electronics

MDPI

Article

Multi-Agent Vision System for Supporting Autonomous Orchard Spraying

Piotr Góral *, Paweł Pawłowski , Karol Piniarski and Adam Dąbrowski

Division of Electronic Systems and Signal Processing, Institute of Automatic Control and Robotics, Poznan University of Technology, 60-965 Poznan, Poland; pawel.pawlowski@put.poznan.pl (P.P.); karol.piniarski@put.poznan.pl (K.P.); adam.dabrowski@put.poznan.pl (A.D.)

* Correspondence: piotr.goral@put.poznan.pl

Abstract: In this article, the authors propose a multi-agent vision system supporting the autonomous spraying of orchards and analyze the condition of trees and occurrence of pests and diseases. The vision system consists of several agents: first, for the detection of pests and diseases of fruit crops; second, for the estimation of the height of trees to be covered with spraying; third, for the classification of the developmental status of trees; and fourth, for the classification of tree infections by orchard diseases. For the classification, modified deep convolutional neural networks were used: Xception and NasNetLarge. They were trained using transfer learning and several additional techniques to avoid overfitting. Efficiency tests performed on the datasets with real orchard photos, showing accuracies ranging from 96.88% to 100%. The presented solutions will be used as part of an intelligent autonomous vehicle for orchard works, in order to minimize harm to the environment and reduce the consumption of water and plant protection products.

Keywords: convolutional neural networks; artificial intelligence; autonomous systems; visual inspection; precise horticulture

check for updates

Citation: Góral, P.; Pawłowski, P.; Piniarski, K.; Dąbrowski, A. Multi-Agent Vision System for Supporting Autonomous Orchard Spraying. *Electronics* **2024**, *13*, 494. https://doi.org/10.3390/electronics13030494

Academic Editor: Fernando De la Prieta Pintado

Received: 12 November 2023
Revised: 20 January 2024
Accepted: 22 January 2024
Published: 24 January 2024

1. Introduction

Horticulture is a fruit-producing industry in which it is necessary to use plant protection products. Even in ecological orchards, protective treatments are carried out with the use of substances of a natural or biological origin. In the course of spraying, fruit trees are covered with a water-based solution by sprayers. Unfortunately, the chemical sprays expose operators with leaky protective clothing or an insufficiently ventilated tractor cabin to harmful dust. In recent years, the control systems of orchard tractors based on video signals have been improved, which extends vehicle navigation and allows for autonomous driving [1–3]. Sprayers and agricultural machines are also equipped with vision systems that ensure precise spraying and targeted cultivation [4,5]. Thus, it can be noticed that the control technology of autonomous vehicles [6–8] is beginning to be used for practical applications on arable plantations [9–11]. The level of robotic autonomy is raised by introducing control systems using intelligent data processing [12,13].

To systematize the level of advancement of solutions, the Horticulture 4.0 classification was introduced [14]. Horticulture 4.0 classifies the levels of advancement of the digitalization technology used to support production. There are three levels:

- Level 1—use of crop-monitoring sensors.
- Level 2—processing of monitoring data supporting decision-making.
- Level 3—production automation using autonomous systems.

Fruit tree pests and diseases usually infect tree leaves [15]. The leaf is a good indicator of plant morphological variability. A single leaf has a unique pattern. These patterns can be input data that the artificial neural network algorithms can use to recognize leaf types [16]. In laboratory conditions, the recognizability of leaf types was obtained at a level from

94.69% to 97.2%. Achieving high accuracy in recognition was possible by using a seven-layer ConvNet network with data augmentation for leaf recognition [17]. In other papers, a pulse-coupled neural network and a support vector machine (SVM) [18], or a probabilistic neural network (PNN) with image and data processing techniques [19,20], were used. In tomato cultivation under cover, deep artificial neural networks were used to detect leaf diseases and pests [21,22]. Recently, some attempts have also been made to automatically recognize leaf diseases using vision systems of artificial intelligence (AI), which offer high accuracy results in laboratory conditions only [23–26]. The work in [27] presents methods for distinguishing basic diseases of apple leaves—rust, scab, and black rot—based on AI. There are also first implementations of vision algorithms based on artificial neural networks that work in a real environment [28–31]. These applications, whether prepared for laboratories or tested at orchard plantations, have some imperfections and recognize only a few types of leaf diseases. Most of the presented solutions, used in practical systems of autonomous spraying, do not allow us to recognize irregularities, which, if they occur, can destroy plantations.

Automating the visual assessment of the condition of leaves and other tree features requires the recording of proper images and their analysis. A process of automatic image analysis consists, generally, of image acquisition, pre-processing, and then appropriate analysis. Marked objects are classified in order to determine their characteristics and/or cluster. The classification of objects is a task of digital image processing and machine learning, which typically consists of two stages: feature extraction and image validation with a selected classifier. To extract the essential features of an object and to reduce the amount of data, which must be further processed, many well-known architectures such as histograms of oriented gradients (HOG) [32], local binary patterns [33], and 1D/2D Haar descriptors [34], or their combinations [35], can be used. Then, the classification decision must be made. Different categories of classifiers are used in the validation stage: support vector machine (SVM), decision trees, AdaBoost, self-organizing maps, deep convolutional neural networks (DCNNs), and their combinations. Some of the most important CNN architectures are AlexNet/CaffeNet [36,37] (as the historically first significant CNN network), VGG [38], ResNet [39], Xception [40], NasNet (in classic or mobile version) [41], EfficientNet and EfficientNetV2 [42–44].

The presented literature review has shown individual solutions concerning topics similar to those discussed in this work, but to the best of the authors' knowledge, no comprehensive fruit tree cultivation system has been proposed so far that would cover so many different aspects. This article presents the results of research conducted to develop vision algorithms for identifying diseases of fruit trees and classifying the developmental states of leaves based on CNN models. Additionally, a concept of using digital image processing algorithms to detect fruit tree pests and determine heights of trees for spraying in real time is presented. These studies constitute an important part of an integrated artificial intelligence system supporting autonomous orchard spraying.

Data for vision algorithms (vision agents) are obtained from the cameras placed on the orchard plantation and on the body of an autonomous orchard tractor with a sprayer. The vision agents supply information about the plantation to the database regardless of the spraying process or during spraying. The most advanced agents operate with artificial neural networks that recognize infected trees in an orchard plantation as a result of extracting leaf disease features and the developmental states of trees. The observation of pheromone traps and the control of leaf wetting time after atmospheric precipitation at an early stage allows for the detection of pests and disease infection, which triggers the implementation of preventive treatments. In this way, the number of treatments, used when symptoms of infection begin to appear on leaves and fruits, is reduced.

This paper is organized as follows: after the Introduction, in Section 2, the characteristics of the occurrence of selected diseases and pests of fruit trees are presented. Next, in Section 3, an intelligent solution is proposed. Section 4 presents the details of a multi-agent vision system. In Section 5, a system for autonomous protective spraying in horticulture, with an integrated AI multi-agent vision system application of convolutional neural networks for supporting orchard spraying, is presented. Section 6 summarizes the obtained results.

2. Characteristics of the Occurrence of Selected Diseases and Pests of Fruit Trees

Before preparing an AI system supporting the cultivation of fruit trees, it is necessary to study the strategies used to protect against the most common pests. Each species and even type of fruit tree requires a separate protective strategy. Obviously, there is the area of common or approximate actions that are taken during the vegetation season of fruit trees in order to prevent infections and pest appearance. In each case, it is extremely important that the right decisions are made regarding how to carry out the protective treatments. It should be noted that most pests that attack orchard plantations appear in particular time gaps, which is why their next appearance in the orchard season is called generation. Experienced growers strive to limit the number of pests as soon as they appear. A similar relationship occurs in the case of fungal diseases. In order to reduce the seeding of primary fungal spores, it is recommended to remove leaves remaining under the trees from the previous season, but the most important action is the appropriate mitigation response or removal of the appearance of the primary fungal infection, which reduces or eliminates the recurrence of the secondary fungal infection.

During the growing season, there may be a few or even a dozen infections that require preventive or intervention treatments. The most important strategies of protection, which have been included in this work, are:

- Strategy of defense against apple scab (lat. *Venturia inaequalis*): the development and sowing of spores of the fungus occurs in specific weather conditions; therefore, monitoring these conditions is of particular importance in the effective protection of apple trees against scab (Figure 1). Weather stations provide up-to-date information on changes in temperature, precipitation, air humidity and air pressure; these data are fed into the experimentally established model of the development and course of mycosis and are a source of knowledge used to make decision for a protective treatment.
- Strategy of defense against rhagoletis cherry (lat. *Rhagoletis cerasi*): protection against this pest consists of hanging traps in fruit plantations with a decoy attracting the rhagoletis cherry insect. In the Eastern-Europe geographical rank, outlets are already observed in mid-May; the observation of trap catches in an experimentally determined amount, as the economic harmfulness threshold for cherries, is a determinant for deciding to apply a protective treatment. During the ripening period of cherry fruits, there are several generations of pests that require spraying with a tractor connected to an orchard sprayer. The female pest lays eggs inside the developing fruit by cutting the skin. The white larvae hatch from the eggs and are about 4 mm long (Figure 1b); they cause the worming of fruit, making it unsuitable for consumption and industrial processing.

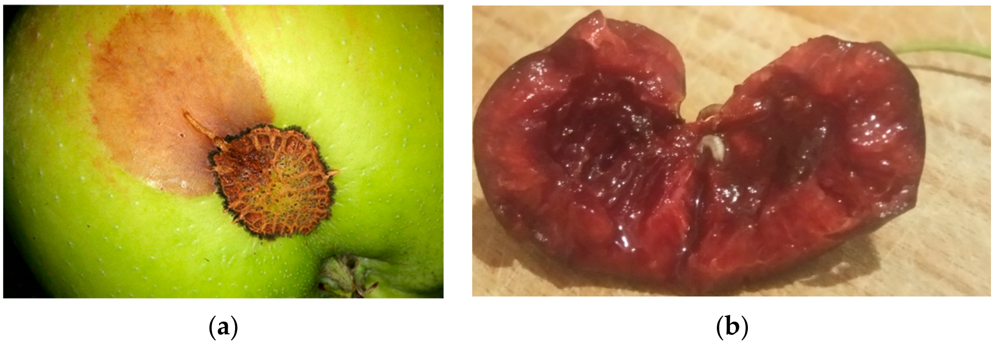


Figure 1. Illustrative infected fruits: (**a**) apple fruit infected with scab, (**b**) cherry fruit infected with the pest larva.

Effective protection against diseases and orchard pests consists mainly of prophylaxis to prevent the development of fungal pathogens or larvae that cause the degradation of green tissues of trees or fruit infestation. There is also irregular seasonality in the occurrence of certain types of diseases or pests, as well as irregularities in the severity. A strong dependence of the effectiveness of treatments on the weather conditions should also be emphasized. All these factors, together with current information from the visual inspection, should be taken into account when deciding to perform a specific protective or intervention treatment.

3. A Proposal for an Intelligent System for Autonomous Protective Spraying in Horticulture

Using the latest data processing methods based on AI [11,26], as well as the experience of one of the authors on the work in growing fruit trees and known strategies for their protection against pests [45], we propose an intelligent system for protective and autonomous spraying in horticulture based on a multi-agent vision system. The system we propose is located in the Horticulture 4.0 classification at levels 1 and 2 and also contains elements of level 3. The block diagram of this system is presented in Figure 2.

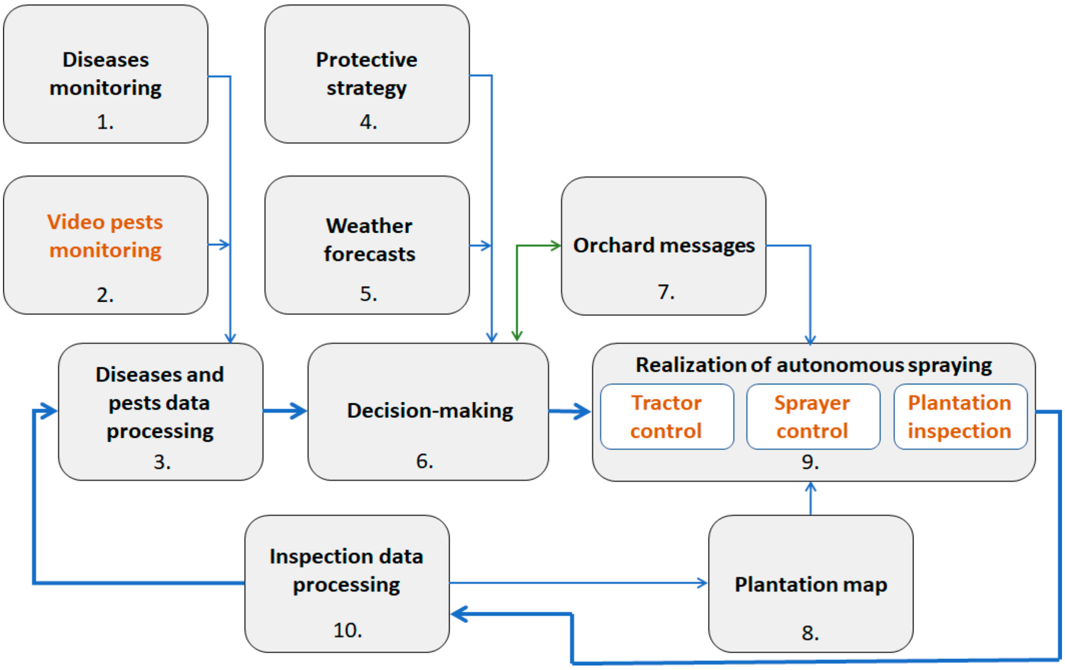
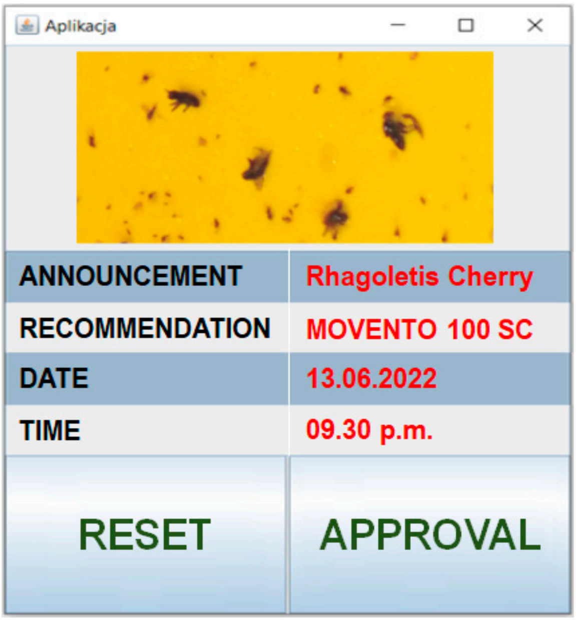


Figure 2. Scheme of the proposed intelligent system for protective spraying in horticulture. Please note that the brown color indicates vision agents for which we describe methods in this publication.

The system consists of ten blocks that perform the following tasks:

- Block 1: Disease monitoring.
- Monitoring the occurrence of diseases of fungal origin is carried out by measuring the wetting time of leaves after rainfall at a specific temperature. The leaf wetting time can be determined using video analysis. The measured values are then fed into the disease development model, on the basis of which the RIM (Relative Infection Measure) is determined [46]. Exceeding the experimentally determined RIM threshold is a signal of the risk of mycosis.
- Block 2: Video pest monitoring.
- Monitoring the occurrence of pests in orchard crops using a vision system provides information on the number of pests caught in special attracting traps [47,48]. Exceeding the experimentally determined number of pests per trap is a signal to perform spraying to remove pests.

- Block 3: Disease and pest data processing.
- Data from Blocks 1 and 2, obtained from weather stations and/or cameras located on the plantation, are processed in Block 3. If a disease or pest is detected in an orchard crop, output information is generated for the decision-making Block 6.
- Block 4: Protective strategy.
- This block contains a database with strategies for the protection of fruit plantations against particular threats. It also contains the assignment of protective measures allowed by relevant permits for fruit production. Each signal from Block 3 requires finding a suitable protection measure in Block 4.
- Block 5: Weather forecasts.
- Based on the data obtained from the forecasted weather conditions, the date of the procedure is calculated. A day is selected when the requirements for specific weather components are met, such as: temperature, humidity, wind strength, etc. This date is critical to the effectiveness of the spraying. For example, too strong a wind makes it difficult to properly cover the entire green tissue of the tree and causes a high consumption of plant protection products, as well as environmental pollution. Further, too low a humidity level causes the evaporation of part of the protection products into the atmosphere. At too low or too high a temperature, the effectiveness of the treatment is significantly lower.
- Block 6: Decision-making.
- After collecting and processing data about diseases and pests (Block 3) and additional data determining the protection measure (Block 4) along with the date of the treatment (Block 5), an automatic decision is generated. At the current stage of testing the proposed system, before further stages (i.e., before performing the tree protection treatment), an additional, manual approval by the experienced fruit grower is required to verify actions and avoid unnecessary treatments in the case of wrong decisions (Figure 3).



ANNOUNCEMENT	Rhagoletis Cherry
RECOMMENDATION	MOVENTO 100 SC
DATE	13.06.2022
TIME	09.30 p.m.

Figure 3. Example of a message generated by the decision-making Block 6.

- The decision of the grower to approve or cancel the automatic suggestion is additional information that teaches the decision-making AI system. After several seasons of testing and learning the system, it will probably be possible to skip the manual acceptance step.
- Block 7: Orchard messages.

- In this block, messages with up-to-date news generated by the commercial consulting companies from the fruit industry, for an area where the plantation is located, are obtained. These messages indicate, to some extent, the type and date of necessary orchard spraying. The messages can support the decision-making Block 6, but also, the local decision to perform the procedure made by Block 6 can be made available to the consulting companies for further processing or publication (see the two-way green arrow between Blocks 6 and 7 in Figure 2).
- Block 8: Plantation map.
- To perform autonomous protective measures, a numerical map of the orchard should be prepared. This map enables the movement of an autonomous tractor connected to an automated sprayer in a given area of an orchard plantation. The map includes the distribution of trees on the plantation, as well as the possible routes.
- Block 9: Realization of autonomous spraying.
- When the grower accepts the decision made by the integrated AI system, autonomous spraying is carried out. It is performed by an automated orchard tractor with a sprayer. The tractor is equipped with the cameras. The video streams from the cameras are transmitted to the specially designed video processing modules which perform three tasks: They support the vehicle control system, control the automatic sprayer, and collect data about the plantation, e.g., size of trees, condition of leaves, damages, possible tree diseases (especially new symptoms that were not detected before spraying), etc.
- Block 10: Inspection data processing.
- The data acquired during the spraying (Block 9) are then processed and used to make further decisions about the next necessary spraying if infection symptoms are detected. This block provides feedback in the system, making decisions more precise. Additionally, it updates the plantation map (connection to Block 8, see Figure 2).

As presented in the detailed task list above, the system is a multi-agent system, and most of the agents process images. Because the thorough testing of the system requires several annual cycles, currently, the proposed system is in development and still in the preliminary testing phase.

The following sections will primarily describe agents related to video processing and artificial intelligence. Their tasks are mainly located in Blocks 2 and 9 (cf. Figure 2).

4. Multi-Agent Vision System

The presented multi-agent vision system supports three main tasks: pest monitoring, tractor control, and sprayer control. The system uses IP color Gemini 612-23W cameras (Delta-Opti Poznań, Poznań, Poland) with a resolution of 1.4 Mpix (1280 × 720 px) at 25 fps with a 1/4 inch OmniVision OV9712 CMOS sensor, a H.264 video codec, and a 10/100 Base-T Ethernet interface.

4.1. Visual Pest-Monitoring Agent

A scheme of the visual pest-monitoring agent is presented in Figure 4. As was mentioned in previous section, this agent monitors the occurrence of pests in orchard crops and provides information on the number of pests caught in special attracting traps. The pheromone trap is continuously observed by the inspection camera (Figure 5a). The image from the camera is denoised and binarized with a threshold in the preprocessing step. This step allows us to remove noise and unwanted objects (e.g., dust or flies that are not pests) from images. Then, insect pests of a certain size are detected and counted by matching to known object pattern definitions (Figure 5b). If the number of pests in the trap exceeds the given number, the agent produces a signal to perform spraying to remove pests. Optionally, the report is supplemented by the resulting image for additional verification.

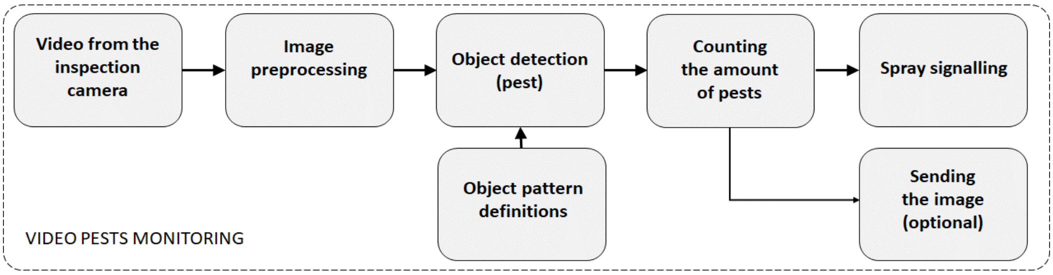

Video from the inspection camera
Image preprocessing
Object detection (pest)
Counting the amount of pests
Spray signalling
Object pattern definitions
Sending the image (optional)
VIDEO PESTS MONITORING

Figure 4. Scheme of visual pest monitoring.

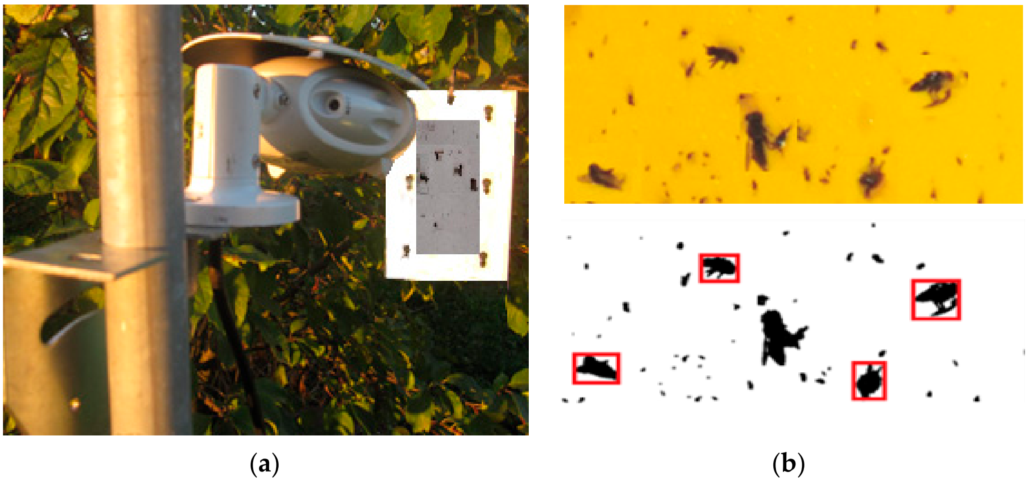


Figure 5. Detection of pests on a pheromone trap using an inspection camera: (**a**) general view of the measurement station on the plantation, (**b**) source image from the inspection camera (**top**) and after processing by the detection algorithm (**bottom**).

4.2. Autonomous Tractor Control Agent

The basic control of a moving vehicle in a closed area inaccessible to outsiders (as in the presented case—in the orchard) can be performed using an autonomous driving agent. The diagram of the control algorithm is shown in Figure 6. The tractor is equipped with a front camera located on the roof of the tractor (Figure 7a). The image from the camera provides information to the autonomous driving agent and can also be used for the remote, manual control of the vehicle via a wireless network (Figure 7b) [7]. Autonomous driving or remote, manual control are much better than classic tractor driving during spraying: the operator is not exposed to harmful conditions. However, it requires the introduction of additional security procedures.

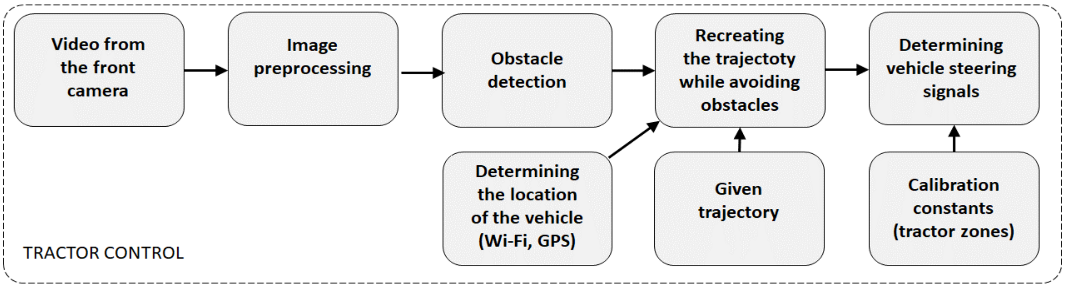
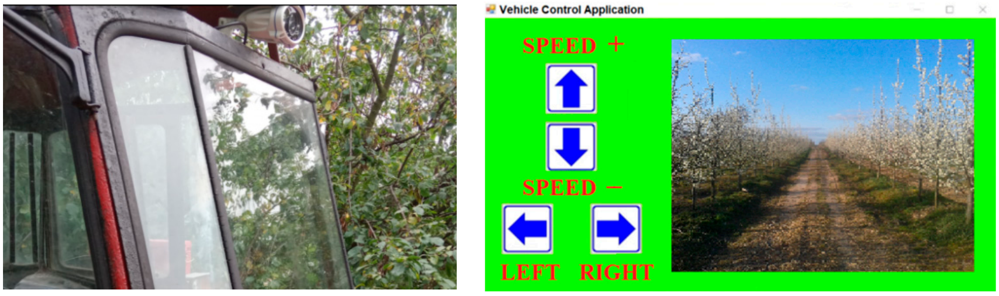

Video from the front camera
Image preprocessing
Obstacle detection
Recreating the trajectoty while avoiding obstacles
Determining vehicle steering signals
Determining the location of the vehicle (Wi-Fi, GPS)
Given trajectory
Calibration constants (tractor zones)
TRACTOR CONTROL

Figure 6. Block diagram of autonomous driving agent.



(**a**) (**b**)

Figure 7. Tractor camera views: (**a**) front camera for supporting the autonomous driving agent or remote control of the orchard tractor, (**b**) application window remotely controlling the tractor.

Remote, manual vehicle control may be necessary in emergency situations, e.g., when there is a need to guide the manual vehicle to the correct trajectory or the pre-set trajectory needs to be modified. The tractor's autonomous control system allows for the detection and recognition of objects appearing in the field of view of the front camera. It supports the function of avoiding obstacles and emergency stopping when it is impossible to avoid an obstacle. Since the orchard is an environment with little variability in time, it is possible to program the spraying trajectory that will be performed by the autonomous driving agent in advance, using the orchard map. Information about the tractor's location is obtained from the location system. The location system may use a global GPS module or a local vehicle-positioning module based on changes in the radio signal strength provided by local Wi-Fi wireless network stations [7]. In the future, vehicle location may be supported or entirely carried out by a vision system.

4.3. Automatic Sprayer Control Agent

The key function of the vision agent controlling the sprayer is the appropriate dosage of the tree protection product. This agent recognizes the developmental state of trees, detects tree height, and additionally, detects infected trees for further analysis.

A diagram of the sprayer control agent is shown in Figure 8. The agent receives the signal from a camera placed on the tractor hood, which is directed sideways, towards the row of trees being sprayed (Figure 9a,b).

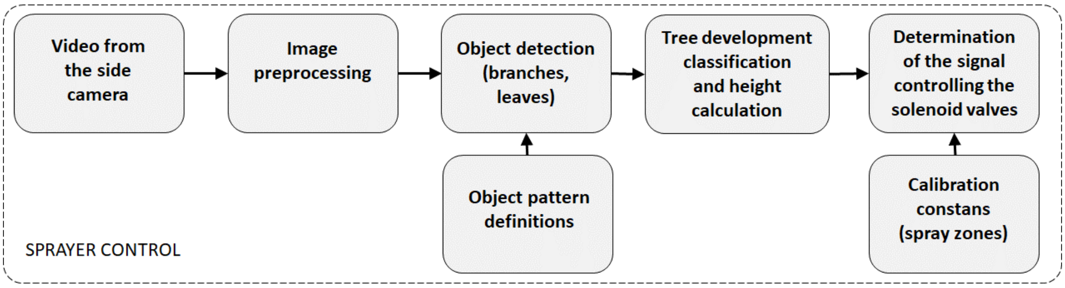

Video from the side camera
Image preprocessing
Object detection (branches, leaves)
Tree development classification and height calculation
Determination of the signal controlling the solenoid valves
Object pattern definitions
Calibration constans (spray zones)
SPRAYER CONTROL

Figure 8. Block diagram of automatic sprayer control agent.

The image from the camera is processed by a vision algorithm [49] that detects the outline of the tree by detecting leaves and branches in four zones (Figure 10). A special function has been prepared for this task. First, edge detection is performed using a Sobel mask, and then one of the morphological operations is performed—dilation. Image processing is performed sequentially to find the outline of a branch with leaves. Then, a structural element is created—a rectangle of a specific size that covers the detected contour. The trees are then assigned to one of four tree height categories. Depending on the assigned tree height category, a smaller or larger number of solenoid valves controlling the sprayer nozzles is opened (Figure 9c). This allows for spraying only the required area of the tree, and at the same time, it saves water and plant protection products.

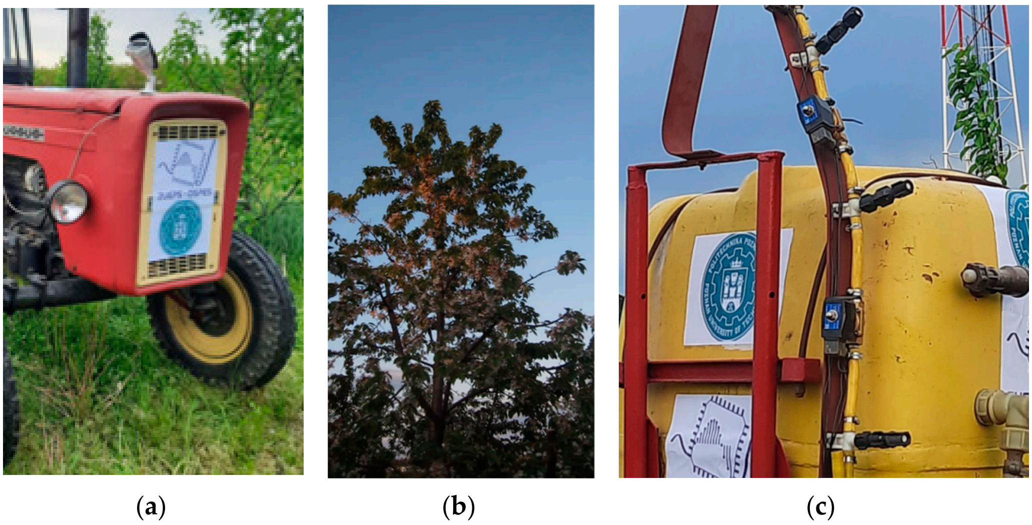


Figure 9. Tractor equipment for spraying: (**a**) side camera placed on the hood for classifying trees and controlling the sprayer, (**b**) example image collected by the side camera, (**c**) orchard sprayer with an attachment equipped with solenoid valves controlling the inlet into spray nozzles.

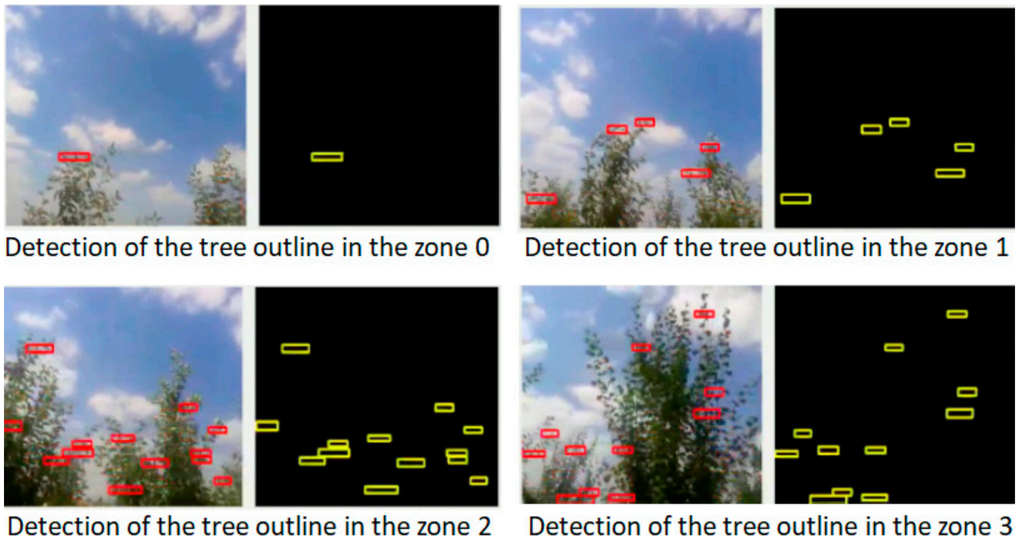

Detection of the tree outline in the zone 0
Detection of the tree outline in the zone 1
Detection of the tree outline in the zone 2
Detection of the tree outline in the zone 3

Figure 10. Height detection of trees: examples within four zones: 0 to 3.

In addition to detecting tree height, the vision agent detects the tree's developmental state and tree infections, which modify the local spraying decision. For example, trees without leaves will not be sprayed. Since automatically detecting the developmental state of a tree, as well as tree diseases, based only on image analysis is a difficult task, the authors decided to solve the problem using convolutional neural networks and carefully examine their operation.

5. Application of Convolutional Neural Networks for Supporting the Orchard Spraying

During spraying, the video agent that controls the orchard sprayer, apart from determining the height of the spraying tree, also recognizes the developmental state of trees and inspects the plantation by a classification of infected trees. The last two functions are realized in the presented system using convolutional neural networks (CNNs). In this section, we deeply analyze these tasks by performing experimental analysis.

5.1. Classification of Stages of Fruit Tree Development

Recognizing the developmental stages of fruit trees involves classifying images obtained from a fruit plantation into five separate sets related to the development of trees during the fruit season. For a proper spraying performance, we are interested in distinguishing the first five stages of the fruit season: leafless, before flowering, blooming, after flowering, and bud growth (cf. Figure 11). During the last stage, the leaves develop the most intensively on trees, and it is necessary to regulate the spray pressure depending on the given developmental state in order to obtain a complete coverage of the green tissue.

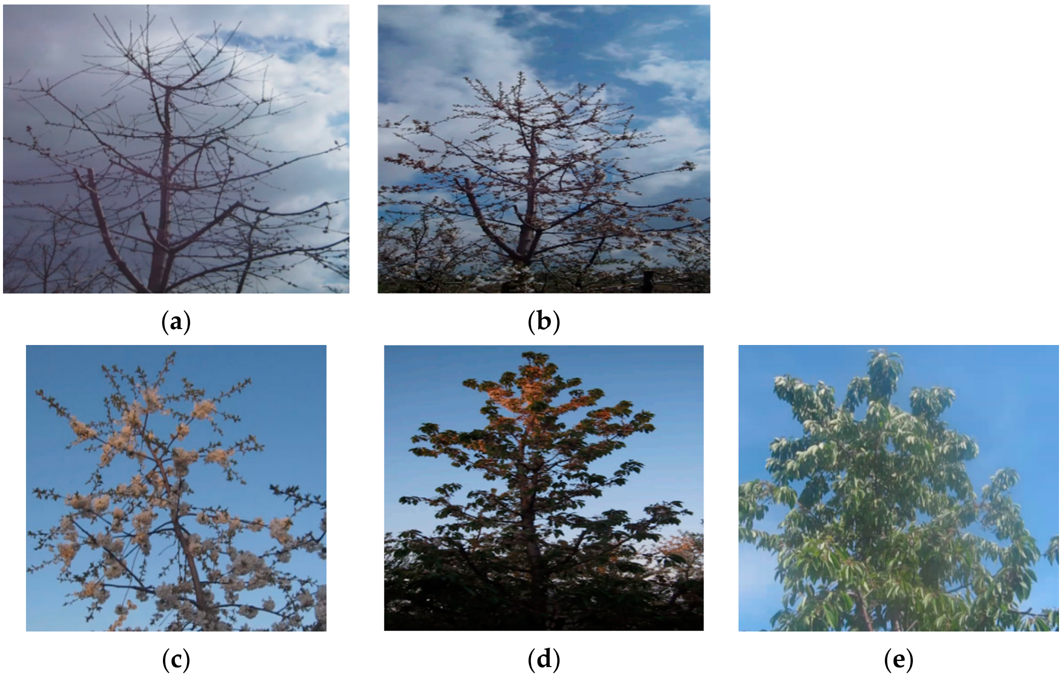


Figure 11. Sample photos from the database of various development stages of trees: (**a**) leafless period, (**b**) before flowering, (**c**) flowering period, (**d**) after flowering, (**e**) fruit growth.

For training the neural networks to work properly with the specific task and then testing their operation, the proper database of photos should be used. Since no appropriate database of photos of the considered growth stages of fruit trees was found, the authors prepared the photo database themselves. During the growing season, 7751 photos were taken in the fruit orchard at different times of the day. Table 1 and Figure 11 present some details about this database.

Table 1. Number of photos for testing the developmental stages of trees.

Tree States	Number of Samples
leafless period	1440
before flowering	1634
flowering period	1408
after flowering	1766
bud growth	1503

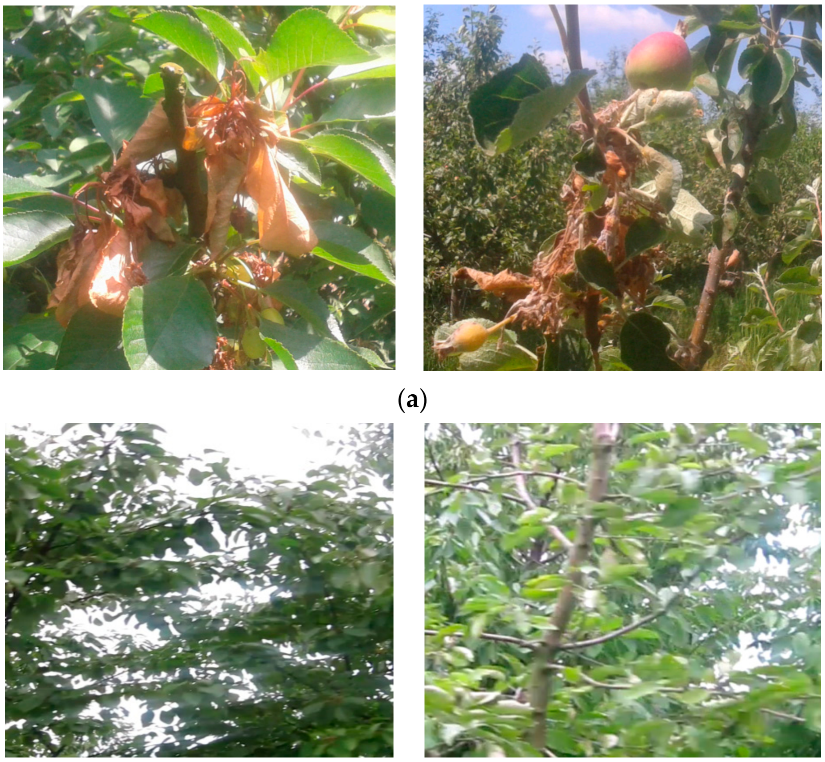
5.2. Classification of Trees Infections

The proposed vision agent can also detect tree diseases and classify disease symptoms identified during spraying. This is helpful for identifying rare and non-cyclical diseases, which, if detected early, prevent the destruction of the plantation.

Similarly, as in the previous classifier, we prepared a database of 3354 photos for training and testing the neural networks. Table 2 and Figure 12 present some details about the database of photos dedicated to the fruit tree infections.

Table 2. Number of photos for testing infected trees.

Trees State	Number of Samples
uninfected	1609
infected	1745


(**a**)

(**b**)

Figure 12. Sample photos from the database of tree infections: (**a**) infected trees, (**b**) no infection.

5.3. Used CNNs: Architectures, Supporting Techniques

5.3.1. Architectures

For performing the assumed tasks, we adopted two convolutional neural network models, namely: Xception [40] and NasNetLarge [41].

The Xception model is an extended version of the Inception V3 model, where Inception blocks have been replaced with convolutional layers used separately for individual image channels. Xception is divided into three main blocks: entry flow, middle flow and exit flow. As a result, it is a linear stack of convolutional layers, with additional connections (residual connections). Compared to Inception V3, Xception shows a small increase in classification accuracy on the ImageNet dataset and a much larger increase in accuracy on the JFT dataset.

The second CNN architecture we used was the NasNet (Neural Architecture Search Network) model. It was originally developed semi-automatically using the reinforcement learning method. Due to the time-consuming process of searching for the optimal architecture, the analysis was performed on a small dataset, and then the block configuration was transferred for verification on a larger dataset. During the experiments, the optimal architecture of a normal cell and a reduction cell was searched for. Although the search was performed on the CIFAR-10 basis, the use of the same cell architecture based on ImageNet allowed us to achieve a very high accuracy of 82.7% (top-1) and 96.2% (top-5).

5.3.2. Transfer Learning

In the case of an insufficient training set, it is possible to use an auxiliary training method—transfer learning. This method uses two stages of training. In the first stage, the model is trained on a very large database (typically, an image net for the object classification task). In the next step, the model is trained on the target learning database. For this purpose, a previously trained model with frozen weights is used. Only the last layers are trained (usually three layers of fully connected neurons), which act as a classifier in the network. In this approach, training is faster and does not require such a large number of training examples.

In the experiments carried out, the Xception network with weights trained on the ImageNet basis was used. Then, the network architecture was adapted to training with the transfer learning method by adding a layer of one-way perceptrons. A summary of the modified Xception network architecture used to classify tree development stages is presented in Table 3. For tree infection classification, only the number of perceptrons in the last layer was changed. Network training was performed only for the new network layer (10,245 parameters trained); the remaining weights were frozen (i.e., approximately 21 million parameters of the Xception architecture).

Table 3. Summary of the modified Xception network architecture: additional layers: sequential is a linear stack of layers, each with one input and one output tensor, rescaling is a preprocessing step that normalizes or standardizes input, global average pooling summarizes the values of all neurons for each patch of the input data into a feature map, dense is a regular deeply connected neural network layer, dropout is a regularization method that randomly shuts down some fraction of a layer's neurons during training to reduce overfitting.

Layer (Type)		Output
Shape	**Param**	
input_4 (InputLayer)	[(None, 331, 331, 3)]	0
input_4 (InputLayer)	[(None, 299, 299, 3)]	0
sequential_1 (Sequential)	(None, None, None, 3)	0
rescaling_1 (Rescaling)	(None, 299, 299, 3)	0
NASNet (functional)	(None, 10, 10, 2048)	20,861,480
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 2048)	0
dropout_1 (Dropout)	(None, 2048)	0
dense_1	(None, 5)	10,245
Total params: 20,871,725		
Trainable params: 10,245		
Non-trainable params: 20,861,480		

The NasNet network in the NasNetLarge version was used with weights trained on the ImageNet basis. Then, the network architecture was adapted to training with the transfer learning method by adding a layer of one-way perceptrons. A summary of the modified NasNetLarge network architecture used to classify tree development stages is presented in Table 4. In the case of the classification of tree infections, only the number of perceptrons in the last layer was changed. The network was trained only for the new network layer (20,165 parameters were trained), and the remaining weights were frozen (about 85 million parameters of the NasNet architecture).

Table 4. Summary of the modified NasNetLarge network architecture: additional layers: sequential is a linear stack of layers, each with one input and one output tensor, rescaling is a preprocessing step that normalizes or standardizes input, global average pooling summarizes the values of all neurons for each patch of the input data into a feature map, dense is a regular deeply connected neural network layer, dropout is a regularization method that randomly shuts down some fraction of a layer's neurons during training to reduce overfitting.

Layer (Type)		**Output**
Shape	**Param**	
input_4 (InputLayer)	[(None, 331, 331, 3)]	0
sequential_1 (Sequential)	(None, 331, 331, 3)	0
rescaling_1 (Rescaling)	(None, 331, 331, 3)	0
NASNet (functional)	(None, 11, 11, 4032)	84,916,818
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 4032)	0
dropout_1 (Dropout)	(None, 4032)	0
dense_1	(None, 5)	20,165
Total params: 84,936,983		
Trainable params: 20,165		
Non-trainable params: 84,916,818		

5.3.3. Preventing Network Overfitting

Model overfitting is a significant risk when training a neural network. It manifests itself mainly with the high value of accuracy obtained at the model training stage, which does not result in high accuracy on the testing set. In order to ensure the stability and reliability of the artificial network training process, a number of methods were used. The following training techniques were additionally used at different stages of learning [50,51]:

- Data augmentation techniques—adding random manipulations into training images:
- Image mirroring in vertical axis (none, mirroring).
- Image rotation (with angles from −10 to 10°).
- Image zooming (up to 40%).
- Image translation (up to 10%).
- Contrast adaptation (with contrast factor = 0.3).
- Training, validation and testing datasets—the whole database was divided into appropriate parts. The author's database of photos was divided into training, validation, and testing sets in the proportion of 60%, 20%, and 20%, respectively.
- K-fold cross-validation—K-fold training of the model: each time, a different part of the dataset becomes a testing set.
- Dropout—randomly forgetting certain neurons and not transmitting information.
- Early stopping—saving of model weights for highest accuracy on the validation set during the training process.

5.4. Experimental Results

Results of experiments on the training, validation and testing of the prepared CNN architectures are presented below. The results were evaluated based on the following expression for accuracy:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

where: TP—True Positives, TN—True Negatives, FP—False Positives, FN—False Negatives.

5.4.1. Classification of the Tree Development Stages

The results of experiments for the classification of tree development stages according to K-fold cross-validation are presented in Table 5 (for the Xception model) and Table 6 (for NasNetLarge). The accuracy plots in the training validation sets for both models are presented in Figures 13 and 14, respectively, for the Xception and NasNetLarge models.

Table 5. Classification accuracy of tree development stages for the Xception model.

K-fold	**Accuracy [%]**		
-----------	**Training**	**Validation**	**Tests**
1	97.95	99.87	99.16
2	97.98	99.14	97.44
3	96.55	99.42	97.24
4	97.43	97.89	98.57
5	97.21	98.14	98.17
Average	97.42	98.89	98.12

Table 6. Classification accuracy of tree development stages for the NasNetLarge model.

K-fold	**Accuracy [%]**		
-----------	**Training**	**Validation**	**Tests**
1	98.40	100.00	99.12
2	97.86	98.45	97.23
3	98.23	99.13	96.88
4	98.07	97.45	97.13
5	97.67	98.51	98.21
Average	98.05	98.71	97.21

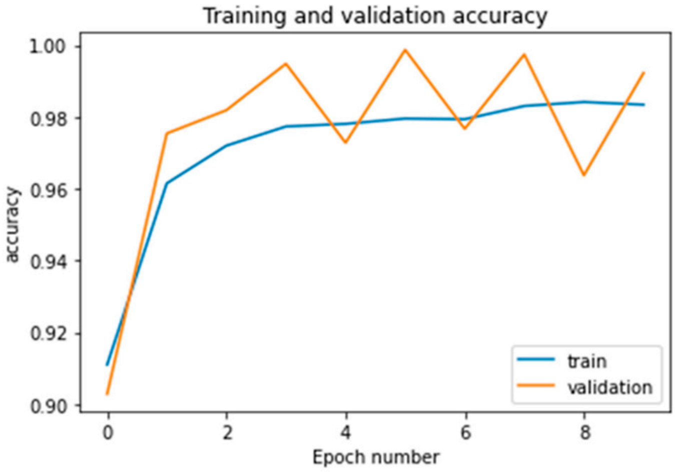


Figure 13. Classification accuracy of tree development stages for the training and validation sets with Xception model.

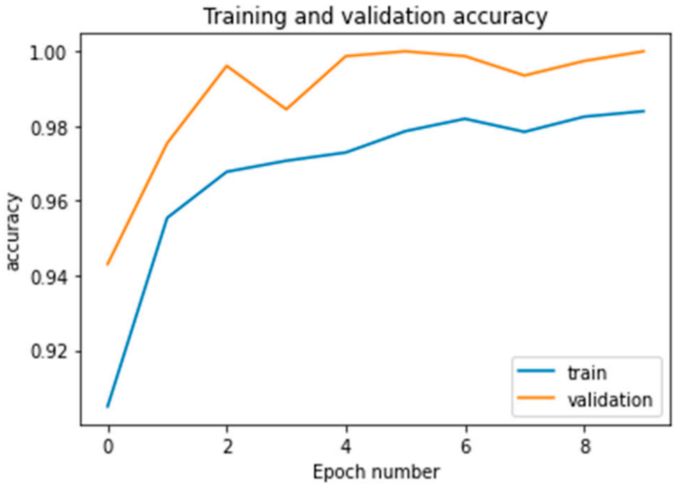


Figure 14. Classification accuracy of tree development stages for the training and validation sets with NasNetLarge model.

5.4.2. Classification of Trees Infections

The results of experiments for the classification of healthy and infected trees according to K-fold cross-validation are presented in Table 7 (for the Xception model) and Table 8 (for NasNetLarge). The accuracy plots in the training validation sets for both models are presented in Figures 15 and 16, respectively, for the Xception and NasNetLarge models.

Table 7. Classification accuracy of trees infections for the Xception model.

K-fold	Accuracy [%]		
-----------	Training	Validation	Tests
1	97.06	100.00	100.00
2	100.00	100.00	97.06
3	100.00	100.00	97.06
4	97.06	100.00	97.06
5	97.06	100.00	97.06
Average	98.24	100.00	97.65

Table 8. Classification accuracy of trees infections for the NasNetLarge model.

K-fold	Accuracy [%]		
-----------	Training	Validation	Tests
1	99.81	99.50	100.00
2	99.91	99.25	100.00
3	100.00	100.00	100.00
4	100.00	100.00	98.43
5	100.00	98.70	99.20
Average	99.94	99.49	99.78

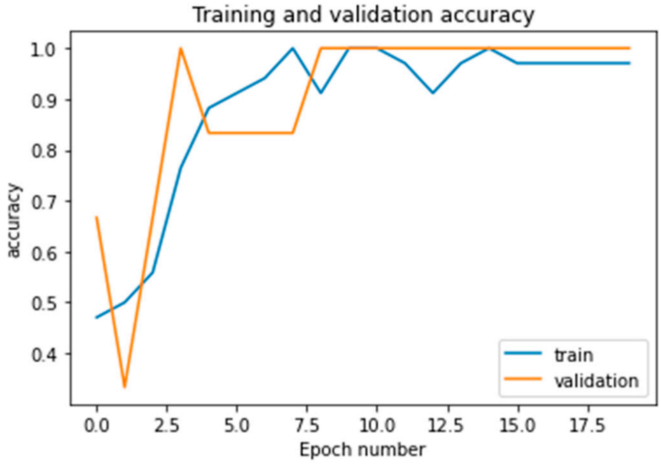


Figure 15. Classification accuracy of tree infections for the training and validation sets with Xception model.

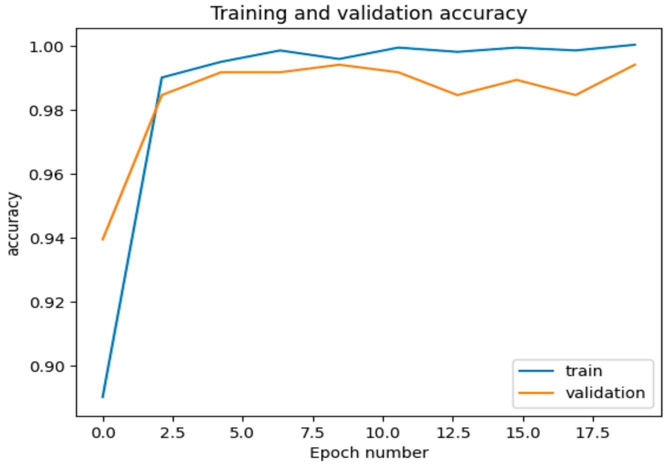


Figure 16. Classification accuracy of tree infections for the training and validation sets with NasNetLarge model.

5.5. Discussion of Results

For both tasks, i.e., for the classification of tree development stages and for the classification of tree infections, very high accuracies of classifications were achieved (at least 97% on the testing datasets). Regardless of the network model used, the obtained results were quite similar. This was also confirmed by the K-fold cross-validation approach (see Tables 5–8).

In general, larger fluctuations in accuracy (in fact in loss and accuracy) during the training process (see Figures 14 and 15) were observed for the validation set than in the results obtained on the training set. This fact is related to the smaller size of the validation set. During the testing process, no significant model over-fitting was detected. This proved that strong-enough methods of data augmentation were used.

In the case of the classification of healthy and infected trees, the CNNs achieved even higher accuracy, i.e., up to 99.78% with the NasNetLarge model on the testing dataset. This is due to the fact that the analyzed images with the plant classes of healthy and diseased trees were easier to visually differentiate from each other, and just two classes were used (healthy and diseased trees). In the future, we also intend to expand this database, but this is a difficult task due to the periodic nature of the observed diseases.

6. Conclusions

In this article, a multi-agent vision system supporting the autonomous spraying of orchards and analyzing the condition of trees and the occurrence of pests and diseases is proposed. Local data collected by agents with information about tree pests, diseases and tree development stages, combined with data from external sources, e.g., weather stations or companies presenting information for fruit growers, are input to the AI decision-making module. Complex tasks such as the classification of the developmental status of trees and the classification of tree infections by orchard diseases are carried out using two CNN types: Xception and NasNetLarge. Efficiency tests performed on datasets with real orchard photos have shown accuracies ranging from 96.88% to 100%.

The proposed architecture of the artificial intelligence system is a part of the area of precision horticulture (HA), which optimizes food production areas, minimizes costs and effectively increases production results. The models, based on which the presented research results were carried out, are part of an artificial system prepared to perform integrated autonomous orchard spraying.

The multi-agent vision system saves the time needed for classic plantation inspection by fruit growers and provides precise data on the areas of the orchard that require spraying. The precise control of the sprayer, which detects heights of trees and their infections, makes it possible to save sprayed water and plant protection products.

The presented solution can be widely implemented on new or even used tractors and orchard sprayers after their rather inexpensive modernization.

In the coming growing seasons, the authors plan to implement the presented system as a whole and examine the achieved overall effectiveness of the proposed solution.

Author Contributions: Conceptualization, P.G. and P.P.; Investigation, P.G., P.P. and A.D.; Methodology, P.G.; Software, P.G. and K.P.; Validation, P.G.; Formal analysis, P.G., P.P. and A.D.; Supervision, A.D.; Writing—original draft, P.G. and P.P.; Writing—review and editing P.G., K.P., P.P. and A.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financed partly by project 0211/SBAD/0223 and partly by the SMART4ALL EU Horizon 2020 project, Grant Agreement No. 872614.

Data Availability Statement: The data presented in this study are available in this article.

Acknowledgments: The authors would like to thank all fruit growers who consulted the developed solutions, and above all Tadeusz Góral and Hubert Fibigier for making the garden and equipment available for testing.

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

References

1. Jiang, A.; Noguchi, R.; Ahamed, T. Tree Trunk Recognition in Orchard Autonomous Operations under Different Light Conditions Using a Thermal Camera and Faster R-CNN. *Sensors* **2022**, *22*, 2065. [CrossRef] [PubMed]
2. Opiyo, S.; Okinda, C.; Zhou, J.; Mwangi, E.; Makange, N. Medial axis-based machine-vision system for orchard robot navigation. *Comput. Electron. Agric.* **2021**, *185*, 106153. [CrossRef]
3. Jiang, A.; Ahamed, T. Navigation of an Autonomous Spraying Robot for Orchard Operations Using LiDAR for Tree Trunk Detection. *Sensors* **2023**, *23*, 4808. [CrossRef] [PubMed]
4. Seol, J.; Kim, J.; Son, H.I. Field evaluations of a deep learning-based intelligent spraying robot with flow control for pear orchards. *Precis. Agric.* **2022**, *23*, 712–732. [CrossRef]
5. Abbas, I.; Liu, J.; Faheem, M.; Noor, R.S.; Shaikh, S.A.; Solangi, K.A.; Raza, S.M. Different sensor based intelligent spraying systems in Agriculture. *Sens. Actuators A Phys.* **2020**, *316*, 112265. [CrossRef]
6. Kato, S.; Takeuchi, E.; Ishiguro, Y.; Ninomiya, Y.; Takeda, K.; Hamada, T. An Open Approach to Autonomous Vehicles. *IEEE Micro.* **2015**, *35*, 60–68. [CrossRef]
7. Haboucha, C.J.; Ishaq, R.; Shiftan, Y. User preferences regarding autonomous vehicles. *Transp. Res. Part C Emerg. Technol.* **2017**, *78*, 37–49. [CrossRef]
8. Góral, P.; Pawłowski, P.; Dąbrowski, A. System bezprzewodowego zdalnego sterowania dla pojazdu autonomicznego. *Przegląd Elektrotechniczny* **2019**, *95*, 114–117. [CrossRef]

9. Baltazar, A.R.; Santos, F.N.d.; Moreira, A.P.; Valente, A.; Cunha, J.B. Smarter Robotic Sprayer System for Precision Agriculture. *Electronics* **2021**, *10*, 2061. [CrossRef]
10. Guerrero-Ibañez, A.; Reyes-Muñoz, A. Monitoring Tomato Leaf Disease through Convolutional Neural Networks. *Electronics* **2023**, *12*, 229. [CrossRef]
11. Bykov, S. World trends in the creation of robots for spraying crops. *E3S Web Conf.* **2023**, *380*, 01011. [CrossRef]
12. Kutyrev, A.; Kiktev, N.; Jewiarz, M.; Khort, D.; Smirnov, I.; Zubina, V.; Hutsol, T.; Tomasik, M.; Biliuk, M. Robotic Platform for Horticulture: Assessment Methodology and Increasing the Level of Autonomy. *Sensors* **2022**, *22*, 8901. [CrossRef] [PubMed]
13. Cantelli, L.; Bonaccorso, F.; Longo, D.; Melita, C.D.; Schillaci, G.; Muscato, G. A Small Versatile Electrical Robot for Autonomous Spraying in Agriculture. *AgriEngineering* **2019**, *1*, 391–402. [CrossRef]
14. Ludwig-Ohm, S.; Hildner, P.; Isaak, M.; Dirksmeyer, W.; Schattenberg, J. The contribution of Horticulture 4.0 innovations to more sustainable horticulture. *Procedia Comput. Sci.* **2023**, *217*, 465–477. [CrossRef]
15. Chen, J.-W.; Lin, W.-J.; Cheng, H.-J.; Hung, C.-L.; Lin, C.-Y.; Chen, S.-P. A Smartphone-Based Application for Scale Pest Detection Using Multiple-Object Detection Methods. *Electronics* **2021**, *10*, 372. [CrossRef]
16. Sekeroglu, B.; Inan, Y. Leaves Recognition System Using a Neural Network. *Procedia Comput. Sci.* **2016**, *102*, 578–582. [CrossRef]
17. Zhang, C.; Zhou, P.; Li, C.; Liu, L. A Convolutional Neural Network for Leaves Recognition Using Data Augmentation. In Proceedings of the 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, Liverpool, UK, 26–28 October 2015; pp. 2143–2150. [CrossRef]
18. Wang, Z.; Sun, X.; Zhang, Y.; Ying, Z.; Ma, Y. Leaf recognition based on PCNN. *Neural Comput. Applic.* **2016**, *27*, 899–908. [CrossRef]
19. Wu, S.G.; Bao, F.S.; Xu, E.Y.; Wang, Y.; Chang, Y.; Xiang, Q. A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network. In Proceedings of the 2007 IEEE International Symposium on Signal Processing and Information Technology, Giza, Egypt, 15–18 December 2007; pp. 11–16. [CrossRef]
20. Jeon, W.-S.; Rhee, S.-Y. Plant Leaf Recognition Using a Convolution Neural Network. *Int. J. Fuzzy Log. Intell. Syst.* **2017**, *17*, 26–34. [CrossRef]
21. Fuentes, A.; Yoon, S.; Kim, S.C.; Park, D.S. A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. *Sensors* **2017**, *17*, 2022. [CrossRef]
22. Harakannanavar, S.S.; Rudagi, J.M.; Puranikmath, V.I.; Siddiqua, A.; Pramodhini, R. Plant leaf disease detection using computer vision and machine learning algorithms. *Glob. Transit. Proc.* **2022**, *3*, 305–310. [CrossRef]
23. Khan, M.A.; Lali, M.I.; Sharif, M.; Javed, K.; Aurangzeb, K.; Haider, S.I.; Altamrah, A.S.; Akram, T. An Optimized Method for Segmentation and Classification of Apple Diseases Based on StrongCorrelation and Genetic Algorithm Based Feature Selection. *IEEE Access* **2019**, *7*, 46261–46277. [CrossRef]
24. Di, J.; Li, Q. A method of detecting apple leaf diseases based on improved convolutional neural network. *PLoS ONE* **2022**, *17*, e0262629. [CrossRef] [PubMed]
25. Bansal, P.; Kumar, R.; Kumar, S. Disease Detection in Apple Leaves Using Deep Convolutional Neural Network. *Agriculture* **2021**, *11*, 617. [CrossRef]
26. Khanna, M.; Singh, L.K.; Thawkar, S.; Goyal, M. PlaNet: A robust deep convolutional neural network model for plant leaves disease recognition. *Multimed. Tools Appl.* **2023**, *83*, 4465–4517. [CrossRef]
27. Fraiwan, M.; Faouri, E.; Khasawneh, N. On Using Deep Artificial Intelligence to Automatically Detect Apple Diseases from Leaf Images. *Sustainability* **2022**, *14*, 10322. [CrossRef]
28. Storey, G.; Meng, Q.; Li, B. Leaf Disease Segmentation and Detection in Apple Orchards for Precise Smart Spraying in Sustainable Agriculture. *Sustainability* **2022**, *14*, 1458. [CrossRef]
29. Jiang, P.; Chen, Y.; Liu, B.; He, D.; Liang, C. Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks. *IEEE Access* **2019**, *7*, 59069–59080. [CrossRef]
30. Konstantinos, P.F. Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* **2018**, *145*, 311–318.
31. Hou, J.; Yang, C.; He, Y. Detecting diseases in apple tree leaves using FPN–ISResNet–Faster RCNN, European. *J. Remote Sens.* **2023**, *56*, 2186955. [CrossRef]
32. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
33. Cao, Y.; Pranata, S.; Nishimura, H. Local Binary Pattern features for pedestrian detection at night/dark environment. In Proceedings of the 2011 18th IEEE International Conference on Image Processing, Brussels, Belgium, 11–14 September 2011; pp. 2053–2056.
34. Wei, Y.; Tian, Q.; Guo, T. An Improved Pedestrian Detection Algorithm Integrating Haar-Like Features and HOG Descriptors. *Adv. Mech. Eng.* **2013**, *5*, 546206. [CrossRef]
35. Zhang, S.; Benenson, R.; Omran, M.; Hosang, J.; Schiele, B. Towards Reaching Human Performance in Pedestrian Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 973–986. [CrossRef] [PubMed]
36. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv* **2014**, arXiv:1408.5093.

37. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
38. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556.
39. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.
40. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv* **2017**, arXiv:1610.0235. [CrossRef]
41. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning Transferable Architectures for Scalable Image Recognition. *arXiv* **2018**, arXiv:1707.07012. [CrossRef]
42. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv* **2019**, arXiv:1905.11946. [CrossRef]
43. Tan, M.; Le, Q.V. EfficientNetV2: Smaller Models and Faster Training. *arXiv* **2021**, arXiv:2104.00298. [CrossRef]
44. Piniarski, K.; Pawłowski, P.; Dąbrowski, A. Tuning of Classifiers to Speed-Up Detection of Pedestrians in Infrared Images. *Sensors* **2020**, *20*, 4363. [CrossRef]
45. Aćimović, S.; Wallis, A.; Basedow, M. Two Years of Experience with RIMpro Apple Scab Prediction Model on Commercial Apple Farms in Eastern New York. *Fruit Q.* **2018**, *26*, 21–27.
46. Daniel, C.; Grunder, J. Integrated Management of European Cherry Fruit Fly *Rhagoletis cerasi* (L.): Situation in Switzerland and Europe. *Insects* **2012**, *3*, 956–988. [CrossRef] [PubMed]
47. Katsoyannos, B.I.; Papadopoulos, N.T.; Stavridis, D. Evaluation of Trap Types and Food Attractants for *Rhagoletis cerasi* (Diptera: T ephritidae). *J. Econ. Entomol.* **2000**, *93*, 1005–1010. [CrossRef] [PubMed]
48. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; Adaptive Computation and Machine Learning series; MIT Press: Cambridge, MA, USA, 2016.
49. Emgu, C.V. Library Documentation. Available online: https://www.emgu.com/wiki/files/4.4.0/document/html/8dee1f02-8c8a-4e37-87f4-05e10c39f27d.htm (accessed on 20 December 2023).
50. Baheti, P. What Is Overfitting in Deep Learning [+10 Ways to Avoid It]. 2021. Available online: https://www.v7labs.com/blog/overfitting#h4 (accessed on 30 May 2023).
51. Li, H.; Li, J.; Guan, X.; Liang, B.; Lai, Y.; Luo, X. Research on Overfitting of Deep Learning. In Proceedings of the 2019 15th International Conference on Computational Intelligence and Security (CIS), Macao, China, 13–16 December 2019; pp. 78–81. [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.