ISSN 2707-1898 (print)
Український журнал інформаційних технологій
Ukrainian Journal of Information Technology
http://science.lpnu.ua/uk/ujit
https://doi.org/10.23939/ujit2025.01.077
Article received 20.03.2025 р.
Article accepted 01.05.2025 р.
UDC 004.8
Correspondence author
D. V. Fedasyuk
dmytro.v.fedasyuk@lpnu.ua

***D. V. Fedasyuk, V. I. Dorofeev***
*Lviv Polytechnic National University, Lviv, Ukraine*

# IDENTIFYING GRAPE DISEASES BY IMAGES USING ARTIFICIAL INTELLIGENCE METHODS

The paper uses modern artificial intelligence methods to investigate models and methods for determining grape disease. The existing methodologies for classification and recognition by images of grape diseases using neural networks are analyzed. Several problems for improving recognition results are highlighted. A key component of the study was the creation of extensive datasets, training neural network models with reduced computational requirements (MobileNetV3_Large, EfficientNet_B1, and ShuffleNetV2_x2), optimizing them for the mobile environment, and integrating them into a cross-platform application using the ReactNative framework. For the study, two main datasets were used: on plant diseases (PlantVillage, PlantDoc) and grape diseases (IDADP, IPM, DownyMildew, ESCA, PlantVillage, data from open search engines). The sets have been thoroughly cleaned and balanced using artificial data augmentation techniques such as rotation, scaling, contrast change, and lighting using the PlayTorch library. Training and testing of neural networks were carried out using the PyTorch library. Transmission learning was applied to improve the performance and accuracy of the models. The models showed high accuracy scores (over 93 %) with accuracy, completeness, and F1-score scores above 85 %, and the ensemble model, combining the predictions of the three architectures, demonstrated an accuracy of about 95 %.

A mobile application has been developed for iOS and Android devices, which provides identification of grape disease based on images of leaves and bunches. The application is focused on convenience and accessibility: an intuitive interface allows you to scan in real time, view and save results, and receive treatment recommendations. Using the prototype of the developed software system using the React Native framework, an analysis of the performance of models was carried out on different types of devices, which made it possible to assess their efficiency and stability in real-world application scenarios. The advantage of the mobile application is the ability to use it in vineyards without Internet access.

In conclusion, the study's main results are presented, and a decision is made about the possibility of using neural networks and an ensemble model to develop and operate a cross-platform mobile software system for recognizing grape diseases.

***Keywords:*** machine learning, convolutional neural networks, datasets, deep learning, ensemble learning.

## Introduction

In recent years, there has been a significant increase in the incidence of crop diseases associated with climate change, environmental pollution, and the consequences of global warming. Traditionally, farmers have relied on the intuitive diagnosis of crop diseases, but this approach has proven unreliable and ineffective. The emergence of deep learning in the field of computer technology, in particular Convolutional Neural Network (CNN) models, has opened up new ways to identify plant diseases.

Especially relevant is the introduction of such techniques in viticulture, where various diseases, such as GrapeLeafroll Disease (GLRD), can lead to a decrease in yield by 30–68 %, causing significant economic losses ranging from 25000 to 40000 USD per hectare in the absence of effective control [1]. Given these challenges and the potential of artificial intelligence, research into effective methods for recognizing grape diseases is timely and socially beneficial.

The object of research is the process of recognizing grape diseases.

The research subject is models and methods for recognizing grape diseases using artificial intelligence.

The purpose of the work is to investigate deep learning models and methods for accurately determining grape disease, improve diagnostic efficiency, and evaluate the possibilities of practical application of the results obtained.

Based on this goal, specific tasks are formulated that must be performed to achieve it:

- to analyse existing approaches and methodologies for recognizing grape diseases using artificial intelligence, to identify current trends, gaps, and areas for improvement;
- form an extensive set of images of grapes covering various diseases and health conditions, as well as provide efficient data preprocessing and artificial enlargement (augmentation) of the sample to improve the ability to generalize models;
- select appropriate neural network architectures, train and test them, and conduct appropriate experiments to ensure the most optimal performance and accuracy;

- to develop a mobile application prototype to assess the effectiveness of implemented models in real conditions and identify prospects for their practical application.

***Materials and methods of research.*** To train the selected architectures of neural network models, open sets of images of plants and grapes, and databases of plant diseases formed by research communities were used. In addition, images extracted from search engines were used. The PyTorch library and the ReactNative framework were used to create a mobile application prototype. The integration of trained models into the application is carried out using the PlayTorch library. To better generalize models and increase the variety of training data, we used image augmentation using the Augmenter library. Model training was conducted in two environments: Google Colab with access to powerful GPUs and on a local machine with an AMD Ryzen 7 7735HS processor, 16 GB of RAM, and an NVIDIA GeForce RTX 4060. The performance of the models integrated into the mobile application was evaluated on the iPhone 13 (Apple A15 Bionic) and Samsung A32 (MediaTek Helio G80).

***Analysis of recent research and publications.*** Current research on identifying and classifying grape diseases using deep learning methods covers a wide range of architectures and approaches.

The paper [2] uses the VGG16 model, which was pre-trained on the ImageNet dataset. Images of diseased and healthy grape and tomato leaves taken from the Plant Village open resource. After the pre-treatment procedure (resizing to 224×224 pixels, noise filtering, and segmentation), the image is supplemented with augmentation methods. As a result, the VGG16 model achieved an accuracy of 98.40 % for grapes and 95.71 % for tomatoes.

The study [3] focuses on using convolutional capsule networks – this approach that reflects the spatial relationships of objects using the so-called "capsules". Images of grape leaves collected from real-world sources and the PlantVillage dataset were scaled to 128×128 pixels and augmented with artificial samples. The model achieved an accuracy of 99.12 %.

Transferable learning and ensemble model creation are the focus of the study [4]. The work of pre-trained models (Vanilla, VGG16, MobileNet, AlexNet) is analyzed here. The images underwent the procedures of resizing, shuffling, and splitting the data into training, validation, and test sets. Artificial augmentation methods (rotation, scaling, mirroring) made it possible to obtain a more diverse training set. The ensemble model, which combined several architectures, achieved 100 % accuracy.

The problem of a lack of training data is considered in the work [5], where the Leaf GAN model is used to generate additional images. It includes a regression channel generator, a tight connection strategy, and instance normalization in the discriminator, which has significantly improved the quality of artificially generated images. Using 4062 original images from PlantVillage made it possible to generate 8124 new samples. Thanks to this, the classification accuracy of models such as Xception has increased to 98.70 %.

The ensemble learning approach is further developed in a study [6] where the VGG16, VGG19, and Xception models previously trained on ImageNet were fine-tuned and combined into a shared system. The use of a dense layer, a batch normalization layer, and a softmax at the output resulted in a 99.82 % accuracy.

Work [7] offers an advanced GLD-DTL approach based on the MobileNetV3 model, where the last convolutional layer is modified using batch normalization, an extraction layer, and a fully connected layer with SoftMax. The resulting version achieves an accuracy of 99.84 %.

A study on adapting DenseNet201 to a specific task showed that fine-tuning the architecture with optimal layer freezing allows for an accuracy of 98 % [8].

A study [9] presents an MS-Xception model that improves the Xception architecture by replacing the ReLU activation function with Mish, improving channel attention mechanisms, and replacing the fully bound layer with a 1×1 convolution. This optimization resulted in an accuracy of 98.61 %.

Thus, in modern research, a comprehensive set of approaches to solving the problem of recognizing grape diseases is widely used. They include a variety of types of LMD, attention-grabbing mechanisms, ensemble and transferable learning, and the application of GAN to extend and improve datasets.

At the same time, several problematic aspects are highlighted in the analysed sources. First, researchers have often used image datasets obtained in laboratory settings where a leaf sample is under direct lighting, which can impair the models' generalization ability and effectiveness in real-world field conditions. Second, most of the work focuses on recognizing diseases only from images of leaves, ignoring affected bunches of grapes or other parts of the plant. Third, insufficient attention is paid to research into models with reduced computational requirements that could run in real time on mobile devices without a permanent Internet connection in remote grape growing areas.

The synthesis of ensemble and transmission learning, the use of lightweight model architectures, as well as the formation of extensive datasets that will include both laboratory and field images of grape leaves and bunches, open the way to improving the ability of models to generalize, increase the efficiency of use in real conditions and possible implementation in mobile applications.

**Research results and their discussion**

***Processing and formation of data sets on plant and grape diseases.*** At the initial stage of our study, it was decided to create a combined dataset on diseases of different plants to use the method of transmission learning [10]. Due to this, preliminary training on a wide variety of images of plant diseases will form a basic knowledge of visual patterns of lesions in the model, which will allow for more effective adaptation of the models to a specialized dataset on grape diseases, increasing the accuracy and reliability of classification.

In our case, the combined plant disease dataset will consist of two datasets: PlantVillage and PlantDoc.

PlantVillage is a sizable open dataset containing more than 54000 images of various crops (apples, grapes, tomatoes, soybeans, corn, and others). The images were taken primarily in the laboratory, which ensures high quality and clarity of features, but may limit the ability of models to generalize results in the field [11]. Unlike PlantVillage, PlantDoc is a dataset containing images with natural backgrounds and various lighting conditions, making it more suitable for use in real-world settings. Thus, the dataset includes 2598 images covering 13 plant species [12].

The next step was to exclude classes related to disease and grape health from the aforementioned datasets to prevent possible retraining of the models in subsequent experiments.

After reconciling the shared classes obtained from PlantVillage and PlantDoc, a division into training and test subsets was performed using random sampling with a ratio of 80 % to 20 %, respectively. As a result, 36 classes of diseases and healthy plant conditions were formed.

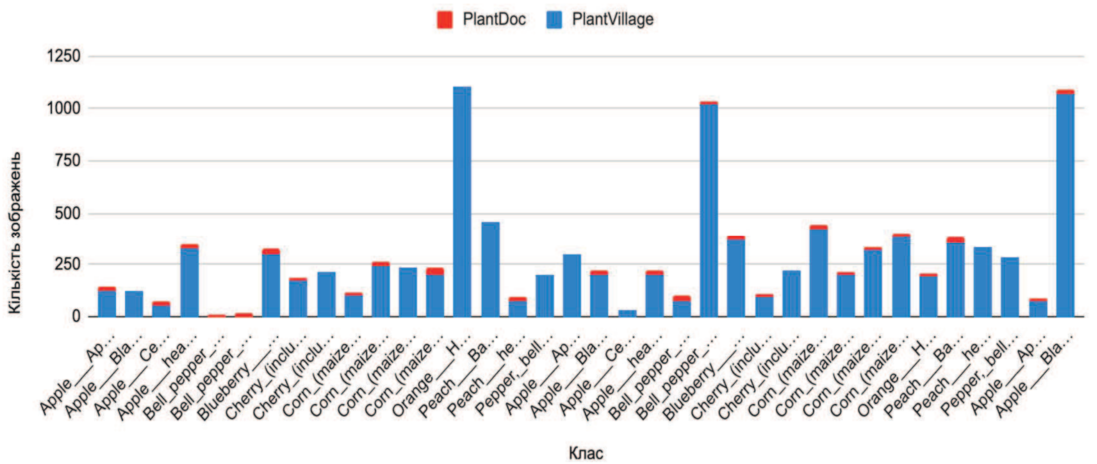



**Fig. 1.** Distribution of images by class in the test subset of the combined plant disease dataset

Since there is an imbalance between classes in the generated dataset, methods of artificially increasing the number of samples (augmentation) using the Augmentor library were applied for the training subset, which allows generating new images through various transformations: random rotation, scaling, changing contrast and brightness, adding deformations, and horizontal mirroring.

Fig. 2 shows a graph of the distribution of images in the instructional subset of the pooled dataset, where each of the 36 classes is represented by 4800 images.

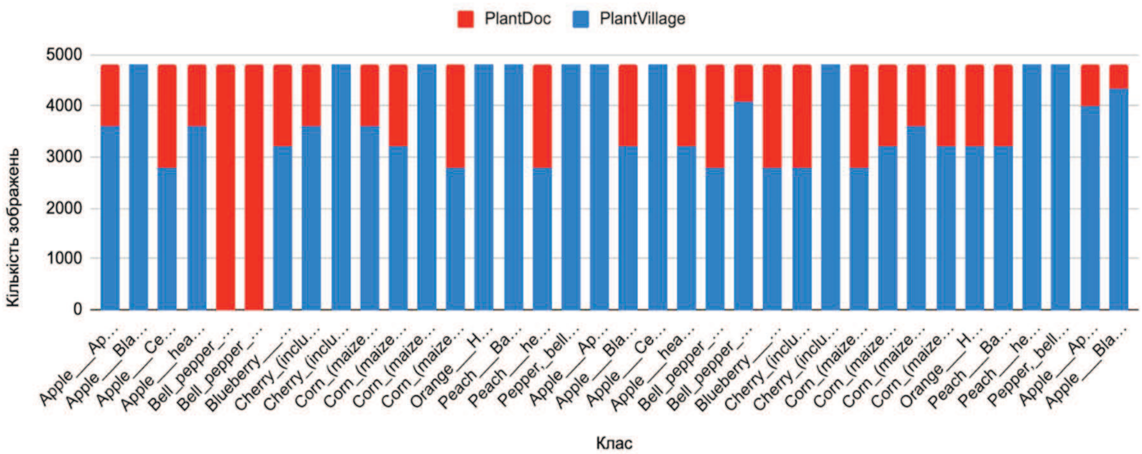



**Fig. 2.** Distribution of images by classes in the training subset of the combined plant disease dataset

By repeatedly using images from PlantDoc during augmentation, the models' generalization ability is ensured, as they will be able to classify plant diseases not only in controlled but also in real field conditions.

The following key step was the formation of a specialized grape disease dataset that combines six different sources.

The first of them, the IDADP set [13], obtained from vineyards in China, was ordered into 9 classes with a clear distribution by the affected part of the plant: the prefix "Cluster_" for diseases that are visible mainly on clusters (e. g., Cluster_Anthracnose), and "Leaf_" for diseases that are clearly manifested on the leaves (Leaf_Downy_ mildew).

The next data source was the IPM Images database [14], a collection of high-quality images often used for academic purposes. With the help of a filtering system available on the IPM Images web resource, images that specifically represent grape diseases were selected and uploaded. The application of a similar logic of distribution by clusters and leaves made it possible to obtain 10 separate classes.

The dataset [15], focused on downy mildew, initially contained 99 high-resolution images of grape leaves. To increase the information content, framing was used to highlight areas with obvious symptoms, and additional selection was carried out according to the clarity of the manifestations of the disease. As a result, 82 samples remained.

The dataset [16] created for the early detection of ESCA disease provides a two-class distribution (Leaf_ESCA and Leaf_Healthy) with almost the same number of images in each category (888 and 882, respectively).

In addition, a subset of PlantVillage [11] dedicated to grapes, which was removed in the previous stages, was used. This allows for combining the benefits of controlled shooting conditions with realistic scenarios from other sets.The dataset is supplemented with samples obtained from open search engines using web scraping to expand the variety of images further. The process included automated image compilation using the Python libraries Selenium (for browser control), BeautifulSoup (for HTML parsing and URL extraction), and Concurrent.futures (for parallel loading). Images are searched by keywords in different languages, thus covering a wider range of regions and sources. Once the data was obtained, a manual check was performed to remove irrelevant instances, and hash functions helped to identify and remove duplicates, ensuring the uniqueness and quality of the final set. This approach enabled the acquisition of an additional 16 classes of grapes and their arrangement according to similar principles.

After combining all the above sources, the extensive dataset on grape diseases is divided into 20 % test and 80 % training data. Fig. 3 shows the distribution of images by class in the test subset of the pooled grape disease dataset.

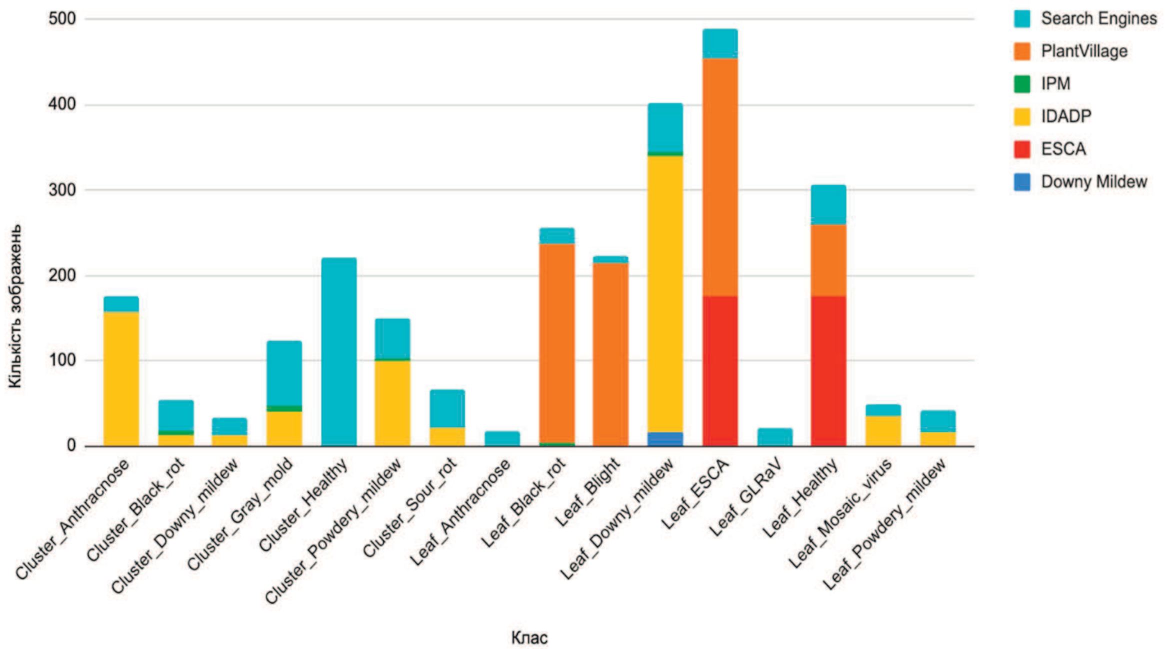



**Fig. 3.** Image distribution by class in the test subset of the combined grape disease dataset

To balance classes and improve the models' ability to generalize, the training subset was subject to augmentation, similar to the approaches used for plant diseases (Fig. 4).

As a result, the total number of classes in the combined set is 16, of which 7 classes show health and diseases that are clearly expressed on the bunches, and 9 – on the leaves.

***Convolutional neural networks with reduced computational requirements.*** For precision farming applications that involve working directly in the field on mobile or embedded devices, it is extremely important to use lightweight convolutional neural networks with reduced computational requirements.

MobileNetV3 is a modern, lightweight architecture optimized for mobile applications. It is built by searching for neural architecture (Neural Architecture Search, NAS) and improving existing ideas. The architecture is presented in two versions: MobileNetV3_Large and MobileNetV3_Small. MobileNetV3_Large provides a 3.2 % increase in classification accuracy on ImageNet compared to MobileNetV2 and reduces latency by 20 %, reaching a top-1 accuracy of 75.2 % [17].

The EfficientNet family is built on compound scaling, which balances the network's depth, width, and resolution. The EfficientNet models (B0-B7) offer varying degrees of complexity, allowing you to choose the best option depending on the available resources. In particular, EfficientNet_B1 has only 7.8 million parameters and reaches the top 1 accuracy of 79.1 % on ImageNet. It is 7.6 times smaller and 5.7 times faster than ResNet-152, providing an excellent compromise between accuracy and efficiency [18].

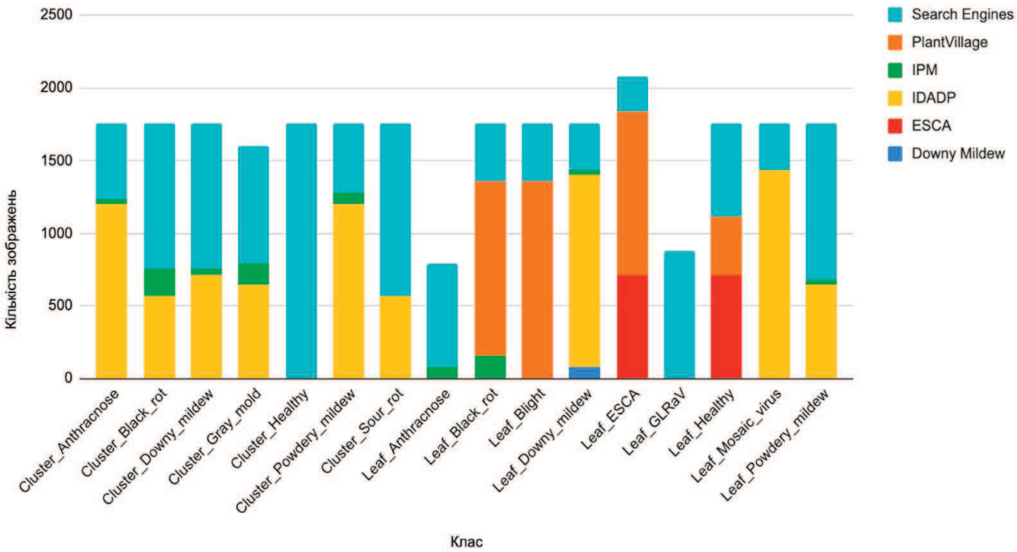



**Fig. 4.** Class distribution of images in the training subset of the combined grape disease dataset

The architecture of ShuffleNetV2 is also optimized for use on mobile platforms. The use of depthwise convolutions and minimization of fragmented operations allows Shuffle NetV2 to achieve high performance under conditions of low computational complexity. This model outperforms Mobile NetV2 and DenseNet in speed and accuracy at the same or lower cost [19].

Combining the advantages of several lightweight models using ensemble methods, such as voting ensembles, allows for increased prediction accuracy. In soft voting, each model generates a distribution of probabilities by classes, and the final forecast is formed based on the average value of these probabilities [20].

Applying soft voting to the combination of Mobile NetV3_Large, EfficientNet_B1, and ShuffleNetV2 will allow for a better balance between accuracy and prediction speed, which is a critical factor for practical use in farming.

***Training and testing neural networks on merged datasets.*** Neural network training was performed in the PyTorch environment, which provided a flexible and efficient process for declaring and using architectures. Scales previously trained on ImageNet were used to train the models on the pooled plant disease dataset.

ImageNet is a large-scale dataset (over 14 million images, 21841 grades) for evaluating computer vision models. Using a subset of 1000 classes in ILSVRC has created a generally accepted basis for comparing neural network architectures. Pre-training on ImageNet allows models to isolate key features, simplifying and accelerating their further training for specific tasks [21].

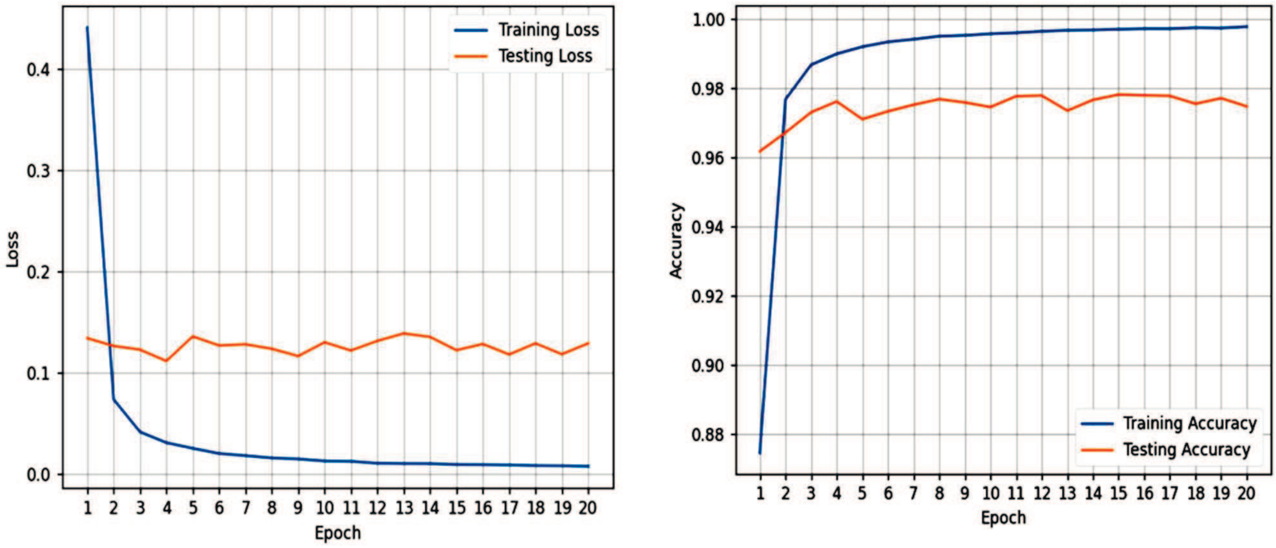
In the case of MobileNetV3_Large, which was used to classify 36 classes of plant diseases, the final layers of the network were updated, adapting them to the original space of the new dataset. Instead of the initial layer for 1000 classes, a sequence of layers (sequential) was implemented using neuronal extraction (dropout layer) with a probability of 50 % to prevent overtraining, a nonlinear activation of hard swish, and a linear layer) for classification into 36 classes. During the training, the Adam optimizer was used with a set training speed of 0.0001, which made it possible to maintain the stability of updating the scales and provide a gradual and confident convergence. The size of the data package during the training was 64 images, which made it possible to efficiently use hardware resources and maintain a balance between accuracy and speed of calculations. For 20 learning epochs, the network ran across the entire training set, after which the results were evaluated on a test subset, tracking loss and accuracy metrics (Fig. 5).

The loss graph (Fig. 5, *a*) indicated successful optimization after the first epoch and the absence of significant retraining during subsequent epochs, and the accuracy curves (Fig. 5, *b*) indicated consistently high indicators of correct classification.

A similar preparation and training process was applied to EfficientNet_B1 and ShuffleNetV2_x2, which included changing the last layers of classification according to the architecture's structure and adapting the input images' parameters and packet sizes. In Table. summarizes the results of training these three models on a plant disease dataset.

Among the options presented, EfficientNet_B1 showed slightly better performance (test accuracy of 98.14 %), while MobileNetV3_Large and ShuffleNetV2_x2 had relatively similar and high results, exceeding 97 %.

***Integration of neural networks into a mobile application and performance research.*** The mobile application was developed using the React Native framework, the choice of which was due to its cross-platform capabilities, which allow targeting iOS and Android devices using a single codebase. The project was initialized with version 0.70.15 to ensure compatibility with the PlayTorch library infrastructure, a solution to integrate PyTorch models into the ReactNative mobile environment. Not actively maintained despite this limitation, the framework remains functional and meets the requirements of this task.




*a* *b*

**Fig. 5.** MobileNetV3_Large Training and testing of the MobileNetV3_Large model for loss metrics (a) and accuracy (b) on the plant disease dataset

**Table 1.** Results of training models on the plant disease dataset

| Architecture | Input size | Number of eras | Package size | Test costs (loss) | Test accuracy (accuracy), % |
|---|---|---|---|---|---|
| MobileNetV3_Large | 224×224 | 17 | 64 | 0.1181 | 97.77 |
| EfficientNet_B1 | 240×240 | 5 | 38 | 0.1004 | 98.14 |
| ShuffleNetV2_x2 | 224×224 | 9 | 64 | 0.1376 | 97.50 |

The main components of the mobile application are shown in Fig. 6. The project is built in such a way as to ensure a clear separation of problems (separation of concerns) and ease of maintenance.

At the root level, we have the Android and iOS directories that contain the native code for a specific platform. These directories provide integration between ReactNative and native components, allowing the application to take advantage of device-specific capabilities.

The source code is organized in the src folder. The first subdirectory, assets, contains images that are used to display the results of scans of diseased or healthy grapes. The models directory stores instances of optimized PyTorch models saved in .ptl format – EfficientNet_B1, Mobile NetV3_Large, and ShuffleNetV2_x2.

The constants catalog contains a variety of configuration constants, including active OS, screen sizes, colour gamut, mapping of model classification results to class names, and more. The contexts directory contains the main React context that manages the global state and provides it for various components.

The application was tested on two devices: iPhone 13 (iOS) with Apple A15 Bionic processor and Samsung A32 (Android) with MediaTekHelio G80 processor. The results are presented in Table 2.

```
mobile
├── android
├── ios
├── src
│   ├── assets
│   │   ├── images
│   │   └── models
│   │       ├── optimized_efficientnet_b1.ptl
│   │       ├── optimized_mobilenet_v3_large.ptl
│   │       └── optimized_shufflenet_v2_x2.ptl
│   ├── constants
│   ├── contexts
│   │   └── ModelsContext.js
│   ├── hooks
│   │   └── useIsOffline.js
│   ├── navigation
│   │   └── RootStack.js
│   ├── screens
│   │   ├── CameraScreen
│   │   ├── HomeScreen
│   │   ├── ImageViewScreen
│   │   ├── ItemScreen
│   │   └── WebViewScreen
│   ├── shared
│   └── utils
├── App.js
└── index.js
```

**Fig. 6.** Main components of the mobile application structure

**Table 2.** Average prediction time of models on different devices

| Device | CPU | OS | Average forecasting time, ms | | | |
|---|---|---|---|---|---|---|
| | | | MobileNetV3_Large | EfficientNet_B1 | ShuffleNetV2_x2 | Modelansamblu |
| iPhone 13 | A15 Bionic | iOS | 834.57 | 941.82 | 858.91 | 981.91 |
| Samsung A32 | MediaTek Helio G80 | Android | 258.59 | 524.23 | 284.20 | 815.79 |

As can be seen from the Table. 2. The Samsung A32 showed faster withdrawal times compared to the iPhone 13 in all models. The difference can be attributed to differences in how each platform handles model results using the PlayTorch library. Among the models tested, Mobile NetV3_Large consistently showed the shortest prediction time on both devices, confirming its effectiveness as a lightweight model suitable for mobile deployment. EfficientNet_B1, which provides higher accuracy, requires significantly longer processing times. ShuffleNetV2_X2 demonstrated a balance between speed and precision. The ensemble model, which averages the outputs from all three networks, had the highest prediction time due to its computational complexity.

The design and functionality of the modular application window are shown in Fig. 7. After scanning the grape leaves or bunches, the app indicates in yellow the grape disease or green to classify the healthy condition of the grapes.

The user is allowed to save the result in the device's local memory. The saving process involves copying the image file to a specific directory for the identified disease. Using the react-native-fs library, the application dynamically creates a directory if it does not already exist and copies the file to it. Each saved result contains a disease name, probability, file path, number of scans, and a timestamp of the last scan. This data is updated in the application's global state to reflect the new result and is additionally stored using AsyncStorage. Once the save operation is complete, the user returns to the home screen, where the recently saved result is displayed in the list of previous scans.

*The scientific novelty of the results* obtained is the method of using an ensemble model for detecting grape disease, which is based on the formation of extended sets of images of plants and grapes for training neural networks, which provides higher diagnostic accuracy and the possibility of implementation for mobile platforms.

*Practical significance of research results* – the possibility of applying the studied models and machine learning methods to determine grape disease on a mobile device, taking into account their characteristics and the results' reliability.

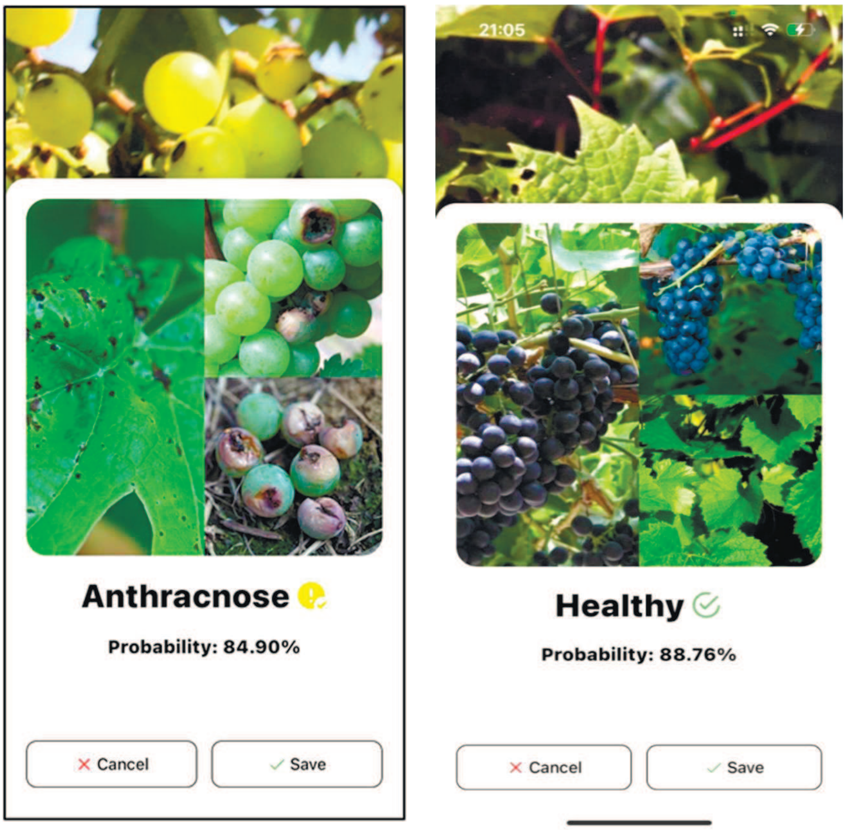



**Fig. 7.** Modal window with the obtained results of grape disease identification

## Conclusions

The definition of grape disease, which is important in viticulture, has been investigated using machine learning methods and open datasets on diseases of various plants and grapes.

Two main datasets were generated for the study: for plant diseases and for grape diseases. The dataset for plant diseases combined images from PlantVillage and PlantDoc, which were previously cleaned of grape-related classes to avoid duplication in subsequent steps, such as rotation, zoom, contrast and lighting, etc. Thanks to this, it was possible to align the number of images in each class and increase the accuracy of the models.

Three architectures were chosen to train neural network models: MobileNetV3_Large, EfficientNet_B1, and ShuffleNetV2_x2. Each of the models showed high accuracy rates at the testing stage. For models trained on the grape disease dataset, accuracy values exceeded 93 %, while precision, completeness (recall), and F1-score scores consistently exceeded 85 %. The ensemble model, which combined the results of all three architectures using the soft voting technique, achieved higher performances, in particular, the accuracy of the ensemble model was about 95 %, and the accuracy, completeness, and F1-score rose to the level of 88 %.

Based on the studied method and models, a mobile application for detecting grape diseases by the image of their leaves and bunches has been implemented, which provides the user with a scan of a new image of grapes, calculate the prognosis of the disease, save accumulated images, and view the history of cases of checking grapes for diseases.

One of the key advantages of the mobile app is its ability to work offline, which ensures that it can be used even in remote vineyards without internet access.

The test results demonstrated that the app is able to work efficiently even on devices with limited resources.

Based on the research results, conclusions are made about the possibility of using neural networks to create and operate a mobile software system for recognizing grape disease.

## References


[1] Atallah, S. S., Gomez, M. I., Fuchs, M. F., & Martinson, T. E. (2011). Economic Impact of Grapevine Leafroll Disease on Vitis vinifera cv. Cabernet franc in Finger Lakes Vineyards of New York. *American Journal of Enology and Viticulture*, 63(1), 73–79. https://doi.org/10.5344/ajev.2011.11055

[2] Paymode, A. S., & Malode, V. B. (2022). Transfer Learning for Multi-Crop Leaf Disease Image Classification using Convolutional Neural Network VGG. *Artificial Intelligence in Agriculture*, *6*, 23–33. https://doi.org/10.1016/j.aiia. 2021.12.002

[3] Diana Andrushia, A., Mary Neebha, T., Trephena Patricia, A., Umadevi, S., Anand, N., & Varshney, A. (2022). Image-based disease classification in grape leaves using convolutional capsule network. *Soft Computing*. https://doi.org/10.1007/ s00500-022-07446-5

[4] Z. Huang, A. Qin, J. Lu, A. Menon and J. Gao (2020). Grape Leaf Disease Detection and Classification Using Machine Learning. *International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, 870–877. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics50389.2020.00150

[5] Liu, B., Tan, C., Li, S., He, J., & Wang, H. (2020). A Data Augmentation Method Based on Generative Adversarial Networks for Grape Leaf Disease Identification. *IEEE Access*, *8*, 102188–102198. https://doi.org/10.1109/access.2020. 2998839

[6] Nader, A., H.Khafagy, M., & Hussien, S. A. (2022). Grape Leaves Diseases Classification using Ensemble Learning and Transfer Learning. *International Journal of Advanced Computer Science and Applications*, *13*(7). https://doi.org/ 10.14569/ijacsa.2022.0130767

[7] Yin, X., Li, W., Li, Z., & Yi, L. (2022). Recognition of grape leaf diseases using MobileNetV3 and deep transfer learning. *International Journal of Agricultural and Biological Engineering*, *15*(3), 184–194. https://doi.org/10.25165/ j.ijabe. 20221503.7062

[8] Ahmed, H. A., Hama, H. M., Jalal, S. I., & Ahmed, M. H. (2023). Deep Learning in Grapevine Leaves Varieties Classification Based on Dense Convolutional Network. *Journal of Image and Graphics*, *11*(1), 98–103. https://doi.org/ 10.18178/ joig.11.1.98-103

[9] Huang, Y., Li, N., & Liu, Z. (2023). An Enhanced Convolutional Neural Network for Accurate Classification of Grape Leaf Diseases. *Information Dynamics and Applications*, *2*(1), 8–18. https://doi.org/10.56578/ida020102

[10] Gupta, J., Pathak, S., & Kumar, G. (2022). Deep Learning (CNN) and Transfer Learning: A Review. *Journal of Physics: Conference Series*, *2273*(1), 012–029. https://doi.org/ 10.1088/1742-6596/2273/1/012029

[11] Hughes, D. P., & Salathe, M. (2016). An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv*. https://doi.org/ 10.48550/ arXiv.1511.08060

[12] Singh, D., Jain, N., Jain, P., Kayal, P., Kumawat, S., & Batra, N. (2020). PlantDoc. У *CoDS COMAD 2020: 7th ACM IKDD CoDSand 25th COMAD*. ACM. https://doi.org/ 10.1145/ 3371158.3371196

[13] YUAN, Y., & CHEN, L. (2022). An image dataset for IDADP-grape disease identification. *China Scientific Data*, *7*(1). https://doi.org/10.11922/11-6035.csd.2021.0077.zh

[14] *Fruits and Nuts : Grape*. IPM Images: The Source for Agriculture and Pest Management Pictures. https://www. ipmimages.org/browse/Areathumb.cfm?area=75

[15] Abdelghafour, F., Keresztes, B., Deshayes, A., Germain, C., & Da Costa, J.-P. (2021). An annotated image dataset of downy mildew symptoms on Merlot grape variety. *Data in Brief*, *37*, 107250. https://doi.org/10.1016/j.dib.2021. 107250

[16] Alessandrini, M., Calero Fuentes Rivera, R., Falaschetti, L., Pau, D., Tomaselli, V., & Turchetti, C. (2021). A grapevine leaves dataset for early detection and classification of esca disease in vineyards through machine learning. *Data in Brief*, *35*, 106809. https://doi.org/10.1016/j.dib.2021.106809

[17] Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L.-C., Tan, M., Chu, G., Vasudevan, V., Zhu, Y., Pang, R., Adam, H., & Le, Q. (2019). Searching for MobileNetV3. У *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE. https://doi.org/10.1109/iccv.2019.00140

[18] Tan, M. and Le, Q. V. (2019) EfficientNet: Rethinking Model Scaling for Convolutional Neural *Networks. Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, 6105-6114. https://doi.org/10.48550/ arXiv. 1905.11946

[19] Ma, N., Zhang, X., Zheng, H.-T., & Sun, J. (2018). ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. У *Computer Vision – ECCV 2018* (c. 122–138). Springer International Publishing. https://doi.org/ 10.1007/978-3-030-01264-9_8

[20] Mohammed, A., & Kora, R. (2023). A Comprehensive Review on Ensemble Deep Learning: Opportunities and Challenges. *Journal of King Saud University - Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci. 2023.01.014

[21] Jia Deng, Wei Dong, Socher, R., Li-Jia Li, Kai Li & Li Fei-Fei. (2009). ImageNet: A large-scale hierarchical image database, *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. https://doi.org/10.1109/ cvprw.2009.5206848

***Д. В. Федасюк, В. І. Дорофеєв***

*Національний університет "Львівська політехніка", Львів, Україна*

## ВИЗНАЧЕННЯ ХВОРОБ ВИНОГРАДУ ЗА ЗОБРАЖЕННЯМ З ВИКОРИСТАННЯМ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ

У роботі досліджено моделі та методи визначення хвороб винограду із використанням сучасних методів штучного інтелекту. Проаналізовано відомі методології класифікації та розпізнавання за зображеннями хвороб винограду з використання нейронних мереж. Виділено низку проблем щодо покращення результатів розпізнавання. Ключовими компонентами дослідження стали створення обширних наборів даних, навчання моделей нейронних мереж зі зниженими обчислюваними вимогами (MobileNetV3_Large, EfficientNet_B1 та ShuffleNetV2_x2), їх оптимізація для мобільного середовища та інтеграція у кросплатформний застосунок за допомогою фреймворку ReactNative. Для дослідження сформовано два основні набори даних: про хвороби рослин (PlantVillage, PlantDoc) та хвороби винограду (IDADP, IPM, DownyMildew, ESCA, PlantVillage, дані з відкритих пошукових систем). Набори було ретельно очищено та збалансовано за допомогою методів штучного збільшення даних, таких як обертання, масштабування, зміна контрасту та освітлення із використанням бібліотеки PlayTorch. Навчання та тестування нейронних мереж здійснювали із застосуванням бібліотеки PyTorch. Для підвищення продуктивності та точності моделей використано передавальне навчання. Моделі показали високі показників точності (понад 93 %) з показниками влучності, повноти та F1-score понад 85 %, а ансамблева модель, що поєднує прогнози трьох архітектур, продемонструвала точність близько 95 %.

Розроблено мобільний застосунок для роботи на пристроях iOS та Android, що забезпечує ідентифікацію хвороби винограду на основі зображень листя та грон. Застосунок орієнтований на зручність і доступність: інтуїтивний інтерфейс дає змогу здійснювати сканування у режимі реального часу, переглядати та зберігати результати, отримувати рекомендації щодо лікування. З використанням прототипу розробленої програмної системи за допомогою фреймворку React Native проаналізовано продуктивність моделей на різних типах пристроїв, що дало змогу оцінити їхню ефективність та стабільність роботи в реальних сценаріях застосування. Перевагою мобільного застосунку є можливість використання на виноградниках без доступу до інтернету.

Наведено основні результати дослідження та зроблено висновок про можливість використання нейронних мереж і ансамблевої моделі для розроблення та функціонування кросплатформної мобільної програмної системи розпізнавання хвороб винограду.

***Ключові слова:*** машинне навчання, згорткові нейронні мережі, набори даних, глибоке навчання, ансамблеве навчання.

---

**Інформація про авторів:**

**Федасюк Дмитро Васильович,** д-р техн. наук, професор, завідувач кафедри програмного забезпечення. **Email:** dmytro.v.fedasyuk@lpnu.ua; https://orcid.org/0000-0003-3552-7454

**Дорофеєв Владислав Ігорьович**, магістр кафедри програмного забезпечення. **Email:** vladyslav.dorofeiev.mpzip.2023@lpnu.ua

**Цитування за ДСТУ:** Федасюк Д. В., Дорофеєв В. І. Визначення хвороб винограду за зображенням з використанням методів штучного інтелекту. *Український журнал інформаційних технологій*. 2025, т. 7, № 1. С. 77–85.

**Citation APA:** Fedasyuk, D. V., & Dorofeiev, V. I. (2025). Identifying grape diseases by images using artificial intelligence methods. *Ukrainian Journal of Information Technology*, 7(1), 77–85. https://doi.org/10.23939/ujit2025.01.077