Apple leaf diseases detection using convolutional neural networks

Sergey Subbotin[†], Andrii Oliinyk[†], Tetiana Kolpakova[†], Vladyslav Perehuda[*†], Danylo Borovyk[†]

National University "Zaporizhzhia Polytechnic", Zhukovs'koho St, 64, Zaporizhzhia, Zaporizhia Oblast, 69063, Ukraine

Abstract

Effective and prompt detection of apple leaf diseases is key to their prevention and treatment, as well as to determine the extent of the damage caused. Today, this task is difficult due to the limited number of available methods and tools. Automated disease detection methods can improve fruit quality and reduce human error. This paper proposes the use of an improved convolutionl neural network consisting of 15 layers. To achive maximum accuracy in identifying and classifying diseases of apple leaves, it's proposed to use the CNN as a basic for the Single Shot Detector (SSD) algorithm. Benchmarking with models such as AlexNet and ResNet-50 showed that the proposed model achieves an accuracy of 96.62%, which outperforms other similar models.

Keywords

convolutional neural network, images classification, single shot detector, apple leaf diseases

1. Introduction

Diseases of agricultural crops significantly affect the yield and quality of the crop, which leads to a decrease in farm economic indicators [1]. Given the variety of diseases, detection of fruit diseases using traditional methods becomes problematic, leading to negative consequences [2, 3]. Therefore, an important problem is the timely detection of symptoms of diseases in the initial stages for their effective control [4]. To solve these problems, it is important to develop automated methods for detecting and identifying diseases at a stage when they can still be successfully treated [5, 6, 7]. Sometimes farmers do not notice diseases or have difficulty identifying them, which can lead to loss of control over the condition of the crop [8, 9].

The basic idea of automatic disease detection is to analyze the specific characteristics of images that indicate the presence of a disease and to classify these images using an appropriate classifier [10, 11]. However, the disadvantage of this approach is that the image preprocessing process is quite complex and slow. Accordingly, there is a need to use the Convolutional Neural Networks (CNN). These networks are often used for image-based

[1]*MoMLeT-2024: 6th International Workshop on Modern Machine Learning Technologies, May, 31 - June, 1, 2024, Lviv-Shatsk, Ukraine*

[*]Corresponding author.

[†]These authors contributed equally.

pereguda.vlad2702@gmail.com (V. Perehuda); daniilborovik1999@ukr.net (D. Borovyk); olejnikaa@gmail.com (A. Oliinyk); subbotin.csit@gmail.com (S. Subbotin); t.o.kolpakova@gmail.com (T. Kolpakova)

0009-0005-3121-4313 (V. Perehuda); 0009-0004-2194-0824 (D. Borovyk); 0000-0002-6740-6078 (A. Oliinyk); 0000-0001-5814-8268 (S. Subbotin); 0000-0001-8307-8134 (T. Kolpakova)

research, which efficiently identify key features of images [12, 13, 14]. CNN minimizes pre-processing and image manipulation by combining feature extraction and image recognition using convolutional and pooling layers, as well as a Softmax classifier. This type of neural network is mainly designed to classify two-dimensional images with various transformations such as scaling, displacement, and distortion. The original image can be used directly as input data for classification without the need for complex pre-processing typical of traditional image classification methods [15, 16]. In this study, a 15-layer CNN model was used as the base model for the Single Shot Detector (SDD) algorithm to detect apple fruit diseases.

2. Related Work

2.1. Dataset

The study included a detailed analysis of apple leaves to detect the presence of diseases such as black rot, cedar rust and apple scab. A public dataset consisting of 480 images of infected leaves was used for this purpose. The distribution of data for each type of disease is shown on Figure 1.

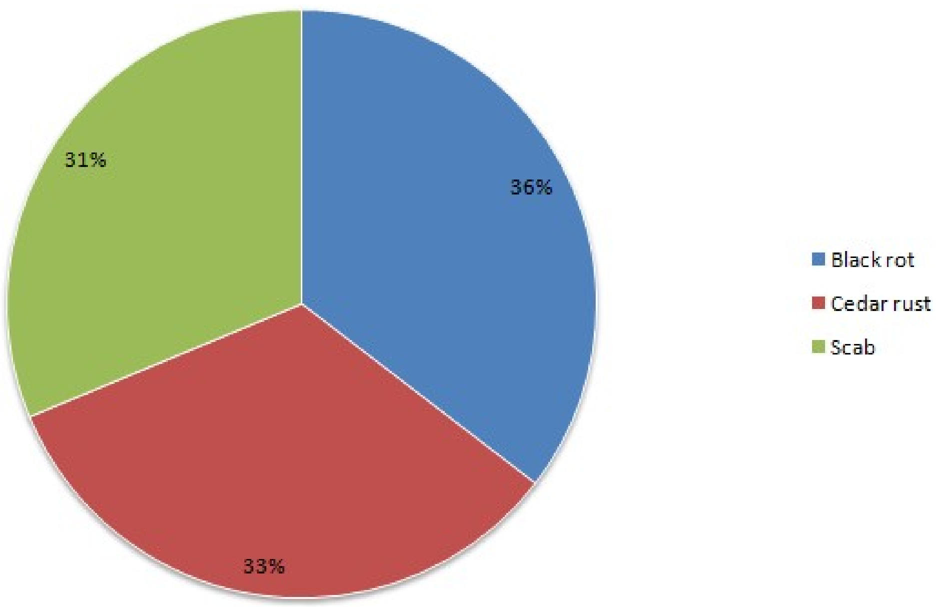

Figure 1. Dataset distribution

In order to train and test the SSD model, which was used to detect disease in images, it was necessary to carry out an annotation process. For this, the Image Labeler tool, available in the MATLAB environment, was used. This tool provided a convenient and efficient way to classify images by highlighting regions of disease using rectangles. Examples of annotated images were demonstrated on Figure 2.

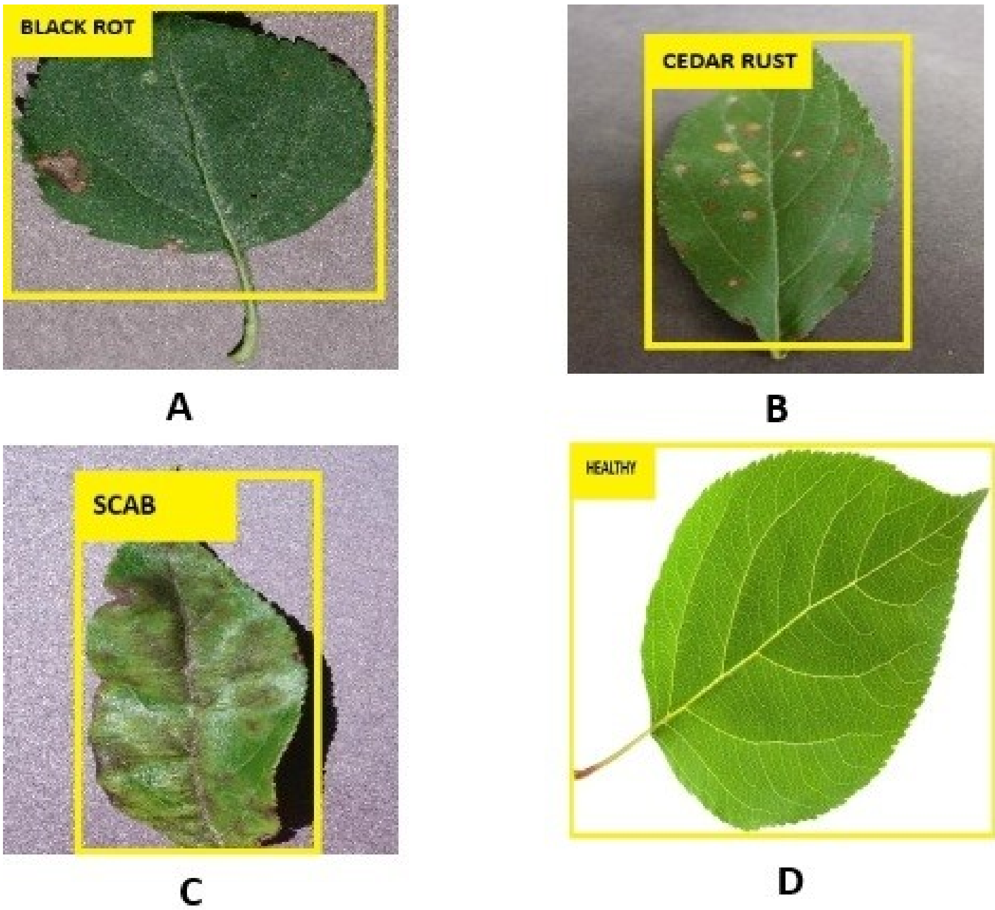

Figure 2. Annotated data examples. (A) Sample annotated Black Rot infected apple leaf. (B) Sample annotated Cedar Rust infected apple leaf. (C) Sample annotated Scab infected apple leaf. (D) Sample annotated healthy apple leaf

2.2. Model and algorithm

Among all known neural network models, convolutional neural networks are considered to be one of the most effective models for image and audio processing. They are widely used in tasks such as pattern recognition, image restoration, video analysis, music signal processing, and speech recognition [17, 18, 19]. There are three main types of layers in convolutional neural networks [20, 21]. The first one is a convolutional layer, which is used to detect various features or features in an image, such as borders, textures, and other local features. A convolutional layer can be followed by pooling layers or additional convolutional layers to further extract significant features and reduce the representation size [22, 23, 24]. The image processing process is completed by a fully connected layer, which is responsible for the classification of objects in the image. As the data passes through the different layers of the convolutional neural network, the model is able to gradually recognize increasingly complex objects in the image, starting with simple features such as color or shape, and ending with full object identification.

The Single Shot Detection (SSD) algorithm is an important tool for computer vision and image processing [25, 26]. It is used to detect objects in images for automatic pattern recognition, video analysis, and implementation of artificial intelligence systems in areas such as autopilots, video surveillance, and facial recognition [27, 28]. The main idea of SSD is that in one pass of the network it performs both object detection and their classification. This means that SSD can simultaneously determine which objects are present in the image and assign them the appropriate classes, for example, a person, a car, a bicycle, etc [29, 30] This approach significantly saves time compared to other algorithms that require separate steps for detection and classification. One of the important advantages of SSD is its high accuracy even when processing low-quality images. This makes it particularly useful for real-world applications where the image may be blurry or have low resolution [31, 32, 33]. Another important feature of SSD is its real-time use, especially in areas where speed is important, such as video surveillance systems and automotive security systems [34, 35, 36] This is achieved thanks to the use of effective algorithms and optimization of the object detection process. The basic structure of SSD algorithm is shown on Figure 3.

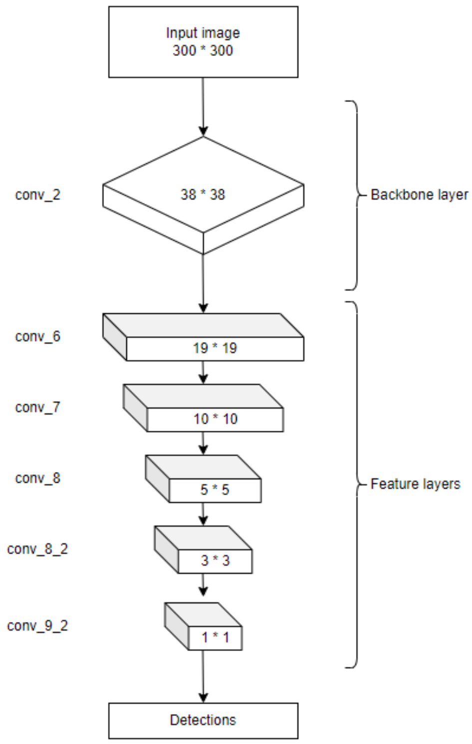

Figure 3. Structure of Single Shot Detector algorithm

3. Proposed model

To optimize the performance of the model, a modified convolutional neural network was proposed. It is crucial for the quality of image processing. This layer is the foundation upon which all input processing work is based. It also affects the process of object recognition and their further classification. In the context of the paper, this solution is a key component to ensure optimal network operation. Figure 4 shows a schematic representation of the proposed convolutional neural network structure.

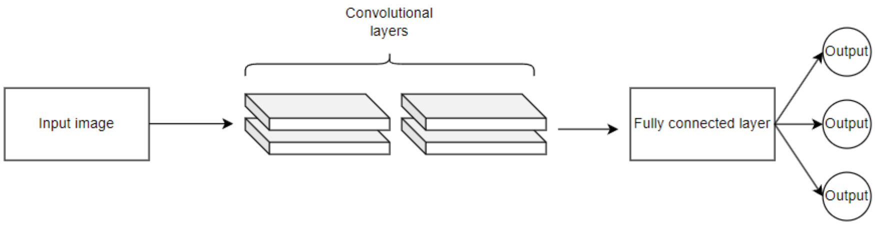

Figure 4. Proposed CNN strucutre

To achieve the maximum performance of the model, it was combined with the Single Shot Detector (SSD) algorithm. The architecture of the proposed convolutional neural network combined with the algorithm is schematically shown in Figure 5.

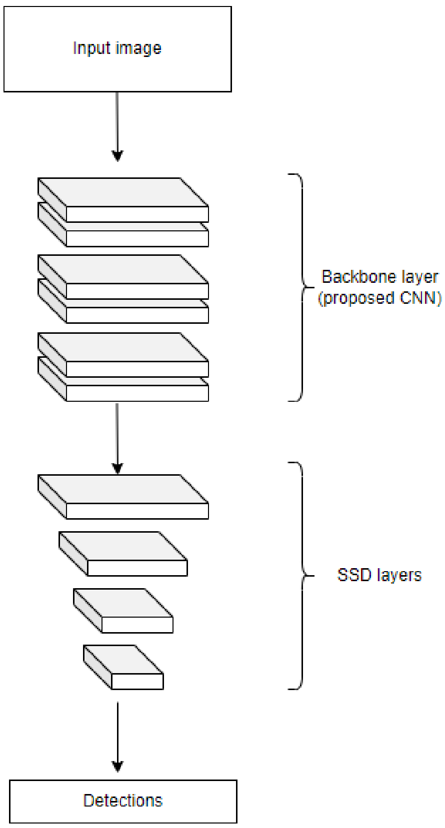

Figure 5. Architecture of proposed CNN with SSD

3.1. Input layer

This stage represents the initial process of neural network operation, when data is entered into the network for processing. Each function in the input layer transmits the received values to the neurons in the first hidden layer sequentially, from top to bottom. Each neuron then computes a weighted sum input by adding each input value multiplied by the corresponding weight. After summing these values, they are passed through the activation function. After that, the processed data is forwarded to the subsequent layer of the neural network for additional processing.

3.2. Convolutional layer

A convolutional layer is used to extract essential feature maps from the input images. The input data of the first convolutional layer is the original images, and for each subsequent convolutional layer, the input data is taken either from the pooling layer or directly from the output of the last convolutional layer. The output value for each layer in CNN can be expressed in Equation 1 as follows:

$$y_n = f(t_n), \tag{1}$$

where f is an activation function and t_n is calculated as follows:

$$f(t_n) = x_{n-1} * W_n + b^n, \tag{2}$$

where $x_{n-1}$is output of the previous layer, $W_n$is weigh and $b_n$is a bias in the convolutional layer, which is usually a sigmoid function.

3.3. Normalization layer

Normalization of the input data is performed by small batches using such a deep learning technique like batch normalization [37]. It helps to accelerate the learning of the network. This process helps to solve the problem of internal covariate shifts, where the input distribution of each layer changes during training, which leads to the degradation of network performance. Batch normalization also makes the network more robust to changes in initialization. It has become a standard component of deep neural networks and is widely used in various architectures.

3.4. ReLu layer

Rectified Linear Unit (ReLU) is a non-linear activation function widely used in neural networks. It performs the function of replacing all negative values with a decrease. It is activated only when the node's input exceeds a certain threshold. This function introduces nonlinearity into the network, allowing it to express more complex relationships between inputs and outputs. One of the advantages of ReLU is its simplicity and speed, after which it does not require large computing resources to compute. In addition, you will avoid the gradient exhaustion problem that can occur when using other activation functions. The use of ReLU makes it possible to learn deep neural networks faster, after which the activation of nodes occurs faster when the threshold value is exceeded, and is omitted in the calculation of exponential functions. Thus, it contributes to increasing the speed of learning and reducing computational costs. One of the main problems with the ReLU activation function is that it continuously resets all negative values to zero, which can limit the model's ability to learn effectively from the data. However, this problem can be easily solved by using different variations of the ReLU activation function. ReLU formula is expressed in Equation 3 as:

$$f(y)=max(0,y), \tag{3}$$

where y is input value in the layer.

3.5. MaxPooling layer

This layer is designed to reduce the spatial dimensions of the feature maps that result after using a convolution layer, while preserving the most important features. It takes the feature map from the previous convolution layer and applies a max pooling operation to it using a specific window and step, resulting in a reduced feature map. This reduced feature map is fed to the next layers of the convolutional neural network. This layer requires two parameters: window size and stride. For a feature map with dimensions H * W * C, the output size is expressed in Equation 4 as:

$$f(h)*f(w)*c, \tag{4}$$

where c is a number of channels in the feature map, h is a height of feature map, w is a width of feature map and $f(x)$ is:

$$f(x)=\frac{x-s+1}{l}, \tag{5}$$

where s is a size of filter and l is a stride length.

3.6. Softmax function

The main purpose of Softmax function is to represent confidence probabilities for neural network outputs by scaling values from 0 to 1. The function takes a vector of n real values and transforms it into another vector of n real values with a total of 1. This process allows network to transform the layer's input values, which can be positive, negative, zero, or even greater than 1, to values in the range 0 to 1. Small or negative inputs will be converted to low probabilities, while large inputs will be converted to high probabilities.

The Softmax function can be mathematically described as follows: for each element of the vector y (Equation 6), using the exponential function, we divide the exponent of this element by the sum of all the exponents of the vector y. This gives us the probability y_i given all elements of the vector (Equation 7).

$$y=(y_1,y_2,y_3,\ldots,y_n), \tag{6}$$

$$softmax(y_i)=\frac{\exp^{y_i}}{\sum_{j=1}^{n}\exp^{y_j}}, \tag{7}$$

where y_i is an element of vector y.

3.7. Output layer

The last layer of the neural network is the output layer, which generates predictions for the output. This layer applies its own set of weights and biases before forming the final prediction. Some issues may have different activation functions for hidden and output levels. For example, in classification issues, softmax activation is used to derive finite classes. Figure 7 shows a brief description of the proposed convolutional neural network (CNN) architecture.

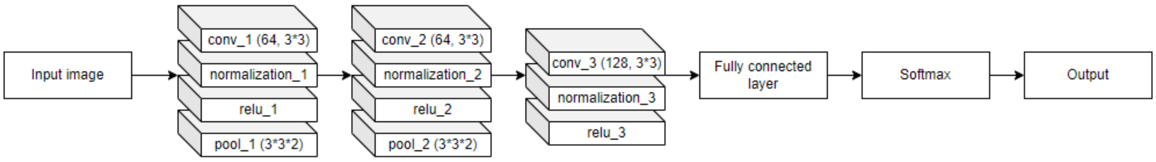

Figure 7. Proposed CNN structure

4. Experiments

This section contains the details of the experiments conducted in the study. The results of the experiments are also provided, which include the analysis of the obtained data, comparison with other methods, the quality of the model and its performance. A public dataset were used for the experiments [38]. This dataset contains 480 images of apple leaves with the following diseases – black rot, cedar rust and scab. To train CNN, images were annotated using MATLAB Image Labeler.

4.1. Precision

Precision is one of the key metrics for determining the effectiveness of a neural network model. It is measured by the ratio of the number of samples with a positive result that were correctly identified as positive to the number of samples with a positive result. Precision is an important measure of model performance, especially in cases where classes are unevenly represented in the data set. Precision is defined in Equation 8 as:

$$P = \frac{Tn}{Tn + Fn}, \tag{8}$$

where Tn is a number of positive samples that were correctly identified, Fn is a number of times negative samples were incorrectly recognized as positive.

4.2. Accuracy

The accuracy or percentage of correctly identified images in all cases is a measure that reflects the effectiveness of correctly detecting images. It measures the proportion of samples that were correctly classified relative to the total amount of data in the test sample. This metric represents the proportion of correctly classified images to the total number of images in the dataset. Accuracy is defined in Equation 9 as:

$$A = \frac{Tp + Tn}{N}, \tag{9}$$

where Tp and Tn is the number of positive and negative samples, respectively, that were correctly identiefied, N is a total amount of samples in the dataset.

4.3. Mean average precision

Mean Average Precision (mAP) is a metric used to evaluate the performance of object detection or information retrieval algorithms in pattern recognition and other fields. It measures the average value of precision for each positive object that was found in the process of testing the model, at different cut-off levels and the average value of this precision for all objects. It is used to compare different models and algorithm parameters in object detection and information retrieval. Mean average precision is expressed in Equation 10 as:

$$mAP = \frac{AP}{N}, \tag{10}$$

where N is a total amount of samples in the dataset and AF is expressed in Equation 11 as:

$$AP = \sum_{t=1}^{t=N} AP_t, \tag{11}$$

where AP_t is the average precision of object class t.

5. Results

The experiments showed that the proposed model with the SSD algorithm demonstrated the following results: precision - 0.9823, mean average precision- 0.9895 and accuracy - 0.9751. The performance evaluation of the proposed CNN model with SSD is presented in Table 1. The comparison results of the proposed model with ResNet-50 and AlexNet are presented in Table 2.

Table 1
Performance of the proposed model

Metric	Result
Precision	0.9823
Mean average precision	0.9895
Accuracy	0.9751

Table 2
Models comparison

	Precision	Mean average precision	Accuracy
Proposed CNN	0.9823	0.9895	0.9751
Alex-Net	0.9431	0.9425	0.9637
ResNet-50	0.9812	0.9853	0.9720

The experimental data set included apple leaf diseases such as black rot, cedar rust, scab as well as healthy apple leaves. Table 3 demonstrates the effectiveness of the proposed model for each class of diseases.

Table 3
Performance of the proposed model for each class of the data

	Precision	Mean average precision	Accuracy
Black rot	0.9420	0.9412	0.9425
Cedar rust	0.9605	0.9553	0.9612
Scab	0.9741	0.9703	0.9752
Healthy	0.9849	0.9813	0.9858

Figure 8 shows examples of experiments and their results, which reflect the confidence of the model in detecting apple leaf diseases. Using the proposed model, infected areas on the leaves were detected and highlighted.

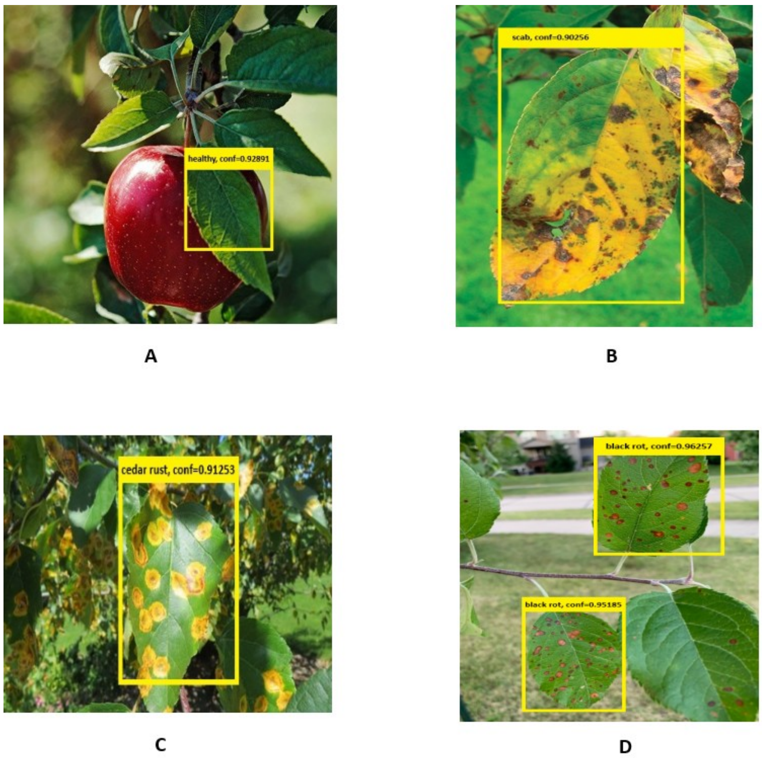

Figure 8. Sample experiment results

6. Conclusions

The problem of detection diseases of apple leaves using neural networks was considered.

The general structure of convolutional neural networks (CNN), which are used for efficient image and audio processing, as well as well-known neural networks that solve the similar problems, was analyzed.

To optimize image processing performance, a modified convolutional neural network that implements the Single Shot Detector (SSD) algorithm was proposed. The developed model was trained on a public dataset. The model showed classification accuracy level of 96.62% on the test data which is much higher than the accuracy of similar models.

7. Appendices

References

[1] A. Jafar, N. Bibi, R. A. Naqvi, A. Sadeghi-Niaraki, D. Jeong, Revolutionizing agriculture with artificial intelligence: plant disease detection methods, applications, and their limitations, Front. Plant Sci 15. (2024). doi:10.3389/fpls.2024.1356260.

[2] M. Hofman-Bergholm, A transition towards a food and agricultural system that includes both food security and planetary health, Mdpi-Foods 12 (2024). doi:10.3390/foods12010012.

[3] I. Fedorchenko, A. Oliinyk, A. Stepanenko, T. Zaiko, S. Korniienko, A. Kharchenko, Construction of a genetic method to forecast the population health indicators based on neural network models, Eastern-European Journal of Enterprise Technologies 1 (2020) 52–63. doi:10.15587/1729-4061.2020.197319.

[4] K. R. Gavhale, and G Ujwalla, An overview of the research on crop leaves disease detection using image processing techniques, IOSR Journal of Computer Engineering 16 (2014) 10-16. doi: 10.9790/0661-16151016.

[5] H. R. Khan, Z. Gillani, M. H.Jamal, A. Athar, M.T. Chaudhry, H. Chao, Y. He, M. Chen, Early identification of crop type for smallholder farming systems using deep learning on time-series sentinel-2 imagery, Mdpi-Sensors 23 (2023). doi:10.3390/s23041779.

[6] Y. Zhong, and M. Zhao, Research on deep learning in apple leaf disease recognition, Comput. Electron. Agric. 168 (2020). doi:10.1016/j.compag.2019.105146.

[7] V. Lysenko, O. Opryshko, D. Komarchuk, N. Pasichnyk, N. Zaets, A. Dudnyk. Information support of the remote nitrogen monitoring system in agricultural crops, International Journal of Computing 17 (2018) 47-54. doi:10.47839/ijc.17.1.948.

[8] P.S. Thakur, T. Sheorey, A. Ojha, A Lightweight CNN model for crop disease identification, Multimedia Tools and Applications 82 (2023) 497–520. doi:10.1007/s11042-022-13144-z.

[9] J. G. A. Barbedo, Digital image processing techniques for detecting, quantifying and classifying plant diseases, SpringerPlus 2 (2013). doi:10.1186/2193-1801-2-660.

[10] M. Ouhami, A. Hafiane, Y. Es-Saady, M. El Hajji, R. Canals, Computer vision, IoT and data fusion for crop disease detection using machine learning: a survey and ongoing research, Remote Sensing 13 (2021). doi:10.3390/rs13132486.

[11] V. V. Shkarupylo, I. V. Blinov, A. A. Chemeris, V. V. Dusheba, J. A. J. Alsayaydeh, On Applicability of Model Checking Technique in Power Systems and Electric Power

Industry., in: A. Zaporozhets, Systems, Decision and Control in Energy III, Studies in Systems, Decision and Control, 2022, volume 399. doi:10.1007/978-3-030-87675-3_1.
[12] X. Ma, Q. Man, X. Yang, P. Dong, Z. Yang, J. Wu, C. Liu, Urban feature extraction within a complex urban area with an improved 3D-CNN using airborne hyperspectral data, Remote Sensing 15 (2023). doi: 10.3390/ rs15040992.
[13] A. Oliinyk, I. Fedorchenko, A. Stepanenko, A. Katschan, Y. Fedorchenko, A. Kharchenko, D. Goncharenko, Development of genetic methods for predicting the incidence of volumes of emissions of pollutants in air, in: Proceedings of the 2nd International Workshop on Informatics and Data-Driven Medicine, IDDM, CEUR Workshop Proceedings, 2019, volume 2488, pp. 340–353.
[14] O. Melnychenko, L. Scislo, O. Savenko, A. Sachenko, P. Radiuk. Intelligent Integrated System for Fruit Detection Using Multi-UAV Imaging and Deep Learning, Sensors 24 (2024). doi:10.3390/s24061913
[15] D. Yadav, Akanksha, and A. K. Yadav, A novel convolutional neural network based model for recognition and classification of apple leaf diseases, Trait. du Signal 37 (2020) 1093–1101. doi:10.18280/ts.370622.
[16] I. Fedorchenko, I. Oliinyk, A. Stepanenko, T. Zaiko, S. Korniienko, N. Burtsev, Development of a genetic algorithm for placing power supply sources in a distributed electric network, European Journal of Enterprise Technologies 5 (2019) 6–16. doi:10.15587/1729-4061.2019.180897.
[17] V. S. Dhaka, S. V. Meena, G. Rani, D. Sinwar, Kavita, M. F. Ijaz, M. Woźniak, A survey of deep convolutional neural networks applied for prediction of plant leaf diseases, Sensors 21 (2021). doi:10.3390/s21144749.
[18] J. A. J. Alsayayde, W. A. Indra, A. W. Y. Khang, A. K. M. Zakir Hossain, V. Shkarupylo, J. Pusppanathan, The experimental studies of the automatic control methods of magnetic separators performance by magnetic product, ARPN Journal of Engineering and Applied Sciences 15 (2020) 922–927. doi:10.5281/zenodo.5163618.
[19] V. Shkarupylo, I. Blinov, A. Chemeris, V. Dusheba, J. Alsayaydeh, A. Oliinyk, Iterative Approach to TLC Model Checker Application, in: Proceedings of 2021 IEEE KhPI Week on Advanced Technology, IEEE, Kharkiv, Ukraine, 2021, pp. 283-287. doi:10.1109/KhPIWeek53812.2021.9570055.
[20] Y. Ji, S. Liu, Y. Hao, Realization of convolutional neural network based on FPGA, in: Proceedings of Third International Conference on Computer Vision and Data Mining (ICCVDM 2022), Hulun Buir, China, 2023. pp. 761–765. doi: 10.1117/12.2660023.
[21] V. V. Shkarupylo, I. V. Blinov, A. A. Chemeris, V. V. Dusheba, J. A. J. Alsayaydeh, On Applicability of Model Checking Technique in Power Systems and Electric Power Industry, Studies in Systems, Decision and Control, 2021, volume 399, pp. 3–21. doi:10.1007/978-3-030-87675-3_1.
[22] T. Y. Hsiao, Y. C. Chang, H. H. Chou, and C. T. Chiu, Filter-based deep-compression with global average pooling for convolutional networks, J. Syst. Archit. 95 (2019) 9–18. doi:10.1016/j.sysarc.2019.02.008.
[23] V. V. Shkarupylo, I. Tomičić, K. M. Kasian, The investigation of TLC model checker properties, Journal of Information and Organizational Sciences 40 (2016) 145-152. doi:10.31341/jios.40.1.7.

[24] W. A. Indra, A. W. Y. Khang, Y. T. Yung, and J. A. J. Alsayaydeh, Radio-Frequency Identification (RFID) Item Finder Using Radio Frequency Energy Harvesting, ARPN Journal of Engineering and Applied Sciences 14 (2019) 3554-3560.

[25] S. Vig, A. Arora, G. Arya, Automated license plate detection and recognition using deep learning, in: Proceedings of Advancements in Interdisciplinary Research: First International Conference, AIR 2022, Prayagraj, India, 2023, pp. 419–431. doi:10.1007/978-3- 031-23724-9_39.

[26] I. Fedorchenko, A. Oliinyk, J. A. J. Alsayaydeh, A. Kharchenko, A. Stepanenko, and V. Shkarupylo, Modified genetic algorithm to determine the location of the distribution power supply networks in the city, ARPN Journal of Engineering and Applied Sciences 15 (2020) 2850-2867.

[27] W. Li, H. Zhang, G. Wang, G. Xiong, M. Zhao, G. Li, R. Li, Deep learning based online metallic surface defect detection method for wire and arc additive manufacturing, Robotics and Computer-Integrated Manufacturing 80 (2023). doi:10.1016/j.rcim.2022.102470.

[28] V. Shkarupylo, R. Kudermetov, A. Timenko, O. Polska, On the Aspects of IoT Protocols Specification and Verification, in: Proceedings of 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 2019, pp. 93-96. doi:10.1109/PICST47496.2019.9061406.

[29] B. N. E. Nyarko, W. Bin, Z. Jinzhi, J. Odoom, Tomato fruit disease detection based on improved single shot detection algorithm, Journal of Plant Protection Research 63 (2023) 405-417. doi:10.24425/jppr.2023.146877.

[30] Y. Zheng, G. Wu, Single Shot MultiBox Detector for Urban Plantation Single Tree Detection and Location With High-Resolution Remote Sensing Imagery, Environmental Informatics and Remote Sensing (2021). doi:10.3389/fenvs.2021.755587.

[31] K. Guravaiah, Y.S. Bhavadeesh, P. Shwejan, A.H. Vardhan, S. Lavanya, Third eye: object recognition and speech generation for visually impaired, Procedia Computer Science 218 (2023) 1144–1155.doi:10.1016/j.procs. 2023.01.093.

[32] I. Fedorchenko, A. Oliinyk, A. Stepanenko, A. Svyrydenko, D. Goncharenko, Genetic method of image processing for motor vehicle recognition, in: Proceedings of 2nd International Workshop on Computer Modeling and Intelligent Systems, CMIS, Zaporizhzhia, Ukraine, 2019, volume 2353, pp. 211-226.

[33] N. F. B. A. Rahim, A. W. Y. Khang, A. Hassan, S. J. Elias, J. A. M. Gani, J. Jasmis, J. A. J. Alsayaydeh, Channel Congestion Control in VANET for Safety and Non-Safety Communication: A Review, in: Proceedings of 6th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Kedah, Malaysia, 2021, pp. 1-6. doi:10.1109/ICRAIE52900.2021.9704017.

[34] F. Wahab, I. Ullah, A. Shah, R.A. Khan, A. Choi, M.S. Anwar, Design and implementation of real-time object detection system based on single-shoot detector and OpenCV, Human-Media Interaction (2022). doi:10.3389/fpsyg.2022.1039645.

[35] A. Oliinyk, I. Fedorchenko, A. Stepanenko, M. Rud, D. Goncharenko, Implementation of evolutionary methods of solving the travelling salesman problem in a robotic warehouse, Lecture Notes on Data Engineering and Communications Technologies 48 (2021) 263–292. doi: 10.1007/978-3-030-43070-2_13.

[36] W.A. Indra, N.S. Zamzam, A. Saptari, J. A. J. Alsayaydeh, N. B. Hassim, Development of Security System Using Motion Sensor Powered by RF Energy Harvesting, in: Proceedings of IEEE Student Conference on Research and Development (SCOReD), Batu Pahat, Malaysia, pp. 254-258. doi:10.1109/SCOReD50371.2020.9250984.
[37] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, Machine Learning (2015). doi: 10.48550/arXiv.1502.03167.
[38] A. M. Hashan Apple Leaf Diseases, 2021. URL: https://www.kaggle.com /datasets/mhantor/apple-leaf-diseases/data.