IRSTI 28.23.15 https://doi.org/10.58805/kazutb.v.1.26-647

RECOGNITION OF PLANT DISEASES FROM LEAF IMAGES USING MACHINE LEARNING TECHNOLOGY

[1]N. Zhumatay, [2]Sh. Akhmetzhanova, [2]A. Abduvalova, [3]K. Beisenbayeva, [3]A. Zhanarbekuly

[1]K. Kulazhanov Kazakh University of Technology and Business, Astana, Kazakhstan,
[2]Taraz Regional University named after M.Kh. Dulaty, Taraz, Kazakhstan,
[3]L.N. Gumilyov Eurasian National University, Astana, Kazakhstan,
Correspondent-author: yaphets9705@gmail.com

This article presents a Python-oriented solution for automatic classification of plants based on image analysis, applicable in agriculture, ecology and botany. Traditional plant identification methods, which require expert analysis, are often time-consuming and error-prone. The developed application uses convolutional neural networks (CNN) implemented on the basis of TensorFlow to recognize plants by their visual characteristics, which allows you to accurately and quickly classify species. An important role is played by the OpenCV library, which is used for image preprocessing, including resizing, color normalization and noise filtering, which improves classification accuracy. The model achieves an accuracy of 86.97%, which confirms its effectiveness and suitability for practical use. The system is equipped with an intuitive interface, which makes it accessible to users of different levels of training. In the future, it is planned to expand the functionality by increasing the database and introducing transfer learning methods to improve accuracy.

Keywords: plant classification, Python, TensorFlow, CNN, OpenCV, machine learning, image processing.

РАСПОЗНАВАНИЕ БОЛЕЗНЕЙ РАСТЕНИЙ ПО ИЗОБРАЖЕНИЯМ ЛИСТЬЕВ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ МАШИННОГО ОБУЧЕНИЯ

[1]Н.Ә. Жұматай, [2]Ш.Е. Ахметжанова, [2]А.Д. Абдувалова, [3]К.Ш. Бейсенбаева, [3]А. Жанарбекұлы

[1]Казахский университет технологии и бизнеса им. К. Кулажанова, Астана, Казахстан,
[2]Таразский региональный университет им. М. Х. Дулати, Тараз, Казахстан,
[3]Евразийский национальный университет имени Л.Н.Гумилева, Астана, Казахстан,
e-mail:yaphets9705@gmail.com

Данная статья представляет Python-ориентированное решение для автоматической классификации растений на основе анализа изображений, применимое в сельском хозяйстве, экологии и ботанике. Традиционные методы идентификации растений, требующие экспертного анализа, часто отнимают много времени и подвержены ошибкам. Разработанное приложение использует свёрточные нейронные сети (CNN), реализованные на базе TensorFlow, для распознавания растений по их визуальным признакам, что позволяет точно и быстро классифицировать виды. Важную роль играет библиотека OpenCV, используемая для предварительной обработки изображений, включающей изменение размеров, нормализацию цвета и фильтрацию шумов, что повышает точность классификации. Модель достигает точности 86.97%, что подтверждает её эффективность и пригодность для практического применения. Система оснащена интуитивно понятным интерфейсом, что делает её доступной для пользователей разного уровня подготовки. В будущем планируется расширение функционала за счёт увеличения базы данных и внедрения методов трансферного обучения для улучшения точности.

Ключевые слова: классификация растений, Python, TensorFlow, CNN, OpenCV, машинное обучение, обработка изображений.

МАШИНАЛЫҚ ОҚЫТУ ТЕХНОЛОГИЯСЫН ҚОЛДАНА ОТЫРЫП, ЖАПЫРАҚ КЕСКІНДЕРІ АРҚЫЛЫ ӨСІМДІК АУРУЛАРЫН ТАНУ

[1]Н.Ә. Жұматай, [2]Ш.Е. Ахметжанова, [2]А.Д. Абдувалова, [3]К.Ш. Бейсенбаева, [3]А. Жанарбекұлы

[1]*Қ. Құлажанов атындағы Қазақ технология және бизнес университеті, Астана қ., Қазақстан,*
[2]*М.Х. Дулати атындағы Тараз өңірлік университеті, Тараз қ., Қазақстан,*
[3]*Л.Н. Гумилев атындағы Еуразия ұлттық университеті, Астана қ., Қазақстан,*
e-mail:yaphets9705@gmail.com

Бұл мақала ауыл шаруашылығында, экологияда және ботаникада қолданылатын кескінді талдау негізінде өсімдіктерді автоматты түрде жіктеуге арналған Python-бағытталған шешімді ұсынады. Сараптамалық талдауды қажет ететін өсімдіктерді анықтаудың дәстүрлі әдістері көбінесе көп уақытты қажет етеді және қателіктерге бейім. Әзірленген қолданба өсімдіктерді визуалды белгілері бойынша тану үшін TensorFlow негізіндегі конволюциялық нейрондық желілерді (CNN) пайдаланады, бұл түрлерді дәл және жылдам жіктеуге мүмкіндік береді. Өлшемін Өзгертуді, түсін қалыпқа келтіруді және шуды сүзуді қамтитын кескіндерді алдын ала өңдеу үшін пайдаланылатын OpenCV кітапханасы маңызды рөл атқарады, бұл жіктеу дәлдігін арттырады. Модель 86.97% дәлдікке жетеді, бұл оның тиімділігі мен практикалық қолдануға жарамдылығын растайды. Жүйе интуитивті интерфейспен жабдықталған, бұл оны әртүрлі деңгейдегі пайдаланушылар үшін қол жетімді етеді. Болашақта деректер базасын ұлғайту және дәлдікті жақсарту үшін трансферлік оқыту әдістерін енгізу есебінен функционалдылықты кеңейту жоспарлануда.

Түйін сөздер: өсімдіктерді жіктеу, Python, TensorFlow, CNN, OpenCV, Машиналық оқыту, кескінді өңдеу.

Introduction. The rapid advancement of image recognition technology has paved the way for innovative applications across various fields, including agriculture, ecology, and environmental sciences. Accurate identification of plant species is a fundamental task in these fields, aiding in biodiversity studies, conservation efforts, and pest management. Traditional plant identification methods, relying on manual classification by experts, are often time-consuming and prone to human error. Automated plant identification systems, leveraging machine learning and computer vision, provide a scalable and efficient alternative to these traditional approaches.

This paper presents a Python-based application designed to classify plant species from user-inputted images. Utilizing convolutional neural networks (CNNs) for feature extraction and classification, the model demonstrates an effective approach to recognizing plants based on visual characteristics. The application is equipped with a user-friendly interface, allowing users to submit an image for analysis, with the model subsequently returning a probable plant species as output.

The objectives of this study include developing a robust preprocessing pipeline to optimize image quality, selecting an appropriate model architecture for high classification accuracy, and evaluating the model' s performance against existing plant databases. By leveraging modern machine learning frameworks and image processing techniques, this research seeks to contribute a practical tool for botanists, researchers, and agriculturalists in the pursuit of reliable and scalable plant identification solutions.

Materials and methods. ***Overview of Python and Libraries for Image Processing and Machine Learning***

Python is widely recognized for its versatility and ease of use, making it the preferred language for scientific and machine learning applications. Its expansive ecosystem includes numerous libraries that support high-performance computing, data analysis, and machine learning, which are essential in building efficient applications for image recognition tasks. In the domain of plant identification, Python enables complex neural network architectures and image preprocessing capabilities, which are fundamental to accurately distinguishing between species based on visual data [1].

TensorFlow, an open-source machine learning library developed by Google, is instrumental in building and deploying machine learning models. Its flexible architecture allows for the deployment of models across multiple CPUs and GPUs, making it highly efficient for training large datasets [2]. TensorFlow's Keras API is particularly useful for developing deep learning models like Convolutional Neural Networks (CNNs). CNNs are known for their ability to process and learn from images by identifying spatial hierarchies of features, such as shapes, textures, and edges, which are critical for distinguishing between different plant species. CNNs achieve this by progressively extracting feature maps from images, making them ideal for tasks that require detailed visual analysis, such as plant disease classification [3].

The use of TensorFlow also allows for implementing transfer learning, where pre-trained models (such as InceptionV3, ResNet, or VGG16) are fine-tuned on a specific dataset. Transfer learning greatly reduces training time and improves model accuracy by leveraging patterns learned from vast image datasets, which are applicable across various domains [4]. This capability makes TensorFlow a powerful tool in developing reliable models for plant identification.

OpenCV (Open Source Computer Vision Library) is an open-source library designed to support real-time computer vision and image processing tasks. OpenCV offers a wide array of tools for manipulating images, such as resizing, filtering, and color adjustments, which are crucial steps in preparing images for machine learning models [5]. For example, resizing images to a standard input size improves consistency in model training, while color normalization adjusts variations in lighting conditions, reducing the influence of environmental factors on classification accuracy [6].

Additionally, OpenCV includes advanced filtering and edge-detection techniques such as GaussianBlur, Canny Edge Detection, and Morphological Transformations, which enhance features and reduce noise in images. These techniques are particularly beneficial in plant identification, where clarity of leaf texture, edges, and shape are essential for accurate classification [7].

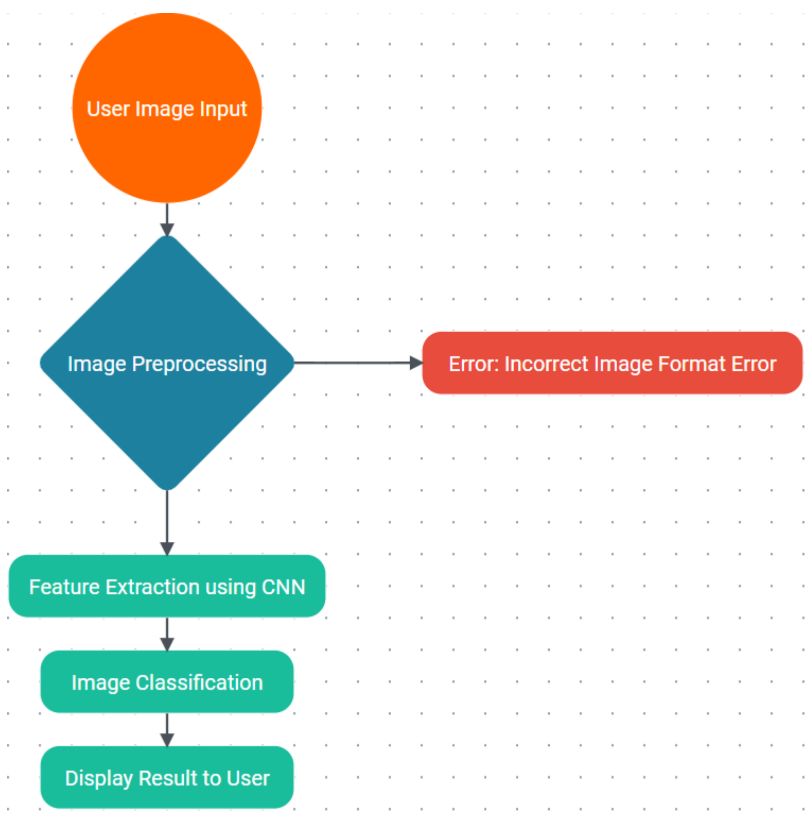

Fig. 1 - Workflow of the plant identification program

By leveraging both TensorFlow and OpenCV, Python provides a cohesive platform to manage the entire image classification pipeline, from data preprocessing to model training and deployment. The use of these libraries enables the creation of highly accurate plant identification systems that can efficiently handle real-world variations in plant appearance. The synergy between machine learning and image processing in Python not only increases the model's accuracy but also optimizes computational resources, making Python an ideal choice for plant recognition tasks [8].

Python's supportive community and comprehensive documentation for libraries such as TensorFlow and OpenCV make it accessible even for researchers with limited programming backgrounds. This accessibility, coupled with the flexibility of Python' s ecosystem, continues to drive its popularity in the scientific and machine learning communities.

Process Flow Diagram Description. The following diagram illustrates the workflow of the plant identification program, detailing each step from user input to the final result display.

This structured approach ensures accurate classification of the plant species based on the input image (Figure - 1).

The process of plant identification begins with the user uploading an image of a plant. This image can depict various parts of the plant, including leaves, flowers, or the entire plant itself. It serves as the primary input data for the program. For accurate processing, it is crucial that the uploaded image conforms to specific requirements in terms of format and resolution. Common formats accepted by the program typically include JPEG, PNG, and BMP. The resolution of the image should be sufficient to capture the details necessary for effective analysis—ideally, the dimensions should not be less than 224x224 pixels to ensure clarity during feature extraction.

Upon receiving the image, the program utilizes the OpenCV library, a powerful tool for image processing and computer vision tasks. The preprocessing phase is critical, as it enhances the image quality and optimizes the performance of the machine learning model.

During this stage, the program first checks the uploaded image for correctness in format. If the image format does not meet the accepted criteria, an error message—"Incorrect Image Format Error"—is displayed. This prompt guides the user to upload a compatible file, ensuring that the input data is valid before proceeding further.

Resizing: The image is resized to specified dimensions (commonly 224x224 pixels). This standardization is crucial to maintain consistency across all inputs, which helps to avoid potential scaling issues that could adversely affect classification accuracy.

Color Normalization: This step adjusts the image for variations in lighting and shadows. Color normalization helps mitigate the influence of external factors, such as differences in illumination or background scenery, thereby creating a more uniform basis for analysis.

Noise Filtering: Noise can often obscure important features in an image, so filtering techniques are applied to eliminate artifacts. Methods such as Gaussian blur or median filtering are employed to smooth the image, which enhances the model' s ability to detect relevant features accurately.

Once the image preprocessing is complete, the preprocessed image is passed to a Convolutional Neural Network (CNN) model, which is implemented using TensorFlow. The CNN is designed to automatically extract hierarchical features from the image. These features include shapes, textures, and edges that are characteristic of different plant species.

This feature extraction process is fundamental for the subsequent classification task. By identifying and isolating unique features, the CNN enables the model to differentiate between various species of plants. Each layer of the CNN extracts increasingly complex features, allowing for a deep understanding of the visual data presented in the image.

After feature extraction, the CNN analyzes the gathered features and classifies the image based on their correlations with specific plant types. The model calculates probabilities for each potential class (plant species) and selects the class with the highest probability as the final classification result.

For instance, if the model analyzes an image of a plant and identifies key features indicative of lavender, it may conclude that the image corresponds to the "Lavender" class with a confidence level of 90%. This confidence score reflects the model's certainty about its classification, which is critical for user trust and usability.

Once the classification is complete, the program displays the results to the user. This output includes the name of the identified plant and may also provide additional information, such as its scientific name, common characteristics, or a brief description of its ecological significance.

In cases where the model' s confidence level is low-indicating uncertainty about the classification-the program offers additional guidance. The user might receive suggestions to upload a clearer image or to try a different angle that better captures the plant' s distinguishing features. This interactive element not only improves the user experience but also enhances the likelihood of obtaining accurate results in subsequent attempts.

Image Preprocessing. Resizing is a fundamental preprocessing step in image analysis, particularly for machine learning applications. The process involves changing the dimensions of an image to a standard size that is compatible with the model' s input requirements. For example, many deep learning models, including Convolutional Neural Networks (CNNs), often expect input images to be of specific dimensions-commonly 224x224 pixels.

The importance of resizing lies in its ability to maintain consistency across all images fed into the model. When images of varying sizes are used, it can lead to discrepancies during feature extraction, making it challenging for the model to learn effectively. Uniformity in size helps in stabilizing the learning process, enabling the model to focus on the actual features rather than adjusting to differences in image dimensions. Additionally, resizing reduces the computational load, allowing for faster processing and improved efficiency during training and inference stages [9].

Filtering techniques are employed to enhance image quality by reducing noise and artifacts that can obscure significant features. Common filtering methods include:

1. Gaussian Blur: This technique smooths the image by averaging pixel values in a neighborhood defined by a Gaussian function. It helps in eliminating high-frequency noise and reduces detail in the image, which can otherwise interfere with the learning process. By suppressing noise, the model can better identify critical features that contribute to accurate classification [10].

2. Median Filtering: Unlike Gaussian blur, median filtering replaces each pixel value with the median of the neighboring pixels. This method is particularly effective at removing salt-and-pepper noise while preserving edges, which are essential for accurate feature extraction. The preservation of edges aids the model in recognizing the contours and shapes that are characteristic of different plant species [11].

By applying these filtering techniques, the preprocessing stage enhances the clarity of the images, allowing the model to learn from more representative data. A cleaner input leads to more reliable feature extraction and ultimately contributes to improved classification accuracy.

Normalization is a process that adjusts the pixel values of an image to ensure consistent representation across different lighting conditions and backgrounds. This method is crucial for reducing the impact of variability in image acquisition, which can arise from different sources of illumination or environmental conditions.

Color normalization techniques may involve transforming the image' s color space (e.g., converting from RGB to HSV) or adjusting pixel intensities to fit a specified range. This process helps in balancing the overall brightness and contrast of the image, enabling the model to focus on intrinsic features rather than extraneous visual noise[12].

Normalization plays a pivotal role in enhancing model accuracy by ensuring that the neural network receives input data that is representative of the actual features it needs to learn. By mitigating the effects of inconsistent lighting and color variations, normalization allows the model to generalize better from training data to unseen instances, thereby improving its performance in real-world applications.

In summary, the preprocessing of images through resizing, filtering, and normalization is critical for enhancing the accuracy of machine learning models, especially in the context of plant identification. Each of these techniques contributes to creating a more uniform and representative dataset, which in turn enables the model to extract meaningful features and make accurate classifications. The importance of these preprocessing methods cannot be overstated, as they lay the groundwork for effective learning and reliable performance in image classification tasks.

Results and discussion. ***Machine learning model for plant disease classification***

In the realm of agricultural technology, the ability to accurately diagnose plant diseases is crucial for maintaining healthy crops and ensuring food security. To achieve this, we developed a machine learning model based on a Convolutional Neural Network (CNN) architecture, specifically saved as cnn_plant_disease_model.keras. CNNs are particularly well-suited for image recognition tasks due to their ability to automatically extract and learn features from images.

The CNN architecture utilized in this study comprises several convolutional layers, each followed by activation functions and pooling layers. This structure allows the model to learn hierarchical representations of the input data, enabling it to capture intricate details in plant images. By employing multiple layers, the CNN can detect features ranging from simple edges and textures in the early layers to complex patterns in the deeper layers.

For effective model training, a comprehensive dataset was essential. The dataset consisted of thousands of images of various plant species, each labeled with the corresponding disease status. The images were collected from diverse sources to ensure variability in conditions, such as lighting and angles, thus enhancing the model' s robustness.

To further improve the model' s performance, data augmentation techniques were applied. These techniques included random rotations, zooming, flipping, and brightness adjustments. By artificially increasing the diversity of the training dataset, we aimed to help the model generalize better to unseen data, thereby reducing the risk of overfitting.

Prior to training, all images underwent preprocessing to optimize them for input into the CNN. The preprocessing steps included resizing images to a standard dimension of 224x224 pixels and normalizing pixel values to a range between 0 and 1. This standardization is vital for improving the model' s convergence speed during training.

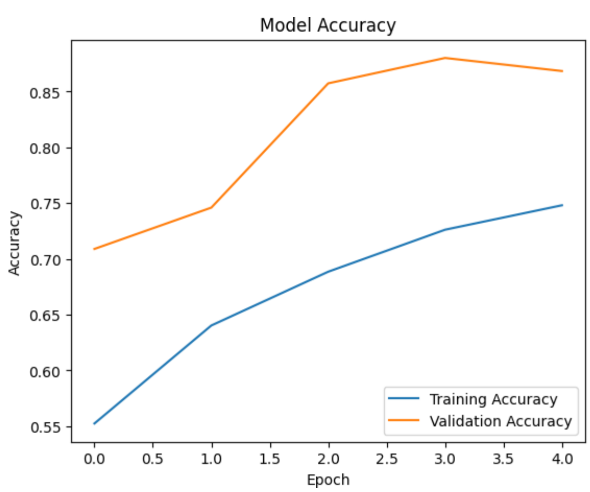

Fig. 2 - Training and validation accuracy over epochs

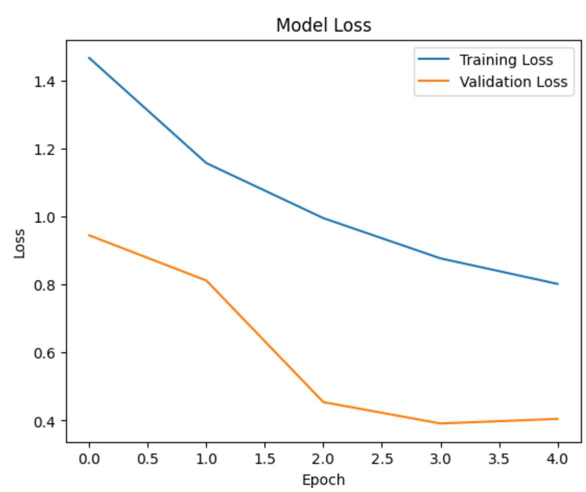

Fig.3 - Training and validation loss over epochs

The model was trained using a robust dataset split into training and validation sets, with 80% of the data allocated for training and 20% for validation. The training process involved multiple epochs, with a batch size of 32. This configuration allowed the model to learn from a sufficient number of samples in each iteration while maintaining efficient computational performance.

We employed the Adam optimizer, a popular choice in deep learning for its adaptive learning rate capabilities. The loss function used was categorical cross-entropy, suitable for multi-class classification tasks. This combination of optimizer and loss function facilitated efficient training and convergence of the model.

The ”Training and validation accuracy over epochs” graph illustrates the model’ s performance in terms of accuracy throughout the training process. On the X-axis, we have the number of Epochs, while the Y-axis represents the Training Accuracy and Validation Accuracy Figure - 2).

As the training progresses through the epochs, we observe a significant increase in both training and validation accuracy. Initially, the model demonstrates relatively low accuracy, which gradually improves as it learns from the training dataset. By the end of the training, the model achieves an accuracy of 86.97% (or 0.8697), indicating that it can correctly classify the majority of plant images in the validation dataset. This high accuracy reflects the model’s effectiveness in capturing the relevant features necessary for plant disease classification.

The gap between training accuracy and validation accuracy should be monitored closely. If the training accuracy is significantly higher than the validation accuracy, it may indicate overfitting. However, in this case, the model appears to maintain a good balance between the two, suggesting effective generalization to unseen data.

The ”Training and validation loss over epochs” graph presents the loss values recorded throughout the training process. The X-axis represents the number of Epochs, while the Y-axis displays the Training Loss and Validation Loss (Figure - 3).

Initially, both training and validation loss values are high, indicating that the model’ s predictions deviate significantly from the actual labels. However, as training progresses, both loss values decrease steadily. The Training Loss reaches a value of 0.4000, demonstrating that the model’ s predictions increasingly align with the actual data as it learns. A lower loss value indicates a better performance in classification tasks, suggesting that the model has effectively minimized the error in its predictions.

The convergence of training and validation loss towards the end of the training process further supports the model’ s stability and its capability to generalize well. If the validation loss starts to increase while training loss continues to decrease, it would be a sign of overfitting; however, this scenario is not observed in our results.

To illustrate the model’ s performance, we generated accuracy and loss curves during the training process. [Insert the accuracy and loss graph here.] These visualizations depict the model’ s learning trajectory, showcasing how accuracy improved and loss decreased over epochs. Observing these trends allows us to assess the effectiveness of the training regimen.

The developed CNN model, cnn_plant_disease_model.keras, holds significant promise for practical applications in agriculture. By enabling quick and accurate identification of plant diseases, this model can assist farmers and agricultural specialists in making timely decisions to manage crop health. This capability can ultimately lead to improved yield and reduced economic losses due to disease outbreaks.

While the current model demonstrates commendable performance, several avenues for future research exist. Exploring more advanced architectures, such as deeper CNNs or transfer learning with pre-trained models, could yield further improvements in accuracy. Additionally, expanding the dataset to include more diverse plant species and diseases would enhance the model’ s applicability across different agricultural contexts.

In summary, the CNN model developed

for plant disease classification, saved as cnn_plant_disease_model.keras, achieved an impressive accuracy of 86.97% with a loss of 0.4000. This indicates a well-optimized model capable of effectively identifying plant diseases from images. As agricultural challenges continue to evolve, the integration of machine learning technologies like this model offers promising solutions for sustainable farming practices.

User interface and interaction in Plant disease classification application. The plant disease classification application is designed to provide a seamless user experience while allowing users to accurately identify various plant species and diagnose potential diseases.

The user interface (UI) is thoughtfully crafted to be intuitive, facilitating easy navigation for individuals with varying levels of technical expertise (Figure - 4).

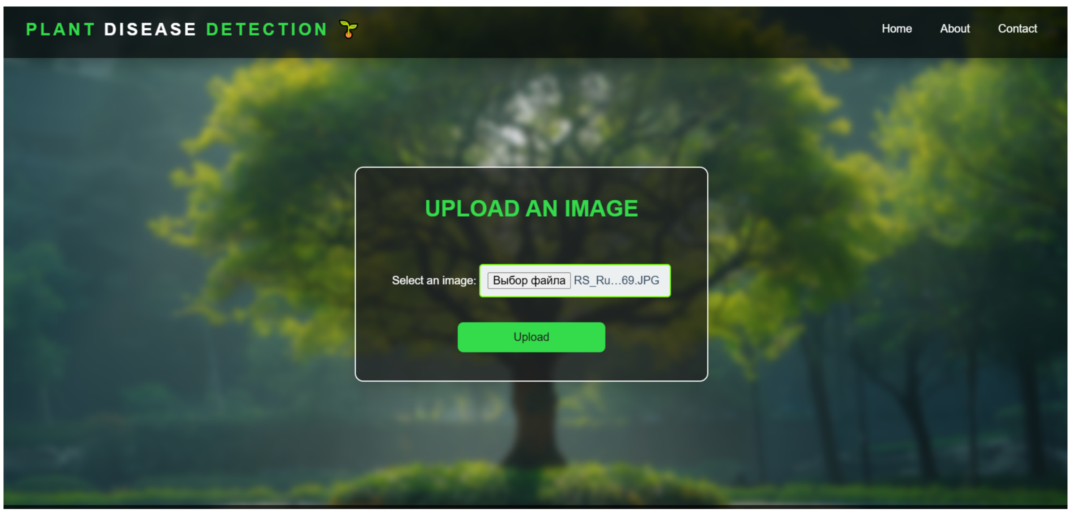

Fig. 4 - Home page

The application guides users through a straightforward process, beginning with the launch of the program on their device. Upon accessing the main screen, users encounter a prominent prompt to upload an image of a plant. This image can depict a leaf, flower, or the entire plant, serving as the crucial input data for the classification model.

Users can initiate the process by clicking the ”Upload Image” button, which opens a file dialog for selecting an image. The application supports multiple image formats, including JPEG, PNG, and BMP, ensuring compatibility with commonly used files. Once an image is selected, the program conducts a validation check to verify that the uploaded file adheres to the necessary format and resolution specifications. If the image format is deemed incorrect, the application promptly displays an ”Incorrect Image Format Error” message, guiding the user to upload a compatible file.

After confirming the validity of the uploaded image, the program utilizes the OpenCV library to preprocess the image. This stage includes resizing, color normalization, and noise filtering, enhancing the image quality for optimal performance of the model. A progress indicator may appear during this phase, reassuring users that the processing is underway. Upon completion of the image processing and classification, the application presents the results prominently on the screen. Users receive information about the identified plant species, including both common and scientific names. Additionally, the program may provide supplementary details, such as characteristics, potential diseases, and care instructions.

The model’ s confidence level in its classification is also communicated to the user. For example, if the program identifies a plant as “Lavender” with a confidence level of 90%, this information helps users assess the reliability of the result. Users are encouraged to provide feedback on the classification results. If they believe the classification is incorrect, they can submit their observations, contributing to the model's continuous improvement.

To ensure a pleasant user experience, the interface incorporates several interactive features. Accessibility features are designed with inclusivity in mind, offering options such as screen reader compatibility, text-to-speech for results, and adjustable font sizes to cater to diverse user needs. A dedicated help section is available, providing comprehensive information on how to navigate the application, troubleshoot common issues, and interpret results. This resource is particularly beneficial for first-time users unfamiliar with the technology.

The option to create user profiles allows individuals to save uploaded images and classification results for future reference. This feature is advantageous for tracking the health of plants over time and comparing different outcomes. The accuracy and performance of the plant disease classification model are paramount to its usability. The model achieved an accuracy of 86.97% and a loss value of 0.4000 during training, reflecting its capability to reliably classify a wide range of plant species.

The accuracy of the model is determined by its ability to correctly classify instances from a validation dataset. An accuracy of 86.97% signifies that the model can accurately identify the majority of plant images, which is essential for practical applications in agriculture where timely and precise diagnosis of plant diseases can significantly impact crop management strategies. Beyond accuracy, additional performance metrics such as precision, recall, and F1-score can be computed to provide a more comprehensive understanding of the model's performance. These metrics are particularly useful in scenarios where class imbalances exist.

A confusion matrix can be generated to visualize the model' s classification performance across various plant species. This tool helps identify specific species that are frequently misclassified, enabling targeted improvements in model performance. The model has been trained to recognize a diverse array of plant species, each exhibiting unique characteristics. For example, it can identify Lavender (Lavandula angustifolia), known for its fragrant purple blooms, which is commonly used in aromatherapy and culinary applications. The model can accurately identify tomato plants (Solanum lycopersicum), a widely cultivated vegetable, by their characteristic green foliage and red fruits. It also recognizes roses (Rosa spp.), known for their beauty, and corn (Zea mays), a staple crop identifiable by its tall stalks and long, narrow leaves. Additionally, the model can recognize basil (Ocimum basilicum), a popular culinary herb identifiable by its broad, green leaves and aromatic scent.

These examples illustrate the practical applicability of the model in identifying common plant species, providing valuable insights for users interested in gardening, agriculture, or plant care.

Conclusions. In conclusion, the development of the plant disease classification model represents a significant advancement in the field of agricultural technology. By leveraging cutting-edge machine learning techniques, particularly Convolutional Neural Networks (CNNs), the model demonstrates a commendable accuracy of 86.97% and a loss of 0.4000.

These results highlight its potential as a reliable tool for identifying plant diseases, thereby assisting farmers and agricultural professionals in making informed decisions about crop management and disease mitigation.

However, as outlined in the discussion of limitations, several factors can affect the model' s performance in real-world applications. Variability in image quality, environmental conditions, and the limitations of the training dataset can lead to misclassifications. Recognizing these challenges is essential for driving further improvements to the model' s robustness and accuracy.

Future enhancements such as data augmentation, the incorporation of user feedback, and expanding the training dataset are crucial for refining the model. Furthermore, adopting a continuous learning approach will ensure that the model remains relevant in a rapidly evolving agricultural landscape, adapting to new plant diseases and variations.

Overall, while the current model is a promising step forward, ongoing research and development are needed to unlock its full potential. By addressing the limitations and exploring innovative strategies for improvement, the plant disease classification model can play a vital role in promoting sustainable agricultural practices and enhancing food security worldwide.

References

1. Oliphant, T. E. (2007). Python for scientific computing //Computing in Science & Engineeringю- 2007.-Vol. 9(3).- P.10-20.DOI 10.1109/MCSE.2007.58

2. Abadi, M., Barham, P., Chen, J., et al. (2016). TensorFlow: A system for large-scale machine learning. Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation. - 2016.-Vol.1.-P.265-283. DOI 10.48550/arXiv.1605.08695

3. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks//Advances in Neural Information Processing Systems.-2017.-Vol. 60(6).- P.84-90. DOI 10.1145/3065386 .

4. Pan S. J., & Yang Q. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering.- 2010.-Vol 22(10) - P.1345-1359. DOI 10.1109/TKDE.2009.191

5. Bradski G. The OpenCV Library//Doctor Dobbs Journal.-2000.- Vol. 25(11)

https://www.researchgate.net

6. Bishop C. M. Pattern Recognition and Machine Learning. Springer. 2006.- 776 p.- ISBN 978-0-387-31073-2. https://link.springer.com

7. Gonzalez R. C., Woods, R. E. Digital Image Processing (3rd Edition). Pearson.-2008.- 976 p.

ISBN 0-13-168728-x 978-0-13-168728-8

8. Zhang L., Zhang L.,Du B. Deep learning for remote sensing data: A technical tutorial on the state of the art. IEEE Geoscience and Remote Sensing Magazine.-2016. Vol. 4(2). - P.22-40.

DOI 10.1109/MGRS.2016.2540798

9.A. Garg A Survey on Content Aware Image Resizing Methods. //KSII Transactions on Internet and Information Systems. - 2020.- Vol.14(7).- P.2997-3017.DOI 10.3837/tiis.2020.07.015

10. R.Gonzalez and R. Woods Digital Image Processing, 4th ed. Pearson.-2018.-1024 p.

ISBN 978-1-292-22304-9

11. John C. Russ, F. Brent Neal The Image Processing Handbook - 2017. -1056 p.

ISBN1138747491, 9781138747494

12. D.P Oppenheim and R. W. Schafer, Digital Signal Processing, 3rd ed. Prentice Hall.- 2009.-P.1108 ISBN-13: 978-0-13-198842-2

Information about the author

Zhumatayn N. - Master' s student, K. Kulazhanov Kazakh University of Technology and Business, Astana, Kazakhstan, e-mail: zhumatayn@gmail.com;

Akhmetzhanova Sh. - acting associate professor, Taraz regional university named after M. KH. Dulaty, Taraz, Kazakhstan, e-mail: she.akhmetzhanova@dulaty.kz;

Abduvalova A. - acting associate professor, Taraz regional university named after M. KH. Dulaty, Taraz, Kazakhstan, e-mail: abduvalova_ad@mail.ru;

Beisenbayeva K. - PhD, Senior lecturer, L.N. Gumilyov Eurasian National University, Astana, Kazakhstan, e-mail: bei_kkulys@mail.ru

Zhanarbekuly A. - Teacher, L.N. Gumilyov Eurasian National University, Astana, Kazakhstan, e-mail: yaphets9705@gmail.com

Сведения об авторах

Жұматай Н. **-** магистрант, Қ. Құлажанов атындағы Қазақ технология және бизнес университеті**,** Астана, Қазақстан, e-mail: zhumatayn@gmail.com;

Ахметжанова Ш. - к.т.н.,и.о.доцента, Таразский региональный университет имени М.Х.Дулати, Тараз, Казахстан, e-mail: she.akhmetzhanova@dulaty.kz ;

Абдувалова А**.**- к.т.н., и.о. доцента Таразский региональный университет имени М.Х.Дулати, Тараз, Казахстан, e-mail: abduvalova_ad@mail.ru

Бейсенбаева К. **-** PhD, Старший преподаватель кафедры криптологии, Евразийского национального университета им.Л. Н. Гумилева, Астана, Казахстан, e-mail: bei_kkulys@mail.ru

Жанарбекұлы А. - преподаватель, Евразийского национального университета им.Л. Н. Гумилева, Астана, Казахстан, e-mail: yaphets9705@gmail.com