AGRONOMY SCIENCE

VOL. LXXX (4) 2025

https://doi.org/10.24326/as.2025.5569

[1] Institute of Agricultural Engineering, Wroclaw University of Environmental and Life Sciences, Chełmońskiego 37b, 51-630 Wrocław, Poland
[2] Institute of Agroecology and Plant Production, Wroclaw University of Environmental and Life Sciences, Grunwaldzki 24A, 50-363 Wrocław, Poland
[3] Department of Biosystems Engineering, Poznań University of Life Sciences, Wojska Polskiego 50, 60-627 Poznań, Poland
* e-mail: katarzyna.pentos@upwr.edu.pl

MICHAŁ KASPROWICZ https://orcid.org/0009-0006-1224-061X

KATARZYNA PENTOŚ*[1] https://orcid.org/0000-0002-0666-1948

WIESŁAW WOJCIECHOWSKI[2] https://orcid.org/0000-0002-2011-2427

SEBASTIAN KUJAWA[3] https://orcid.org/0000-0002-5461-9170

JASPER TEMBECK MBAH[1] https://orcid.org/0000-0002-8888-6672

Advancements in plant protection – the application of machine learning to the detection of maize infestations

Zastosowanie uczenia maszynowego w wykrywaniu szkodników kukurydzy

Abstract. Plant infestations cause significant economic losses in agriculture, necessitating rapid and accurate detection for optimized agrotechnical operations and reduced environmental pollution. This study addresses this challenge by proposing a customized convolutional neural network (CNN) architecture for detecting corn leaf worm infestations in maize. The research focuses on developing unique CNN models through extensive experimentation, systematically adjusting hyperparameters like optimizers, filter numbers, and kernel sizes. The study's main contributions include the design of an accurate CNN classifier, and its implementation in a user-friendly smartphone application. The research highlights the importance of hyperparameter tuning in CNN performance, demonstrating that optimal configurations lead to high accuracy (up to 95% for accuracy, precision, recall, specificity, and F1-score). While the current model focuses on a single pest, the findings underscore the potential of custom CNN classifiers in vision systems for automated crop inspection, offering a promising solution for minimizing crop losses and the environmental impact of chemical plant protection products.

Keywords: corn leaf worm, convolutional neural network, plant protection, image recognition

Citation: Kasprowicz M., Pentoś K., Wojciechowski W., Kujawa S., Mbah J.T., 2025. Advancements in plant protection – the application of machine learning to the detection of maize infestations. Agron. Sci., 80(4), 39–55. https://doi.org/10.24326/as.2025.5569

INTRODUCTION

The aetiology of plant diseases is multifactorial, with biotic stressors (pathogens, fungi, bacteria, and insects) and abiotic stressors (weather, soil conditions, and chemicals) both playing a role [Oliveira 2019]. Plant diseases have the potential to cause significant damage to agricultural crops, resulting in reduced yields. In the event of a large-scale attack, this can even lead to the complete failure of a crop, with serious economic consequences [Li et al. 2020, Khanramaki et al. 2021]. In order to reduce the negative impact of diseases on crop quality and yield, it is necessary to detect them quickly. This is a challenging task due to the labour-intensive and time-consuming nature of traditional methods of pest and diseases detection and identification. Furthermore, farmers often lack the necessary knowledge to identify diseases or insects, which can result in the inappropriate application of agrochemicals with negative environmental consequences.

Maize is one of the three most important crops in the world, along with rice and wheat. However, since the end of the second decade of the twenty-first century, it has ranked second in terms of area sown. A number of factors have contributed to this state of affairs, the most significant of which are its versatility of use (as fodder, food, and for industrial and energy purposes), high yields, and the advancement of breeding progress (including the development of hybrid cultivars and the breeding of early-maturing cultivars). The earlier maturation of the cultivars permits their cultivation in cooler climates with a shorter growing season. Poland provides an illustrative example, where the area planted with this species increased nearly threefold (2.84 times) over a 10-year period (2010–2020) [FAOSTAT 2024]. It is regrettable that the expansion of maize cultivation is leading to an intensification of pest pressure, particularly from those species that have been identified as the most problematic. Until recently, the European corn borer (*Ostrinia nubilalis* H.) and the frit fly (*Oscinella frit* L.) were considered the most significant pests in Europe. However, more recently, the western corn rootworm (*Diabrotica virgifera* KeConte) has emerged as a growing concern, and with the anticipated effects of climate change, the corn leaf worm (*Spodoptera frugiperda*) may become a prominent issue in the near future.

The corn leaf worm is a polyphagous pest that most often attacks plants of the panicle, nightshade, cabbage family, as well as many vegetables. Of the cereals, it is most dangerous in maize crops. Its grey-brown front and white back wings are distinguished by distinct brown veins. The voracious larvae are the most damaging, and their distinctive feature is the inverted Y on their head. They cause leaf skeletonization, which significantly reduces photosynthetic capacity of crops. They also often damage flower buds, plant growth tips and even cobs and grain, resulting in reduced maize yield and quality. As reported by van der Berg et al. [2021], yield losses of up to 30% to 70% have been documented in the Americas, while losses of 11% to 100% have been observed in Africa. The corn leaf worm is mainly known for causing significant damage to crops in South America, however, there have been observations of its migration to colder regions of the world. Despite being considered a thermophilic species, the pest has also been observed in cooler regions, including Europe [Jeger et al. 2018, Babendreier et al. 2022].

The efficacy of the utilisation of agrochemicals is contingent upon the prompt identification of crop diseases. The expeditious advancement of image recognition algorithms employing machine learning in recent years has rendered the real-time identification of selected crop pests and diseases. One limitation of this method of detecting threats to crops is the necessity to represent the effects of infection in the image. Additionally, the acquisition of images at different crop growth stages and under different lighting conditions is important because these factors affect the quality of the classification. Based on image data, machine learning-based vision systems for automatic detection of pests and crop infections are being developed. Image processing with SVM classifier was utilised by Ebrahimi et al. [2017] to detect of thrips on the strawberry canopies in greenhouses. Mohan et al. [2016] developed system for detection and recognition of brown spot disease, leaf blast disease and bacterial blight disease in paddy plants. The methods used were AdaBoost classifier for detection and k-Nearest Neighbour (k-NN) and Support Vector Machine (SVM) algorithms for recognition.

In recent years, deep learning methods based on convolutional neural networks (CNNs) have become a widely used tool in agriculture for the purpose of solving classification problems. Examples of such applications include the detection of weeds [Gao et al. 2020, Hasan et al. 2021], plant water stress identification [Kamarudin et al. 2021], yield prediction [Srivastava et al. 2022], crop type classification [Kussul et al. 2017], as well as vegetable and fruit quality assessment [Gill et al. 2022]. There are also numerous applications related to the detection of pests and crop diseases [Jiao et al. 2020, Wang et al. 2020, Xu et al. 2022, Yang et al. 2022]. The availability of open databases containing images that can be used as training data represents a significant advantage in the construction of systems for plant diseases and insects recognition. Ferentinos [2018] employed an open database comprising 87,848 images of plants belonging to 25 different species. The database was utilized to train five convolutional neural network architectures, namely AlexNet, AlexNet-OWTBn, GoogLeNet, Overfeat, and VGG. The VGG model yielded the most optimal results, with a success rate of 99.53%. The RGB insect images from the three publicly available insect datasets were employed by Thenmozhi and Reddy [2019] to train a CNN architecture. The authors compared their model with pre-trained models such as AlexNet, ResNet, and VGG. The original images were converted to grayscale. The CNN model developed in this work demonstrated superior performance to the pre-trained models with transfer learning, achieving classification accuracy approximately 2% higher for the tested datasets. The PlantVillage dataset [Hughes and Salathe 2015] which contains images of common diseases across a range of crops, has been used by some researchers [Mohanty et al. 2016, Barbedo 2018, Ferentinos 2018]. Computer vision systems that employ convolutional neural networks are integral components of comprehensive solutions, including harvesting robots [Jia et al. 2020], automated sprayers [Khan et al. 2021, Storey et al. 2022], and smartphone applications, which facilitate the prediction of yields [Coviello et al. 2020] and the identification of weeds, pests, and crop diseases [Loyani and Machuve 2021, Lanjewar and Panchbhai 2023].

This study evaluates the performance of custom-designed convolutional neural network architectures on images of maize plants, with a focus on how varying hyperparameters affect model accuracy and efficiency. Unlike traditional studies that rely on pre-existing architectures, the authors developed unique CNN architectures through extensive experimentation, with numerous iterations and refinements ultimately yielding the presented model. To optimize performance, the architecture was tested under varying hyperparameters, including different optimizers, the number of filters, and kernel sizes. This exploration led to the investigation of 25 distinct models, each differing in parameters such as batch size, dropout rate, and optimizer type. By systematically adjusting these parameters, we gained insights into how specific choices impact the model's accuracy and computational efficiency, providing valuable guidance for CNN design in agricultural image classification tasks. Furthermore, a CNN classifier was implemented in a smartphone application to create a rapid detection tool for maize infestation by corn leaf worms, designed for use in field conditions.

MATERIALS AND METHODS

Custom CNN model

Convolutional neural networks represent an advanced deep learning architecture that has significantly transformed computer vision and image analysis tasks. CNNs are built to automatically and progressively learn spatial feature hierarchies directly from input data, adapting to patterns at multiple levels of abstraction. This architecture makes the neural network particularly well-suited for tasks involving image and video recognition, classification, and segmentation. The fundamental building blocks of a CNN include convolutional layers, pooling layers, and fully connected layers. Convolutional layers apply a set of learnable filters to the input, each detecting specific features at various locations. Pooling layers reduce the spatial dimensions of the feature maps, providing a form of translational invariance. Fully connected layers, typically used at the end of the network, combine these features for final classification or regression tasks.

The proposed custom CNN model's design (fig. 1) incorporates convolutional layers with multiple kernel sizes spread across them, dropout layers, max-pooling, normalization, and an early stopping feature. The selection of this structure was informed by the findings of preliminary research. In the course of this research, 10 CNNs were examined. The results of the accuracy dependence on the number of adjustable parameters are presented in table 1.

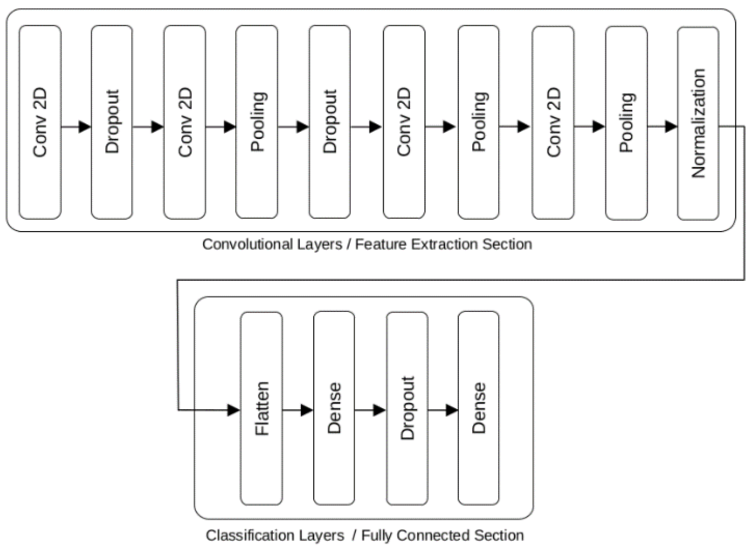

Fig. 1. Building blocks of the model

Table 1. The accuracy of the custom CNN models of various structures

No.	Number of parameters	Accuracy
1	23,907,650	0.71
2	23,907,842	0.94
3	22,245,186	0.76
4	93,506	0.72
5	2,424,802	0.62
6	11,169,218	0.79
7	23,077,282	0.68
8	10,104,418	0.66
9	31,874	0.87
10	66,599,572	0.95

Despite its strengths, the complexity of the proposed CNN model may introduce potential challenges, as larger kernel sizes (7 × 7 and 9 × 9) significantly increase the number of parameters, potentially resulting in longer training times and higher memory requirements. This can be problematic, especially when computational resources are limited and/or when training a significantly larger amount of data. Additionally, the risk of overfitting remains, particularly with a high number of parameters (tab. 2), or in case the training data is insufficient. To mitigate these issues, we employed the Early Stopping callback from the Keras library, which halts training when the validation loss ceases to improve, preventing overfitting. Early Stopping was configured with a patience of five epochs to accommodate minor fluctuations without prematurely stopping the training. This approach ensures that the model achieves optimal performance while efficiently utilizing computational resources.

Table 2. The summary of the custom CNN model

Layer (type)	Output shape	Number of parameters
Conv2D	(400, 400, 32)	896
Dropout	(400, 400, 32)	–
Conv2D	(400, 400, 64)	51,264
MaxPooling	(133, 133, 64)	–
Dropout	(133, 133, 64)	–
Conv2D	(133, 133, 128)	401,536
Pooling	(44, 44, 128)	–
Conv2D	(44, 44, 256)	2,654,464
MaxPooling	(22, 22, 256)	–
Normalization	(22, 22, 256)	1,024
Flatten	123,904	–
Dense	512	63,439,360
Dropout	512	–
Dense	2	51,026

Such CNN architecture offers significant advantages – using varying kernel sizes enables the network to capture both fine details and larger structures in the images. For example, a 3 × 3 kernel is effective at detecting small features, while larger kernels (5 × 5, 7 × 7, and 9 × 9) are better suited for recognizing bigger patterns and even entire objects. Dropout layers serve a critical role in regularization by randomly deactivating a certain fraction of neurons during training, thereby reducing the risk of overfitting and enhancing the model's generalization capabilities. Max-pooling layers further reduce the feature map dimensions and the number of parameters, mainly to lower the computational load. Normalization layers stabilize the training process by making activation distributions more predictable, leading to a faster and more stable outcome.

This CNN model has several structural and functional advantages for image-based tasks. The number of filters, starting with 32 filters and progressively increasing up to 256 allows the model to learn hierarchical representations. Initial layers capture basic features (like edges and textures) with fewer filters, while deeper layers capture more complex patterns and object parts with more filters. With four convolutional layers, the model can gradually increase abstraction levels. This depth helps the network capture both low-level and high-level spatial hierarchies in the data. By varying kernel sizes (3, 5, 7, and 9), the model learns to detect features at multiple scales within each layer. Smaller kernels focus on fine-grained details, while larger kernels cover broader spatial regions. This can help capture both small and large patterns within images, improving robustness and accuracy, especially if objects vary in scale within the dataset. Dropout layers with rates of 0.25 and 0.5 help reduce overfitting by randomly „dropping out" a fraction of neurons during each training step, ensuring the model doesn't become overly reliant on specific paths within the network. This is especially important in deeper networks. Batch normalization applied before the fully connected layers stabilizes the training process by normalizing layer outputs, accelerating convergence, and further reducing overfitting. Each MaxPooling layer progressively reduces the spatial dimensions of feature maps, which lowers computational load, enabling efficient training on large images (e.g., 400 × 400 input size) as well as helping the network focus on the most prominent features, as MaxPooling selects the strongest activations, aiding in translation invariance. After flattening, a dense layer with 512 neurons serves as a feature synthesizer, integrating information from all previous layers before the final classification. The ReLU activation in this dense layer supports effective learning and gradient flow. A final softmax layer with two outputs is suitable for binary classification, producing probabilities for each class, which is intuitive for tasks with two possible labels.

A key feature of the model training process is the use of an Early Stopping function, which prevents the network from utilizing the full number of epochs (set to 100 in this case) if it reaches a plateau in performance. This approach allows the model to halt training as soon as validation performance stabilizes, saving computational resources and preventing overfitting by avoiding unnecessary training cycles. Setting a high number of epochs (100) provides enough time for the network to learn complex patterns within the maize dataset. However, Early Stopping, along with dropout and batch normalization layers, acts as a balance to prevent overfitting, ensuring that the model does not continue learning beyond the point of optimal generalization. The dropout layer, in particular, with variations tested from 0.0 to 0.49, introduces controlled regularization by randomly deactivating certain neurons during training. This reduces dependency on specific neurons, enhancing generalization capability. Interestingly, dropout 0.5 causes the model to crash and run out of memory.

Dataset overview

The images employed in this study to train the neural network were sourced from Kaggle's „corn leaf infection dataset" [Acharya 2020]. This dataset encompasses images of corn leaves, illustrating both healthy specimens and those affected by pathogens such as the larvae of *Spodoptera frugiperda* (fall armyworm moth). The training set consists of 3770 images, with 1794 images representing healthy leaves and 1976 images representing infected leaves. During training, this data set was divided into training and validation sets in a ratio of 3 : 1. The test set contains 454 images with 204 healthy and 250 infected leaves. Illustrative images utilised for the training of the models are depicted in figure 2.

Given the high-definition nature of the original images, data preprocessing was imperative. This preprocessing included resizing the images to a uniform resolution, normalizing pixel values to standardize the dataset. These preprocessing steps were essential to ensure the dataset was adequately prepared for effective neural network training, thereby improving the models' accuracies in identifying healthy and infected corn leaves under various conditions.

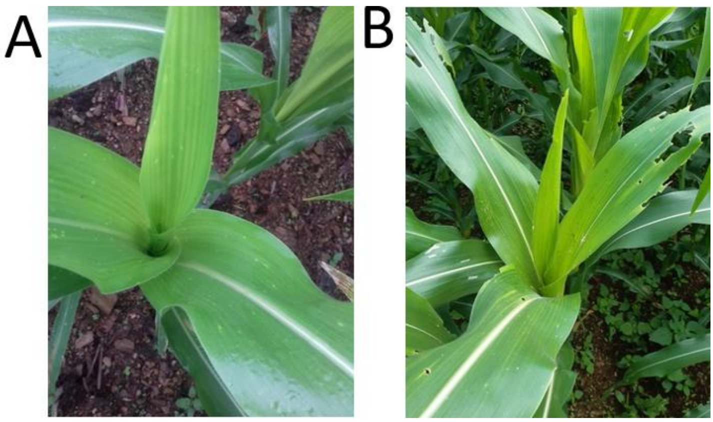

Fig. 2. Images of healthy (A) and infected (B) plant

Data pre-processing and model parameters

The original high-definition images in the dataset were resized to a resolution of 400 × 400 pixels to accommodate memory limitations. Consequently, the training set was converted into a four-dimensional tensor with dimensions 3770 × 400 × 400 × 3, representing the number of training images, the size of each image (400 × 400 pixels), and the three RGB colour channels. Similarly, the test set was converted into a four-dimensional tensor with dimensions 451 × 400 × 400 × 3. This tensor format facilitates the input of image data into the neural network for training and subsequent evaluation of the model's performance on unseen data. The dataset was pre-labeled, and a manual verification process was conducted to ensure label accuracy. The custom CNN model was employed using the TensorFlow framework. The study involved over 25 experiments, each training a new neural network on the same dataset and comparing the aggregate results. These variations were driven by adjustments in hyperparameters to identify the optimal configuration for accurate disease detection.

The hyperparameters considered during the experiments included the number of epochs (set to 100), a validation split of 25%, and batch sizes of either 32 or 64. Dropout rates varied from 0% to 49% (0%, 10%, 30%, and 49%) to assess their impact on model regularization and performance. Additionally, various optimizers, such as SGD, Adam, Ada, and Adadelta, were utilized to determine the most effective optimization strategy for this specific task. These systematic variations and evaluations were critical in refining the model to achieve high accuracy in distinguishing between healthy and diseased corn leaves, demonstrating the robustness and versatility of the CNN approach in agricultural diagnostics.

The CNN models were developed using python programming language, utilizing libraries such as NumPy, which is a fundamental tool for array manipulation and computation. It is widely recognized and utilized in the development and implementation of artificial neural networks. Complementing NumPy, the pandas library extends its functionality, serving as a versatile and comprehensive tool for data manipulation and analysis, a kind of „Swiss army knife" in the realm of data science. Creation, training and deployment of neural networks were performed based on libraries associated with neural networks – specifically Keras and TensorFlow. After training, the model was deployed for use in the production phase. The neural network was trained based on the TPU (Tensor Processing Unit) architecture, enabling the efficient execution of artificial neural networks on the Google Colab platform, which leverages cloud computing resources.

Evaluation metrics

The following metrics were used to evaluate the quality of the CNN models:

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

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

$$\mathrm{Specificity} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

$$\mathrm{F1} - \mathrm{score} = \frac{2 \cdot \mathrm{Precission} \cdot \mathrm{Recall}}{\mathrm{Precission} + \mathrm{Recall}}$$

where TP (true positive) and FP (false positive) are the number of correctly and incorrectly classified images of infected plants. TN (true negative) and FN (false negative) are the number of correctly and incorrectly classified images of non-infected plants.

RESULTS

In order to obtain an optimal model for the classification of images of maize leaves infected and non-infected by corn leaf worms, a range of values for batch size, dropout and different types of optimisers were tested. The results show notable trends and some striking contrasts in CNN performance based on hyperparameter variations, as summarised in table 3.

Table 3. The time of training and error metrics for models tested (the best model is marked in bold)

No.	Optimiser	Batch size	Dropout	Time	Accuracy	Precision	Recall	Specificity	F1-score
1	SGD	64	0	5:17:18	0.95	0.98	0.90	0.99	0.94
2	SGD	64	0.1	1:20:59	0.93	0.94	0.94	0.93	0.94
3	SGD	64	0.3	5:28:35	0.76	0.58	0.98	0.66	0.73
4	SGD	64	0.49	2:45:03	0.55	1.00	0.55	–	0.71
5	SGD	32	0	3:48:12	0.95	0.96	0.95	0.95	0.95
6	SGD	32	0.1	5:37:00	0.87	0.78	0.97	0.78	0.86
7	SGD	32	0.3	2:40:45	0.90	0.90	0.91	0.88	0.91
8	SGD	32	0.49	1:58:05	0.55	1.00	0.55	–	0.71
9	Adam	64	0	2:01:56	0.59	1.00	0.57	1.00	0.73
10	Adam	64	0.1	2:10:37	0.57	1.00	0.56	1.00	0.72
11	Adam	64	0.3	2:04:28	0.64	1.00	0.60	0.98	0.75
12	Adam	64	0.49	2:04:00	0.58	0.69	0.61	0.54	0.64
13	Adam	32	0	3:37:54	0.84	0.99	0.78	0.99	0.87
14	Adam	32	0.1	4:02:02	0.60	1.00	0.58	1.00	0.73
15	Adam	32	0.3	2:30:59	0.62	1.00	0.59	1.00	0.75
16	Adam	32	0.49	2:08:37	0.55	1.00	0.55	–	0.71
17	Adadelta	64	0	4:48:49	0.91	0.98	0.82	0.98	0.89
18	Adadelta	64	0.1	2:06:07	0.71	1.00	0.65	0.99	0.79
19	Adadelta	64	0.3	2:05:50	0.55	1.00	0.55	–	0.71
20	Adadelta	64	0.49	1:12:27	0.55	1.00	0.55	–	0.71
21	Adadelta	32	0	7:54:10	0.92	0.97	0.90	0.96	0.93

Utilising the Adadelta optimiser with a batch size of 32 and a dropout greater than 0, the training process encountered an issue pertaining to inadequate memory resources, thus resulting in its failure.

The findings reveal considerable variability in accuracy of models and computational time across different configurations, underscoring the critical role of tuning the hyperparameters in optimizing CNN performance for this specific task. The choice of optimizer significantly affects accuracy. In general, the best results were obtained with optimiser SGD and the weakest with optimiser Adam. With the SGD optimiser, the F1-score ranged from 0.71 to 0.95, while with Adam optimizer the F1-score ranged from 0.64 to 0.87. The F1-score is defined as the harmonic mean of the precision and recall, and is widely regarded as a metric that correctly indicates the reliability of a model [Vujovic 2021]. The most common metric utilised for the assessment of classification models is accuracy, which indicates the percentage of correctly classified cases. Nevertheless, it must be noted that this particular metric is not sufficiently comprehensive in order to provide a full and detailed assessment of the quality of the classifier, particularly in instances where the classes are imbalanced [Chicco and Jurman 2020]. In this work, the test set was reasonably well balanced, and when employing the SGD optimiser, accuracy was in the range of 0.55 to 0.95, with a value of 0.55 occurring in two models where all cases were classified as infected. In the context of plant disease diagnosis, the classifier can be utilised as a component of a decision-making system, in conjunction with a vision system, to determine the application of a plant protection product. The utilisation of chemical plant protection products in situations where they are not required has deleterious consequences for the environment and the economic aspect of crop management. Conversely, the non-application of plant protection products in the event of a disease outbreak can lead to yield losses. Consequently, the classifier developed in this study places significant emphasis on metrics such as recall and specificity. Recall signifies the probability of accurately predicting positive cases, whereas specificity ensures the precision of negative classifications [Baldi et al. 2000]. It was observed that models which achieved high F1-scores and accuracy values also exhibited high recall and specificity values when employing the SGD optimiser. The findings revealed no substantial impact of batch size on the quality of the models obtained. However, it was observed that smaller batch sizes (32) frequently exhibited marginal superiority over larger ones (64). In contrast, the selection of dropout probability proved to be of significant importance. The incorporation of dropout into CNNs is intended to reduce the number of parameters that require adjustment during model training, thereby counteracting the issue of overfitting and reducing the necessary training time. The experimental findings demonstrated that the introduction of dropout typically resulted in a reduction in training time. However, models that employed a low dropout probability value exhibited higher classification quality. For all optimisers employed, the highest classification quality was attained for a batch size of 32 and dropout of 0. Models with longer training times did not necessarily result in higher accuracy, indicating that resource allocation might be optimized by reducing training time without significantly impacting performance.

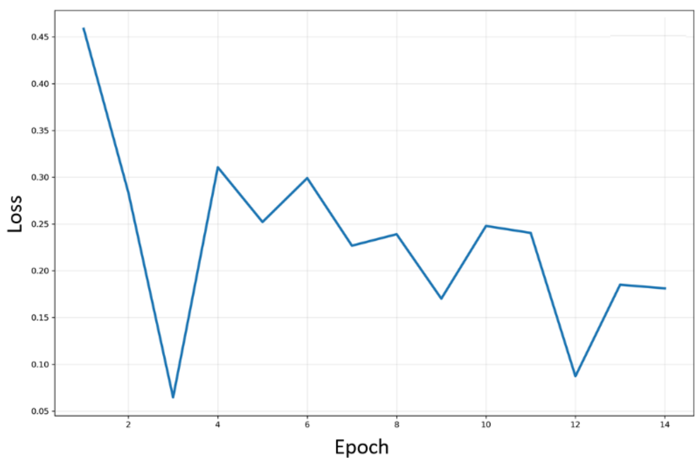

Fig. 3. Loss graph of the best model

The study identified the following models as the most effective classifiers: models 1, 2, 5, 7 and 21. The majority of the metrics analysed for these models exceeded 0.9. Model 5, for which all metrics reached a value of at least 0.95, can be considered the most effective. The loss graph of this model is presented in figure 3 and the confusion matrix showing the classification results is shown in figure 4. The results of the study suggest that a combination of SGD optimiser, low dropout and smaller batch size tends to give optimal results for this CNN on the maize dataset, providing the best possible balance between regularisation and model complexity, thus enhancing the model's ability to generalise well across the dataset.

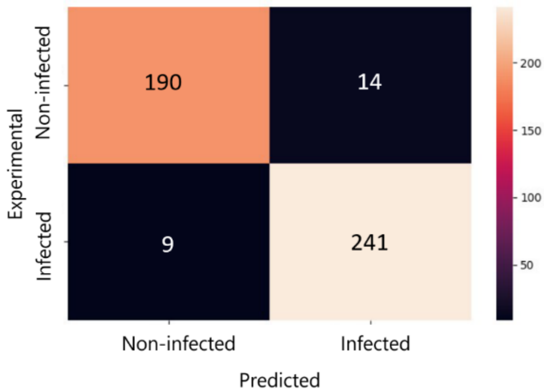

Fig. 4. Confusion matrix obtained from test dataset for the best model

In a departure from standard practice, the authors employ the Softmax function in the final layer for binary classification, even though it is typically used in multi-class scenarios [Duan et al. 2003]. Consequently, the proposed method proves more efficient than the conventional sigmoid function.

The mobile application for the detection of maize infestation by corn leaf worms

The application leverages Discord as a platform to facilitate seamless user interaction while offloading the computational tasks of image recognition and classification to a hosted cloud environment. This approach offers significant advantages. First, since Discord is a widely accessible platform, any user with a phone and a Discord account can use the application without requiring additional installations, making it universally accessible and user-friendly. Additionally, Discord provides a safe and robust environment where user management and communication security are inherently managed, reducing the need for developers to address these aspects independently. By focusing computational resources solely on the cloud-based image classification, the system achieves optimized performance, as it does not expend resources on user interface maintenance or peripheral features.

Furthermore, the hosted environment collects all images uploaded by users, enabling ongoing, seamless improvement of the neural network model through continuous training with real-world data. This process allows for smooth, quick, and painless updates to the network's capabilities, as the CNN can be re-trained with new data, improving accuracy and adaptability over time. Importantly, users benefit from these updates without needing to adjust or update anything on their devices, as all upgrades are handled directly in the cloud. This architecture thus ensures that users experience consistently enhanced functionality without the complexity or inconvenience of manual updates, making the application both highly accessible and continuously refined. Additionally, as only the initial image transfer is necessary for classification, the application remains effective even with weak internet connectivity, maximizing usability in diverse network environments.

The presented classification system integrates a convolutional neural network with a user-friendly messaging interface. The application employs a pretrained deep learning model built using TensorFlow/Keras framework to perform binary classification of plant images into „infected" or „healthy" categories. The system architecture comprises four primary components: a Discord bot interface utilizing the Discord.py library for user interaction, a CNN model for image classification, an image preprocessing module leveraging PIL (Pillow) for image manipulation, and a comprehensive logging system for tracking user interactions and analysis results. The preprocessing pipeline standardizes input images through RGB conversion, resizing to 400 × 400 pixels, and normalization before feeding them to the neural network, ensuring consistent model performance across various input format files up to 8 MB in size.

The system workflow operates through command-based interactions where users submit plant images via Discord attachments and execute classification using the "!check" command. Upon receiving an image, the application performs multi-stage validation including file format verification, size constraints checking, and image integrity assessment before processing. The CNN model processes the normalized image tensor through batch expansion and generates predictions with binary output corresponding to disease presence or absence. The implementation incorporates robust error handling mechanisms at multiple levels – model loading, image processing, Discord API communication, and file operations – while maintaining comprehensive audit trails through structured logging. This integration of machine learning capabilities with a widely-adopted communication platform provides an accessible and efficient solution for rapid plant health assessment, demonstrating the practical application of deep learning in agricultural disease detection systems. The application is deployed and accessible through the Discord platform on a dedicated server named Traynia, providing users with direct access to the classification service.

Figure 5 demonstrates the application's functionality. As demonstrated in figure 5A, the primary user interface enables the uploading of a photograph of a plant. As demonstrated in figure 5B, the image that has been uploaded for the purposes of evaluation is displayed. As demonstrated in figure 5C, the application yielded the anticipated outcome.

The presented model has the potential for implementation not only on mobile devices, but also as a component of a vision system installed on a drone, for example. The implementation of such a system has the potential to facilitate the monitoring of crops, with a particular focus on large-scale crops. The early and accurate detection of plant infestations facilitates rapid action to control the pest. This approach has the potential to assist in the mitigation of crop losses and the reduction of the adverse economic and environmental consequences associated with the utilisation of chemical plant protection products.

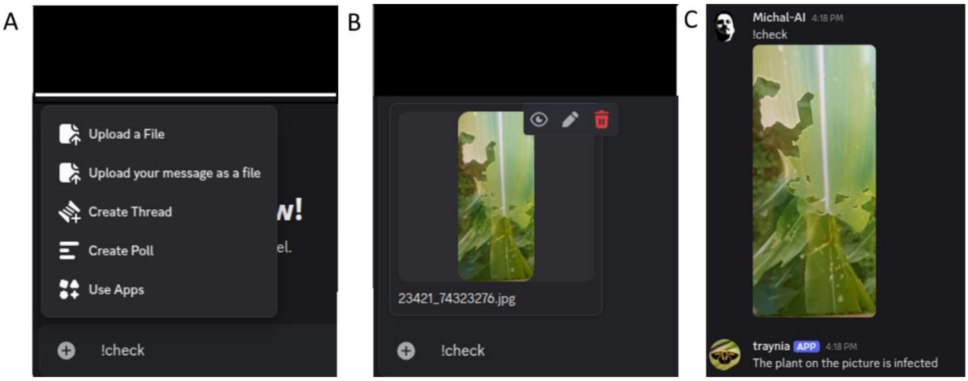

Fig. 5. The example of the functionality of the application. A – the primary user interface, B – the image that has been submitted for analysis, C – the result of the classification

DISCUSSION

Vision systems supported by artificial intelligence tools are becoming increasingly popular in agricultural applications. Convolutional networks are often used in such solutions because of their ability to extract accurate features automatically. They enable accurate classification of various objects, such as pests. Qureshi et al. [2024] proposed a system based on CNN to classify the severity levels of maize stalk rot. The system recognised six classes of stalk rot with an accuracy of 83.58%.

Models based on convolutional networks can work accurately even on devices that do not have large memory resources, such as smartphones. This makes it easy for farmers to use applications that use CNNs. Such applications include the detection and identification of pests and plant diseases. Mallick et al. [2023] presented a CNN model based on MobileNetV2 architecture for detection of six different types of mung bean diseases and four types of pests. They optimised the model and reduced its size from 20.6 MB to 6.02 MB, achieving 93.65% accuracy for the test dataset. This enabled the model to be implemented as a native application on the Android platform. Berka et al. [2023] developed a CNN-based application called CactiViT. The application used a visual image transformer to diagnose cactus cochineal with an average accuracy of 88.34%. The CactiViT mobile application allows the assessment of the health status of cacti based on images captured by the user. Lanjewar and Parab [2024] tested a customised CNN and four pre-trained deep CNNs to recognise citrus leaf diseases (black spot, melanose, canker and greening). The high-accurate (accuracy of 98%, and F1-score of 99%) optimised CNN with 15 layers and a size of 1.68 MB was deployed to the platform as a service cloud and released to the users as a link. Similar research was carried out by Peyal et al. [2023] for diseases of tomato and cotton crops. Their custom CNN model outperformed VGG-16, VGG-19, Inception-V3, MobileNet and MobileNetV2, achieving an accuracy of 97.36% and an F1-score of 97%. The model was implemented as an Android application that can detect 12 different diseases in an average time of 4.84 ms.

It is evident that both transfer learning-based CNNs (e.g., MobileNet, ResNet) and custom CNNs (e.g., the one under discussion) are effective. As demonstrated in this research, the technology is employed in the field of precision agriculture for the purpose of detecting plant diseases and pests. Nevertheless, the decision to develop and optimise a custom CNN classifier constitutes a significant methodological contribution of this study. Whilst transfer learning offers rapid deployment, the architecture presented here focuses on systematic hyperparameter tuning (including optimisers, filter numbers and kernel sizes) to achieve optimal performance for the specific task of detecting corn leaf worm infestations. Additionally, the architecture presented in this study was selected based on preliminary research, which involved training various CNN architectures. Each of the architectures examined was associated with a distinct number of adjustable parameters. Despite the network with a considerably reduced number of parameters (No. 2 in table 1) attaining only marginally lower levels of accuracy, the remaining metrics also exhibited diminished values (precision = 0.93, recall = 0.94, and F1-score = 0.94). In circumstances where the implementation of the model is constrained by limited resources, it is recommended to select that model without a substantial compromise in the quality of classification.

The rigorous approach presented in this study demonstrates that it is possible to achieve state-of-the-art detection accuracy (up to 95% across all key metrics: accuracy, precision, recall, specificity, and F1-score) using a lighter, less complex model dedicated solely to this single-pest problem. This is a critical factor for practical application. Typically, custom models are smaller in size and computationally more efficient, enabling their seamless implementation in user-friendly mobile applications for direct use in the field. Consequently, the high accuracy, when combined with the low computational overhead of the custom architecture, yields a significant practical advantage, namely the ability to execute real-time, highly localized detection. This capacity enables agricultural practitioners to transition from conventional, prophylactic spraying to precision chemical application in affected areas, thereby optimising the utilisation of plant protection products from both economic (cost reduction) and environmental (reduced chemical runoff) perspectives.

CONCLUSIONS

In the field of agriculture, infestations of plant life have been demonstrated to result in considerable economic losses. The rapid and accurate detection of diseases and pest attacks is instrumental in optimising agrotechnical operations, thereby minimising production costs and environmental pollution. The issue is especially pronounced in the context of large-scale crop cultivation, where effective monitoring and management of plant health become particularly challenging. In such cases, the combination of modern vision systems with machine learning algorithms has been proven to be a highly effective solution. Such systems can be installed, for example, on drones and thus automate the crop inspection process. The findings of our research suggest that custom CNN classifiers can be utilised in such systems. The identification of the optimal set of hyperparameters for the model is a prerequisite for the creation of an accurate classifier. In the case of the proposed model, the levels of accuracy, precision, recall, specificity, and F1-score were determined to be at the 95% level. Moreover, the findings of this study demonstrated that the optimisation of the hyperparameters of model and the training process exerted a substantial influence on the model's accuracy. Depending on these hyperparameters, we obtained models ranging from highly inaccurate to those that may have practical applications. The limitation of this work is that the model has been trained on data concerning only one type of pest. Consequently, the model is capable of accurately identifying leaf damage caused exclusively by corn leaf worm infestations. It is important to note that images showing similar damage caused by other factors, such as hail or other pests, may be classified by the model as corn leaf worm damage. The results presented herein refer to preliminary research that focused on the possibility of using custom CNN structures to recognise plant infestations. Future research will entail the development of models capable of recognising various diseases and pest attacks. In addition to employing convolutional networks for this purpose, it is also planned to utilise quantum tensors, which will facilitate more accurate image analysis, including fine details and interpretation.

REFERENCES

Acharya R., 2020. Corn Leaf Infection Dataset, https://www.kaggle.com/datasets/qramkrishna/corn-leaf-infection-dataset [access: 04.15.2023].

Babendreier D., Toepfer S., Bateman M. et al., 2022. Potential management options for the invasive moth spodoptera frugiperda in Europe. J. Econ. Entomol. 115(6), 1772–1782. https://doi.org/10.1093/JEE/TOAC089

Baldi P., Brunak S., Chauvin Y. et al., 2000. Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics 16(5), 412–424. https://doi.org/10.1093/BIOINFORMATICS/16.5.412

Barbedo J.G.A., 2018. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. Comput. Electron. Agric. 153, 46–53. https://doi.org/10.1016/J.COMPAG.2018.08.013

Berka A., Hafiane A., Es-Saady Y. et al., 2023. CactiViT: Image-based smartphone application and transformer network for diagnosis of cactus cochineal. Artif. Intell. Agric. 9, 12–21. https://doi.org/10.1016/J.AIIA.2023.07.002

Chicco D., Jurman G., 2020. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics 21(1), 1–13. https://doi.org/10.1186/s12864-019-6413-7

Coviello L., Cristoforetti M., Jurman G. et al., 2020. GBCNet: In-field grape berries counting for yield estimation by dilated CNNs. Appl. Sci. 10(14), 4870. https://doi.org/10.3390/APP10144870

Duan K., Keerthi S.S., Chu W. et al., 2003. Multi-category classification by soft-max combination of binary classifiers. In: T. Windeatt, F. Roli (eds), Multiple classifier systems. MCS 2003. Springer–Berlin–Heidelberg. https://doi.org/10.1007/3-540-44938-8_13

Ebrahimi M.A., Khoshtaghaza M.H., Minaei S. et al., 2017. Vision-based pest detection based on SVM classification method. Comput. Electron. Agric. 137, 52–58. https://doi.org/10.1016/J.COMPAG.2017.03.016

FAOSTAT, 2024. https://www.fao.org/faostat/en/#data/QV [access: 23.10.2024].

Ferentinos K.P., 2018. Deep learning models for plant disease detection and diagnosis. Comput. Electron. Agric. 145, 311–318. https://doi.org/10.1016/J.COMPAG.2018.01.009

Gao J., French A.P., Pound M.P. et al., 2020. Deep convolutional neural networks for image-based *Convolvulus sepium* detection in sugar beet fields. Plant Methods 16(1), 1–12. https://doi.org/10.1186/s13007-020-00570-z

Gill H.S., Khalaf O.I., Alotaibi Y. et al., 2022. Multi-model CNN-RNN-LSTM based fruit recognition and classification. Intell. Autom. Soft Comput. 33(1), 637–650. https://doi.org/10.32604/IASC.2022.022589

Hasan A.S.M.M., Sohel F., Diepeveen D. et al., 2021. A survey of deep learning techniques for weed detection from images. Comput. Electron. Agric. 184, 106067. https://doi.org/10.1016/J.COMPAG.2021.106067

Hughes D.P., Salathe M., 2015. An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv, 1511.08060. https://doi.org/10.48550/arXiv.1511.08060

Jeger M., Bragard C., Caffier D. et al., 2018. Pest risk assessment of *Spodoptera frugiperda* for the European Union. EFSA J. 16(8). https://doi.org/10.2903/J.EFSA.2018.5351

Jia W., Tian Y., Luo R. et al., 2020. Detection and segmentation of overlapped fruits based on optimized mask R-CNN application in apple harvesting robot. Comput. Electron. Agric. 172, 105380. https://doi.org/10.1016/J.COMPAG.2020.105380

Jiao L., Dong S., Zhang S. et al., 2020. AF-RCNN: An anchor-free convolutional neural network for multi-categories agricultural pest detection. Comput. Electron. Agric. 174, 105522. https://doi.org/10.1016/J.COMPAG.2020.105522

Kamarudin M.H., Ismail Z.H., Saidi N.B., 2021. Deep learning sensor fusion in plant water stress assessment: a comprehensive review. Appl. Sci. 11(4), 1403. https://doi.org/10.3390/APP11041403

Khan S., Tufail M., Khan M.T. et al., 2021. Deep-learning-based spraying area recognition system for unmanned-aerial-vehicle-based sprayers. Turk. J. Elec. Eng. Comp. Sci. 29(1), 241–256. https://doi.org/10.3906/elk-2004-4

Khanramaki M., Askari Asli-Ardeh E., Kozegar E., 2021. Citrus pests classification using an ensemble of deep learning models. Comput. Electron. Agric. 186, 106192. https://doi.org/10.1016/J.COMPAG.2021.106192

Kussul N., Lavreniuk M., Skakun S. et al., 2017. Deep learning classification of land cover and crop types using remote sensing data. IEEE Geosci. Remote Sens. Lett. 14(5), 778–782. https://doi.org/10.1109/LGRS.2017.2681128

Lanjewar M.G., Panchbhai K.G., 2023. Convolutional neural network based tea leaf disease prediction system on smart phone using paas cloud. Neural Comput. App. 35(3), 2755–2771. https://doi.org/10.1007/S00521-022-07743-y

Lanjewar M.G., Parab J.S., 2024. CNN and transfer learning methods with augmentation for citrus leaf diseases detection using PaaS cloud on mobile. Multimed. Tools App. 83(11), 31733–31758. https://doi.org/10.1007/S11042-023-16886-6

Li Y., Wang H., Dang L.M. et al., 2020. Crop pest recognition in natural scenes using convolutional neural networks. Comput. Electron. Agric. 169, 105174. https://doi.org/10.1016/J.COMPAG.2019.105174

Loyani L., Machuve D., 2021. A deep learning-based mobile application for segmenting tuta absoluta's damage on tomato plants. Eng. Technol. Appl. Sci. Res. 11(5), 7730–7737. https://doi.org/10.48084/ETASR.4355

Mallick M.T., Biswas S., Das A.K. et al., 2023. Deep learning based automated disease detection and pest classification in Indian mung bean. Multimed. Tools App. 82(8), 12017–12041. https://doi.org/10.1007/S11042-022-13673-7

Mohan K.J., Balasubramanian M., Palanivel S., 2016. Detection and recognition of diseases from paddy plant leaf images. Int. J. Comput. App. 144(12), 975–8887.

Mohanty S.P., Hughes D.P., Salathé M., 2016. Using deep learning for image-based plant disease detection. Front. Plant Sci. 7, 215232. https://doi.org/10.3389/FPLS.2016.01419

Bosco de Oliveira A. (ed.), 2019. Abiotic and biotic stress in plants. IntechOpen. https://doi.org/10.5772/intechopen.77845

Peyal H.I., Nahiduzzaman M., Pramanik M.A.H. et al., 2023. Plant disease classifier: detection of dual-crop diseases using lightweight 2D CNN architecture. IEEE Access 11, 110627–110643. https://doi.org/10.1109/ACCESS.2023.3320686

Qureshi S.H., Khan D.M., Razzaq A. et al., 2024. Comparison of conventional and computer-based detection of severity scales of stalk rot disease in maize. Sabrao J. Breed. Genet. 56(1), 292–301. https://doi.org/10.54910/SABRAO2024.56.1.26

Srivastava A.K., Safaei N., Khaki S. et al., 2022. Winter wheat yield prediction using convolutional neural networks from environmental and phenological data. Sci. Rep. 12(1), 1–14. https://doi.org/10.1038/s41598-022-06249-w

Storey G., Meng Q., Li B., 2022. Leaf disease segmentation and detection in apple orchards for precise smart spraying in sustainable agriculture. Sustainability 14(3), 1458. https://doi.org/10.3390/SU14031458

Thenmozhi K., Srinivasulu Reddy U., 2019. Crop pest classification based on deep convolutional neural network and transfer learning. Comput. Electron. Agric. 164, 104906. https://doi.org/10.1016/J.COMPAG.2019.104906

van den Berg J., Britz C., du Plessis H., 2021. Maize Yield response to chemical control of *Spodoptera frugiperda* at different plant growth stages in South Africa. Agriculture 11(9), 826. https://doi.org/10.3390/AGRICULTURE11090826

Vujovic Z., 2021. Classification model evaluation metrics. Int. J. Adv. Comput. Sci. Appl. 12(6). https://doi.org/10.14569/IJACSA.2021.0120670

Wang J., Li Y., Feng H. et al., 2020. Common pests image recognition based on deep convolutional neural network. Comput. Electron. Agric. 179, 105834. https://doi.org/10.1016/J.COMPAG.2020.105834

Xu W., Sun L., Zhen C. et al., 2022. Deep learning-based image recognition of agricultural pests. Appl. Sci. 12(24), 12896. https://doi.org/10.3390/APP122412896

Yang C., Teng Z., Dong C. et al., 2022. In-field citrus disease classification via convolutional neural network from smartphone images. Agriculture 12(9), 1487. https://doi.org/10.3390/AGRICULTURE12091487

Data availability statement: The software used in the study are openly available in repository: https://bazawiedzy.upwr.edu.pl/info/researchdata/UPWR0067067009ef4735aefdad4793e1b6df/.

The source of funding: Wrocław University of Environmental and Life Sciences.

Received: 9.07.2025
Accepted: 19.11.2025
Published: 31.12.2025