300X300	SSDLITE MOBILENET V2 COCO	YES	200,000
M3	Faster RCNN	600x1,024	FASTER RCNN INCEPTION V2 COCO	NO	177,184
M4	Faster RCNN	600x1,024	FASTER RCNN INCEPTION V2 COCO	YES	200,000
M5	Faster RCNN	600x1,024	FASTER RCNN INCEPTION RESNET V2 ATROUS COCO	YES	110,295
M6	SSD	300x300	SSD INCEPTION V2 COCO	NO	200,000

TABLE 6: Deep learning experiment results.

Metric	M1	M2	M3	M4	M5	M6
Precision Egg Trialeurodes vaporariorum@0.5IOU	0.54	0.63	0.58	0.71	0.56	0.59
Precision Egg Bemisia tabaci@0.5IOU	0.16	0.13	0.15	0.12	0.16	0.13
Precision Insect Trialeurodes vaporariorum@0.5IOU	0.72	0.63	0.65	0.74	0.69	0.69
Precision Insect Bemisia tabaci@0.5IOU	0.34	0.35	0.30	0.27	0.32	0.33
Recall Insect Bemisia tabaci@0.5IOU	0.20	0.16	0.32	0.27	0.34	0.16

shows better performance according to the requirements. The criteria for inspection technique selection will be based on accuracy and speed.

For the experiment, the following points were considered:

- (i) Machine learning KNN model T6, machine learning MLP model T4, deep learning model M4 Faster RCNN, and deep learning model M6 SSD were used.
- (ii) 100 images with possible pest of the generated 4,331 images dataset were used. The set of images was not previously used to train any model.
- (iii) 50 images with healthy tomato leaves were used.
- (iv) The algorithms' input is a picture and the output is whether the image has or not any disease. If a disease is detected, it needs to classify the disease.
- (v) For this experiment, if there is more than one disease, the disease with the largest number of insects or eggs will be the disease of the picture.
- (vi) The selected diseases are *Trialeurodes vaporariorum* eggs, *Trialeurodes vaporariorum* insect, *Bemisia tabaci* eggs, and *Bemisia tabaci* insect.

The experiment pipeline is the following:

- (i) 100 images with possible pests and 50 images without pest were selected randomly.
- (ii) Each of the images in the set of pictures with pests contains an XML file with its insects and/or eggs annotation. Using this file and a script, every insect and egg was cropped and saved. All the numbers of insects and eggs per disease were also stored in a database. Cropped pictures and the database are considered the ground truth of the experiment.
- (iii) Computer vision pest detection algorithm loads the 150 pictures and generates a set of regions to inspect by the machine learning algorithm. The number of generated pictures and the processing time were measured.
- (iv) Machine learning KNN model T6 and MLP model T4 pest classification model classifies the generated cropped pictures on different categories. All this information was stored in a database. The processing time was measured.
- (v) Deep learning pest detections M4 Faster RCNN and M6 SSD models load the 150 pictures. The number of detected pests per categories was stored in a database. All the detected insects and eggs images were cropped. The processing time was measured.
- (vi) Generated data on the database was analysed, and conclusions were extracted.

First, the computer vision experiment was performed. The computer vision test selected 27,976 different regions with possible pest from 150 images. There are 186 cropped images on average per original picture to classify by the machine learning model. The test took 430 seconds to generate all

the cropped pictures, that being 2.86 seconds on average per image.

Second, the machine learning experiment was performed. The machine learning experiment used the output of the computer vision experiment. Two different machine learning models were tested, using two different factors of confidence in order to categorize an image inside a category. In machine learning KNN and MKP models, 0.5 and 0.75 confidence thresholds were used. The KNN machine learning model took 324 seconds and the MLP model took 251 seconds to classify all the cropped pictures. On average, the KNN model took 2.16 seconds to classify all the cropped pictures per image, and the MLP took 1.67 seconds.

Third, the deep learning experiment was done. Two different deep learning models were tested. Both models were tested with 0.5 and 0.75 confidence thresholds. The Faster RCNN took 571 seconds and the SSD took 431 seconds to detect and classify all the pests. On average, the Faster RCNN took 3.8 seconds and the SSD took 3.54 seconds to analyse each image. It is important to note that the use of a GPU on the robot to classify the pictures will reduce drastically the inspection time.

All the information obtained for the machine learning and deep learning models was stored in a database in order to perform metrics later. The image name, the method used to extract the disease (ground truth, MLP-0.5, MLP-0.75, KNN-0.5, KNN-0.75, SSD-0.5, SSD-0.75, RCNN-0.5, and RCNN-0.75), number of insects *Bemisia tabaci* and *Trialeurodes vaporariorum*, number of eggs, the main disease of the image, or whether the plant is healthy were stored on the database. Executing SQL queries gives us extra information of each model results in order to update and improve it.

Table 7 shows the experiment results for each model in order to classify an image as healthy or unhealthy. Every tested model contains the percentage and the number of samples on brackets. The results are compared with the ground truth that is obtained from the labelled pictures using the image labelling tool. The best unhealthy plant disease detection is obtained with the MLP-0.5/MLP-0.75 machine learning model and the best healthy plant detection is achieved with the Faster RCNN-0.75 deep learning model. In general, the best healthy and unhealthy pest detection average rate is also obtained with the Faster RCNN-0.75 deep learning model with a success rate of 82.51%. Both machine learning MLP and KNN disease detection algorithms are getting very good unhealthy success rate result because they tend to classify many pictures as unhealthy. They often find eggs or insects in images in which there exist any diseases. As consequence, their healthy success rate is very low. Models like this that generate many false disease alarms are not useful for the purpose of the GreenPatrol project, as they will suggest the use of insecticides to stop the disease, when the disease actually does not exist, causing an overapplying of pesticides. The use of a model with a high accuracy and good balance between the healthy and unhealthy plant detection is crucial.

Table 8 describes the pest identification experiment. Some pictures contain more than one disease (e.g., eggs and one type of insect), because of that the disease with the largest number of insects or eggs is considered as the real disease

TABLE 7: Disease detection comparison rate.

	MLP 0.5	MLP 0.75	KNN 0.5	KNN 0.75	SSD 0.5	SSD 0.75	RCNN 0.5	RCNN 0.75	GROUND TRUTH
Unhealthy success rate	94,32% (83)	94,32% (83)	92,05% (81)	92,05% (81)	77,27% (68)	72,73% (64)	85,23% (75)	79,55% (70)	100% (88)
Healthy success rate	33,87% (21)	35,48% (22)	35,48% (22)	35,48% (22)	61,29% (38)	66,13% (41)	54,84% (34)	85,48% (53)	100% (62)
Success average	64,09%	64,90%	63,76%	63,76%	69,28%	69,43%	70,03%	82,51%	100%

TABLE 8: Disease identification rate.

	MLP 0.5	MLP 0.75	KNN 0.5	KNN 0.75	SSD 0.5	SSD 0.75	RCNN 0.5	RCNN 0.75	GROUND TRUTH
<i>Bemisia tabaci</i> insect success rate	0,00% (0)	0,00% (0)	6,90% (2)	6,90% (2)	55,17% (16)	51,72% (15)	75,86% (22)	65,52% (19)	100% (29)
Egg success rate	100,00% (13)	100,00% (13)	100,00% (13)	100,00% (13)	15,38% (2)	15,38% (2)	7,69% (1)	7,69% (1)	100% (13)
<i>Trialeurodes vaporariorum</i> success rate	0,00% (0)	0,00% (0)	0,00% (0)	0,00% (0)	52,17% (24)	45,65% (21)	41,30% (19)	41,30% (19)	100% (46)
Healthy success rate	33,87% (21)	35,48% (22)	35,48% (22)	35,48% (22)	61,29% (38)	66,13% (41)	54,84% (34)	85,48% (53)	100% (62)

of the image. The success rate on the machine learning MLP and KNN is perfect, but the result is misleading because the models classify many pictures as eggs. Therefore, the insect detection rate is very low in machine learning models. The experiment shows that the egg detection and identification task is very challenging. Eggs are difficult to detect and classify. The Faster RCNN-0.75 gets the best results, but the eggs identification needs to be improved.

5. Conclusions

Once computer vision, machine learning, and deep learning experiments have been explained, measured, and compared, different conclusions are extracted.

Computer vision for pest detection selects too many regions with possible disease on each image. For example, pictures with few insects or eggs also generate a big number of regions to analyse by the machine learning pest classification model. It is very slow to classify a big number of regions by the machine learning model, because of that, the combination of computer vision and machine learning approach cannot work in real time to inspect the greenhouse's plants. The balance between inspection speed and accuracy is critical for the real time inspection. Images are going to be analysed both in real time and later on the server side, but real time information is needed in order to modify if the autonomous robot inspection route is necessary. Computer vision algorithms will be affected by the illumination changes in order to segment plants from the background and to select regions with possible insects or eggs.

Using MLP machine learning pest classification model, the project is getting an accuracy of 82.34% on average on

the machine learning k-cross validation test. *Bemisia tabaci* and *Trialeurodes vaporariorum* insects are quite similar, but the machine learning model can distinguish between them. Selected manual features in Section 3.3.2 are working to discriminate between both categories, but these features are not working to distinguish between them as an egg level. *Bemisia tabaci* eggs and *Trialeurodes vaporariorum* eggs are very similar in colour and shape in the early stages. The first machine learning models contained both *Bemisia tabaci* egg and *Trialeurodes vaporariorum* egg category, but the results were poor. After different feature selection changes and tests, we decided to join both categories. Deep learning for pest detection and identification gets better results to differentiate both categories as an egg level.

Pictures that are categorized with a low confidence by the machine learning classification algorithm are removed for the selection. Computer vision algorithm will generate different regions with possible pest that will not have any insect or egg. Due to the machine learning classification confidence factor, these regions will not be categorized as a disease. A 0.5 and 0.75 confidence factor is set on this first approach.

Deep learning technique is a better solution than the combination between computer vision and machine learning. Unlike the computer vision and machine learning approach, the deep learning technique is developing the disease detection and classification in one step. It gets better accuracy, and it can distinguish between *Bemisia tabaci* egg and *Trialeurodes vaporariorum* egg category. The fastest deep learning model (SSD) and the use of a computer with a GPU can help the autonomous robot to change inspection route, while another deep learning model with higher accuracy inspects the pictures on the server side. For example, deep

learning models generated with Faster RCNN architecture can be used on server side because they have better results, but they are slower in processing time.

We also realize that all insects and/or eggs are not always discovered by the computer vision or the deep learning approach. The final goal of the inspection approach is to determine whether a plant is infected. When a plant is infected, the inspection approach needs to infer its disease. One tomato plant can be defined as not healthy if an insect or egg is found. It is not necessary to detect all the insects or eggs to achieve the project requirements, but the performance will be better. In general, more pictures from the cultivation chamber and the greenhouse need to be labelled in order to generate more robust models.

Disease detection approach using deep learning is based on the most popular image classification models. On image classification tasks, the network receives one input image and it generates one class label as an output. However, on disease detection tasks, the network gets one input image and it generates various bounding boxes with each class label. The disease detection approach using deep learning is composed of two networks. One network is used for generating region proposals using a Region Proposal Network (RPN), and the other network is used to detect diseases on the selected regions. The RPN proposes a set of different regions of several sizes that apparently contains a disease. Proposed regions are inspected by a regressor and a classifier in order to find regions with an insect or an egg. The RPN region proposal is faster than the results obtained using computer vision, and the quality of the selected regions is also better than using the machine learning approach. Computer vision and machine learning approach is based on features that are provided manually, and deep learning approach extracts these features automatically from the training dataset. Because of that, deep learning pest detection and identification approach has better accuracy, and it also works better in different scenarios and different illumination conditions. A pest detection and identification model using deep learning, labelled correctly by experts with a huge dataset of different pictures, will extract automatically many features that are impossible to infer by a human. All the combinations of possible features (color, shape, contours, size, etc.) are impossible to add to a computer vision and machine learning algorithm.

6. Future Work

Once the deep learning approach is selected to be the pest detection and identification technique, different improvements need to be added to the dataset and to the model. First, more data augmentations techniques need to be added to the model such as image resizing, translation, scaling, flipping, rotation, perspective transformations, and lighting changes. Model accuracy changes need to be tested in order to know if some of these data augmentation techniques are improving the dataset variability. Second, the generation of different synthetically generated scenes with insects and eggs can also improve the dataset variability. Third, in this first approach, pictures of *Tuta Absoluta* were not added to the model. Pictures as an insect level and as egg level need to be

added and labelled. Pictures of other diseases are also possible to be added to the model. Fourth, the egg detection and identification work is a big challenge. Part of the future work will be focused on trying to find them by the manipulation module. The deep learning algorithm needs to be improved in order to be able to detect them with a better accuracy. Finally, an Integrated Pest Management strategy will allow the autonomous robot to decide the inspection route, the inspection time in each plant, the number of images to analyse in real time, and other factors in order to detect selected pests on the greenhouse.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The GreenPatrol project has received funding from the European GNSS Agency under the European Union's (EU) Horizon 2020 research and innovation programme under grant agreement no. 776324 (project GreenPatrol, <http://www.greenpatrol-robot.eu/>).

References

- [1] PlantVillage, <https://plantvillage.org>.
- [2] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: a survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018.
- [3] P. Rajan and B. Radhakrishnan, "A survey on different image processing techniques for pest identification and plant disease detection," *International Journal of Computer Science and Network (IJCSN)*, pp. 137–141, 2016.
- [4] O. Russakovsky, J. Deng, H. Su et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [5] E. Salahat and M. Qasaimeh, "Recent advances in features extraction and description algorithms: a comprehensive survey," in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT)*, pp. 1059–1063, IEEE, 2017.
- [6] T. Rumpf, A.-K. Mahlein, U. Steiner, E.-C. Oerke, H.-W. Dehne, and L. Plümer, "Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance," *Computers and Electronics in Agriculture*, vol. 74, no. 1, pp. 91–99, 2010.
- [7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 1097–1105, 2012.
- [8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations*, pp. 1409–1556, 2015.

- [9] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proceedings of the European Conference on Computer Vision*, pp. 818–833, Springer, Cham, China, 2014.
- [10] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, IEEE, Boston, Mass, USA, 2015.
- [11] U. Mokhtar, M. A. Ali, A. E. Hassanien, and H. Hefny, "Identifying two of tomatoes leaf viruses using support vector machine," in *Information Systems Design and Intelligent Applications*, pp. 771–782, 2015.
- [12] W. Dake, "The support vector machine (SVM) based near-infrared spectrum recognition of leaves infected by the leafminers," in *Proceedings of the 1st International Conference of Innovative Computing, Information and Control (ICICIC)*, vol. 3, pp. 448–451, 2006.
- [13] S. Raza, G. Prince, J. P. Clarkson, N. M. Rajpoot, and D. Perovic, "Automatic detection of diseased tomato plants using thermal and stereo visible light images," *PLoS ONE*, vol. 10, no. 4, article e0123262, 2015.
- [14] D. L. Hernández-Rabadán, F. Ramos-Quintana, and J. Guerrero Juk, "Integrating SOMs and a bayesian classifier for segmenting diseased plants in uncontrolled environments," *The Scientific World Journal*, vol. 2014, Article ID 214674, 13 pages, 2014.
- [15] M. B. Brahim, K. Boukhalifa, and A. Moussaoui, "Deep learning for tomato diseases: classification and symptoms visualization," *Applied Artificial Intelligence*, vol. 31, no. 4, pp. 299–315, 2017.
- [16] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, 2016.
- [17] Mendel University in Brno (Faculty of AgriSciences), <http://af.mendelu.cz/en/>.
- [18] Tzutalin, LabelImg, <https://github.com/tzutalin/labelImg>.
- [19] HALCON library, <https://www.mvtec.com/products/halcon/>.
- [20] M. Abadi, P. Barham, J. Chen et al., "Tensorflow, a system for large-scale machine learning," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation*, pp. 265–283, 2016.
- [21] J. Huang, V. Rathod, C. Sun et al., "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 4, IEEE, 2017.
- [22] L. Taylor and G. Nitschke, "Improving deep learning with generic data augmentation," in *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1542–1547, IEEE, Bangalore, India, 2018.
- [23] W. A. Liu, "SSD: single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, pp. 21–37, 2016.
- [24] S. H. Ren, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.
- [25] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

