JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Real-time Plant Disease Classification Using OpenCV and Machine Learning Models

[1] Dr. Bhrantav Vora, [2] Palak Navneetkumar Patel
[1]Assistant Professor, [2]Student of MCA Semester 4
[1]MCA Department,
[1]LDRP - ITR, Gandhinagar, Gujarat, India

Abstract : Plant diseases can greatly hinder crop development and impact food supply, which makes early detection a critical step in modern agriculture. This project introduces a technique that combines image processing with machine learning to identify plant diseases at an early stage. The process begins by capturing images of plant leaves. These images are then enhanced using OpenCV tools to reduce noise and improve clarity. From the cleaned images, important visual features are extracted, highlighting patterns that indicate specific plant diseases. These features are used to train and test a machine learning model capable of recognizing and classifying various types of infections. Our results indicate that this method can successfully detect several plant diseases with reliable accuracy. With this system, farmers can quickly identify problems and take timely action, ultimately protecting crops and supporting better yields.

__ **Keywords**— deep learning; convolutional neural networks; inception V3; inception ResNet V2

I. INTRODUCTION

Automated Detection and Classification of Plant Diseases Using Image Processing and Deep Learning. Agriculture remains a cornerstone of global food security, with healthy crops being vital for sustaining high yields. However, plant diseases continue to threaten agricultural productivity, often resulting in significant economic losses and disruptions in food supply chains. To mitigate these risks, early and accurate detection of plant diseases is essential, allowing timely interventions that minimize crop damage and enhance food production.

Traditional approaches to diagnosing plant diseases rely heavily on expert inspection. While effective in some cases, these methods are often slow, labor-intensive, and subject to human error. With recent advancements in computational technologies, automated systems powered by artificial intelligence (AI) and image processing have emerged as promising alternatives for large-scale and real-time plant disease detection.

This research explores a hybrid approach that combines image-processing techniques with machine learning algorithms for the automatic identification and classification of plant diseases from leaf images. OpenCV is employed for preprocessing tasks such as noise reduction and image enhancement, as well as for extracting meaningful visual features. These features are then used to train machine learning models capable of distinguishing between healthy and diseased plants.The primary goal of this work is to build an accurate and efficient system that can assist farmers by providing timely diagnoses based on visual symptoms, thus enabling early and effective disease management. The broader impact of this solution includes reducing crop losses, boosting agricultural efficiency, and contributing to food security on a global scale.

The advancement of AI, particularly due to increased computational capabilities brought about by graphics processing units (GPUs), has significantly accelerated the application of machine learning in agriculture. Earlier efforts using traditional computer vision focused on identifying disease-affected regions and estimating the extent of infection. These systems have since evolved with the advent of deep learning, which enhances image analysis through automated feature extraction and classification.

Convolutional Neural Networks (CNNs), introduced in the early 1990s, have become a powerful tool in this context. They surpass traditional methods by learning hierarchical representations of image features, improving both the speed and accuracy of classification tasks. A CNN typically includes multiple layers—such as input, convolutional, pooling, and fully connected layers—that work together to extract and learn patterns from images.

Due to their versatility and strong performance, CNNs are now widely used across various domains, including face recognition, natural language processing, time-series forecasting, and notably, and precision agriculture. In plant disease detection, CNNs have demonstrated high accuracy and scalability, making them an ideal choice for developing intelligent diagnostic tools for farmers.

II.Literature Review

Plant disease diagnosis has undergone a significant transformation, shifting from traditional expert-based inspections to automated methods powered by deep learning. Conventional machine learning algorithms such as Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) initially gained popularity but required manual feature extraction, which was both labor-intensive and error-prone. The rise of deep learning—particularly Convolutional Neural Networks (CNNs)—revolutionized the field by enabling automatic feature extraction, leading to improved accuracy and scalability.

The availability of large annotated datasets, such as PlantVillage, and the use of advanced imaging technologies, including RGB, multispectral, and hyperspectral sensors, have further enhanced the capabilities of plant disease detection systems. However, several limitations persist, such as insufficient real-world data, reduced model performance under uncontrolled conditions, and the challenge of estimating disease severity effectively. To address these issues, current research focuses on enhancing model robustness, supporting multi-disease detection, and embedding AI-powered tools into real-time agricultural applications [1].

The use of deep learning, especially CNNs, has shown great potential in transforming how plant diseases are identified. Traditional methods like visual assessments and chemical diagnostics are slow and often inaccurate. AI-powered approaches, particularly those using CNNs, now allow for rapid, image-based detection. Studies such as those by Mohanty et al. applied pre-trained CNNs to the PlantVillage dataset, while Fan et al. enhanced Inception V3 for detecting diseases in crops like apples and coffee. However, these models, mostly trained on clean lab datasets, often struggle to generalize to noisy, real-world farming environments. To overcome this limitation, the current study focuses on tomato leaf disease detection using both the PlantVillage dataset and field-collected data. By employing transfer learning with architectures like Inception V3 and Inception ResNet V2, the aim is to create a model with stronger generalizability and field reliability [2].

Deep learning has consistently outperformed traditional machine learning in the context of tomato disease detection. Pretrained CNN models such as VGG16, ResNet, and MobileNetV2 have achieved classification accuracies exceeding 97%. Some hybrid approaches further enhance accuracy by combining deep learning with traditional ML techniques, such as integrating attention-based dilated CNNs with logistic regression. Despite these advancements, challenges like dataset variability, overfitting, and high computational requirements still hinder widespread deployment. To address these, a new architecture called EffiMob-Net has been proposed. It combines efficient deep learning models to deliver both high accuracy and real-world usability in tomato disease recognition tasks [3].

Research has also highlighted the benefits of combining CNNs with methods like Principal Component Analysis (PCA), Support Vector Machines (SVM), and Faster R-CNN to boost classification accuracy and reduce computational cost. GANs (Generative Adversarial Networks) have been explored to balance datasets by generating synthetic examples of underrepresented disease classes. A notable hybrid approach, PCA DeepNet, integrates PCA for feature selection, CNN for disease classification, and Faster-RCNN for object detection, addressing both classification and localization tasks effectively [4].

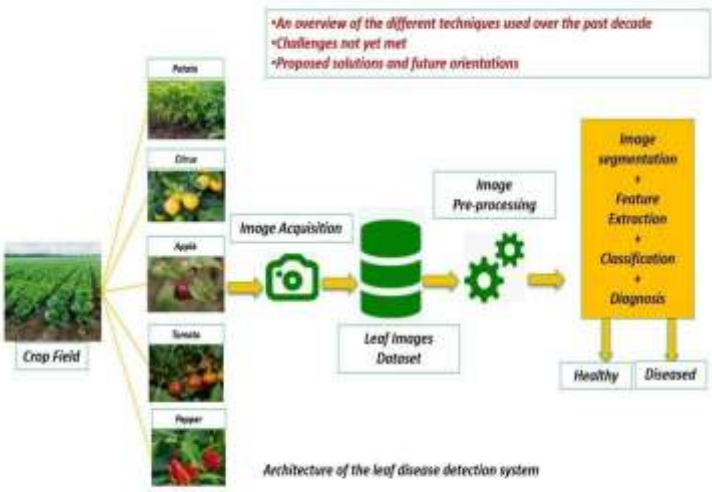
Dataset quality remains a critical factor in model performance. While datasets like PlantVillage offer a large number of labeled images (over 54,000), they lack the complexity of real-world backgrounds. The smaller PlantDoc dataset suffers from inconsistent annotations. The recently introduced FieldPlant dataset, with over 5,000 images and expert-validated annotations for 27 diseases, captures more realistic field conditions. Still, models like EfficientNet and MobileNetV2, though effective in lab conditions, often struggle with complex backgrounds, poor lighting, and variable leaf orientations found in field environments. Object detection networks like YOLO and U-Net have shown promise in filtering out background noise, though fine-grained classification remains a challenge [5].

In addition to standard classification tasks, newer studies have explored ensemble methods, transfer learning, and deep belief networks (DBNs) to improve accuracy. These techniques often rely on large, well-annotated datasets and significant computational power. A key challenge is ensuring that models generalize across different plant species, environmental conditions, and geographical regions. Continued progress in model generalization, improved segmentation, and dataset diversity are essential for real-world application [6].

Another emerging area is the use of drones for large-scale plant disease monitoring. Drones offer real-time, wide-area coverage and can be equipped with advanced sensors for detailed imaging. Literature reviews in this space help map the landscape of existing work, identify key contributions, and avoid redundancy, guiding future research directions [7].

Finally, recent analyses of CNNs in assessing disease severity provide insights into model accuracy and reliability. Studies have addressed methodological issues such as dataset size, evaluation metrics, and annotation consistency. These insights are crucial for refining CNN-based models and improving their application in precision agriculture [8].

Figure 1. Architecture of the leaf disease detection System



III. METHODOLOGY

The primary goal of this system is to develop an efficient and responsive plant disease detection model capable of not only identifying the disease affecting a plant but also recommending suitable pesticides for treatment. To achieve this, the system is divided into two main phases: the training phase and the testing phase.

- Training Phase:
  This phase involves three key steps:

1. Image Acquisition: Collecting images of plant leaves, both healthy and diseased.
2. Image Preprocessing: Enhancing image quality through techniques such as resizing, noise reduction, normalization, and augmentation to improve model robustness.

3. Model Training: Using a Convolutional Neural Network (CNN) to learn features from the preprocessed images and associate them with specific plant diseases.

Figure 2. Sample images in the dataset

- Testing Phase:
  This phase is responsible for deploying the trained model and consists of:

1. Image Acquisition and Preprocessing: Capturing and preparing new leaf images in a similar way as the training phase.
2. Classification: The CNN model classifies the input image by identifying the disease present.
3. Disease Identification and Remedy Suggestion: The system outputs the name of the detected disease and suggests an appropriate pesticide or treatment method for it.

1.1 Dataset

The effectiveness of any plant disease detection model largely depends on the quality and diversity of the dataset used for training. A typical plant disease dataset consists of images of plant leaves labeled with their respective health status—either healthy or showing symptoms of a particular disease.

The PlantVillage dataset is one of the most widely used and comprehensive datasets in this domain. It includes over 87,000 images, covering 38 different diseases across 14 plant species. The images are mostly captured in controlled environments, making them ideal for initial training phases.

Other datasets, such as the AI Challenger Agricultural Dataset, offer images taken in more realistic field conditions, helping to improve the model's generalizability and robustness in real-world applications.

Before being used for training, images typically undergo several preprocessing steps, including:

- Resizing to a consistent resolution
- Brightness and contrast adjustments
- Image augmentation (e.g., rotation, flipping, scaling) to improve model generalization

Despite their usefulness, these datasets come with challenges such as:

- Class imbalance, where certain diseases have significantly more images than others
- Varying lighting conditions in real-world images
- Background noise and occlusions, which can confuse the model

These challenges highlight the importance of using diverse and well-annotated datasets and implementing preprocessing and augmentation techniques to enhance model performance [1].

1.2 Dataset Collection and Description

The quality and structure of the dataset significantly influence the performance and accuracy of any plant disease detection model. In this project, the dataset primarily includes high-resolution images of plant leaves, both healthy and affected by various diseases. Each image is annotated with its respective disease label or classified as healthy, providing essential ground truth for training supervised learning models.

One of the most widely adopted datasets for this purpose is the PlantVillage dataset, which contains around 87,000 images across 14 different crop species and 38 distinct disease classes. The dataset is neatly arranged into subdirectories, with each folder corresponding to a specific disease or a healthy condition. For example, categories are labeled as Apple___Apple_scab, Tomato___Late_blight, Pepper,_bell___healthy, and Grape___Black_rot, indicating both the plant species and the disease type.

Each image is tagged to reflect the plant's health status—whether it's afflicted by a particular disease such as Bacterial Spot, Early Blight, or Black Rot, or if it is disease-free. This labeling system supports the model in learning discriminative features for accurate classification.

To enhance the model's generalization capabilities and prevent overfitting, the dataset is commonly split into three subsets:

- Training set: Used to teach the model patterns and features associated with each class.
- Validation set: Helps in tuning model hyperparameters and monitoring performance during training.
- Test set: Used for final evaluation on unseen data to assess the model's real-world applicability.

This systematic organization and data splitting strategy contribute to building a robust and reliable model, capable of performing accurately under practical field conditions. [2].

1.3 Data Augmentation Techniques

Data augmentation is a widely used strategy in machine learning, particularly in deep learning, to enhance the volume and variety of training data without the need for manual labeling. By applying controlled transformations to existing samples, this technique helps mitigate issues such as overfitting and limited dataset size, leading to better model generalization and robustness.

In image-based applications like plant disease detection, common augmentation techniques include:

- Rotation: Randomly rotating images to simulate different orientations.
- Translation: Shifting the image along the X or Y axis.
- Flipping: Mirroring images horizontally or vertically.
- Zooming and Cropping: Scaling in or out and randomly cropping sections of the image to introduce spatial variance.
- Brightness and Contrast Adjustments: Simulating different lighting conditions.

These operations generate varied versions of the same image, allowing the model to learn invariant features and better adapt to real-world scenarios.

While image augmentation is most relevant for this project, it's worth noting that similar principles apply to other data types:

- Text data: Techniques like synonym replacement, random word insertion, and sentence paraphrasing help generate new textual inputs.
- Time-series data: Methods such as jittering (adding noise), scaling, or temporal shifting are used to introduce variability.

Through these approaches, data augmentation effectively increases the diversity of the training dataset, enabling models to capture more complex patterns and improving performance—particularly in cases where annotated data is scarce or imbalanced. [3]

1.4 Data preprocessing and feature extraction:

Data preprocessing is a critical step in any computer vision-based system, particularly in tasks like plant disease detection, where image quality directly affects model accuracy. Figure 2 illustrates the sequential steps applied to each image to enhance clarity and prepare it for feature extraction and classification.

To begin, the input RGB image is converted to grayscale to simplify processing and reduce computational load. A Gaussian filter is then applied to smooth the image and suppress noise, ensuring clearer feature boundaries.

Next, Otsu's thresholding is used to binarize the image by automatically selecting an optimal threshold value that separates the foreground (leaf) from the background. To enhance the binary image, morphological transformations—specifically closing operations—are applied to fill small holes within the detected foreground regions.

Once the foreground is clearly segmented, a bitwise AND operation is performed between the binarized image and the original RGB image. This isolates the leaf from the background, producing a segmented RGB image of the leaf.

Following segmentation, shape, texture, and color features are extracted:

- Shape Features:
  - Leaf area and perimeter are computed using contours, which trace the boundary points of uniform color or intensity.
- Color Features:
  - The mean and standard deviation of the R, G, and B channels are calculated to understand color distribution.
  - To measure greenness, the image is converted to HSV color space, and the green pixel ratio is determined. Pixels with Hue (H) values between 30 and 70 are considered green. The ratio of green pixels to total pixels in the channel provides a greenness metric.
  - The non-green area is then calculated by subtracting the green pixel ratio from 1.
- Texture Features:
  - Texture information is extracted from the grayscale image using methods such as gray-level co-occurrence matrix (GLCM) or other statistical descriptors (this part can be expanded based on your exact method).

These preprocessing steps collectively help to isolate the relevant parts of the image and extract meaningful features, which are crucial for accurate classification by the machine learning model. [4].

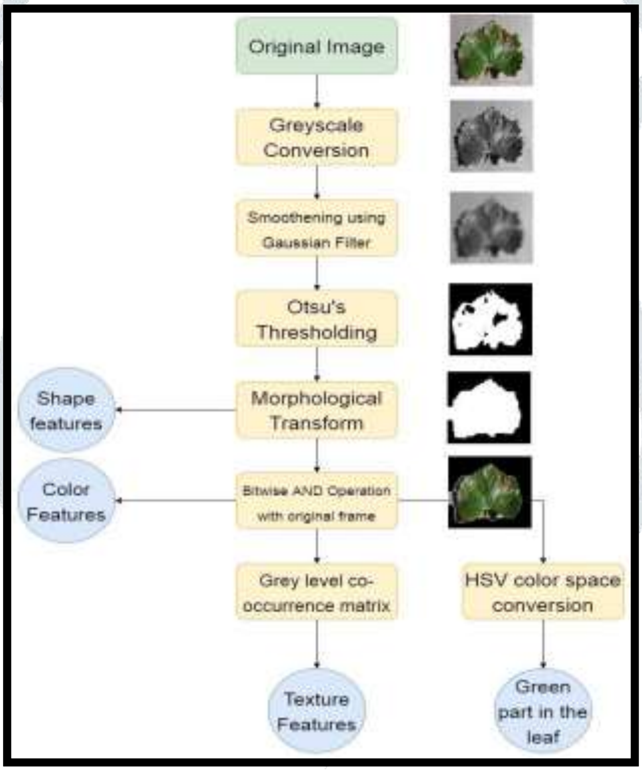


Figure 3. Steps for data preprocessing and feature extraction

1.5 Model Selection

CNN:

Convolutional Neural Networks (CNNs) are a specialized class of deep learning models designed for processing and analyzing visual data. Inspired by the organization of the visual cortex in the human brain, CNNs are highly effective in tasks involving image recognition, object detection, and image segmentation.

The core strength of CNNs lies in their ability to automatically and adaptively learn spatial hierarchies of features through convolutional layers. These layers apply filters (kernels) that slide over the input image to detect patterns such as edges, textures, and complex shapes. This enables the network to capture local and global features at various levels of abstraction.

CNN architectures typically include the following key components:

- Convolutional Layers: Perform feature extraction by applying kernels to the input.
- Activation Functions (e.g., ReLU): Introduce non-linearity into the model.

- Pooling Layers (e.g., Max Pooling): Downsample feature maps to reduce dimensionality and computation.
- Fully Connected Layers: Act as classifiers, mapping the extracted features to the output classes.

Due to their high accuracy and robustness, CNNs have become the foundation for many computer vision applications, including plant disease detection. Their ability to learn directly from raw image data—without the need for manual feature engineering—makes them particularly suitable for real-world agricultural use cases where variations in lighting, background, and leaf orientation are common. [5]

1.6 Sequential model

The Sequential class in the Keras deep learning library serves as a fundamental tool for building and training neural networks. Designed with simplicity in mind, this class allows developers to construct models by stacking layers sequentially—making it especially suitable for feedforward neural networks and convolutional neural networks (CNNs) where data flows in a straight path from input to output.

With its user-friendly syntax, the Sequential model is ideal for both beginners and experienced practitioners. It enables rapid prototyping by allowing layers to be added one at a time using the .add() method. Each layer takes the output of the previous one as its input, resulting in a linear architecture that is easy to understand and implement.

The Sequential model is best used when the architecture does not require multiple inputs or outputs, shared layers, or non-linear connectivity between layers. For more complex designs, Keras offers the Functional API, but for standard image classification tasks like plant disease detection, the Sequential model often provides all the necessary functionality with a streamlined workflow.

In summary, Keras's Sequential class is a convenient and efficient way to develop deep learning models, offering both flexibility and ease of use for a wide range of applications in computer vision and beyond. [6]

1.7 Training

Training a neural network involves optimizing the parameters—kernels in convolutional layers and weights in fully connected layers—so that the model can accurately map input data to corresponding output labels. This process minimizes the difference between the model's predicted outputs and the actual ground truth labels using a loss function and backpropagation.

In this study, 87% of the dataset was allocated for training purposes. During this phase, the Convolutional Neural Network (CNN) learns to extract meaningful features from plant leaf images, such as texture, color, shape, and disease-specific patterns. These features are used to build internal representations that enable the network to distinguish between healthy and diseased leaves, as well as among various disease classes.

The training process is iterative, involving multiple epochs in which the model adjusts its internal parameters to reduce the loss. Optimization techniques such as Stochastic Gradient Descent (SGD) or Adam are employed to fine-tune the model. Additionally, techniques like data augmentation, dropout, and early stopping may be integrated to enhance the model's generalization and prevent overfitting.

The goal of training is to enable the model to learn robust and discriminative features that can later be used to make accurate predictions on unseen plant leaf images during testing.

1.8 Testing

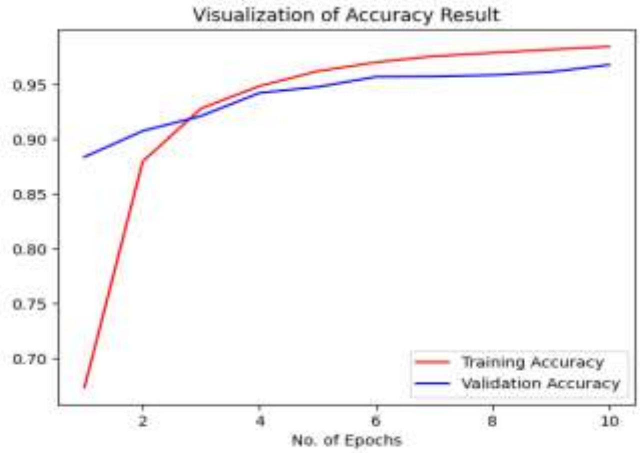
The testing phase serves as a critical step in evaluating the generalization capability of the trained model. Unlike training and validation data, which are used during the learning process, the test dataset provides an unbiased assessment of the final model's performance on unseen data.

In this study, 13% of the dataset was reserved exclusively for testing. This data was not exposed to the model during the training phase, ensuring that the evaluation reflects the model's ability to recognize plant leaf diseases in real-world scenarios.

During testing, the trained Convolutional Neural Network (CNN) processes each test image and applies the learned features—extracted during training—to classify the image as healthy or diseased, and further identify the specific disease type if applicable. The effectiveness of the model is measured using standard performance metrics such as accuracy, precision, recall, F1-score, and confusion matrix analysis.

This stage verifies how well the model has learned to generalize from the training data and determines its readiness for deployment in real agricultural environments, where accurate and timely plant disease identification is essential.

IV. RESULT AND DISCUSSION



Accuracy

Figure 4. Training and Validation Accuracy

Table 1: Accuracy Results Over Epochs

Epoch	Training Accuracy	Validation Accuracy	Observation
1	0.68	0.88	Initial low training, decent validation
2	0.88	0.91	Big jump in training accuracy
3	0.92	0.925	Training catches up with validation
4	0.95	0.94	Both continue to improve
5	0.96	0.945	Slight increase, stable pattern
6	0.97	0.955	Small gap starts to emerge
7	0.975	0.957	Gap slightly increases
8	0.978	0.958	Model close to saturation
9	0.979	0.96	Very minimal improvement
10	0.981	0.965	Near convergence

The line graph titled "Visualization of Accuracy Result" illustrates the performance of the model over 10 training epochs, comparing both training accuracy (represented by the red line) and validation accuracy (depicted by the blue line).

From the graph, it is observed that the training accuracy steadily increases from approximately 0.68 in the first epoch to nearly 0.98 by the tenth epoch, indicating that the model is effectively learning the patterns present in the training data. In parallel, the validation accuracy also demonstrates a consistent upward trend, starting around 0.88 and reaching approximately 0.965 by the final epoch. This positive trend suggests that the model maintains strong generalization capabilities when applied to previously unseen data.

However, a slight divergence between the training and validation accuracy curves begins to emerge after epoch 4, which may hint at the early stages of overfitting—where the model starts to memorize the training data rather than learning to generalize. Despite this, the gap remains narrow, which implies that the model continues to perform well on the validation set and does not exhibit significant overfitting.

In conclusion, the graph demonstrates that the training process is effective and that the model achieves high accuracy while maintaining strong generalization, making it suitable for real-world deployment.

Loss

The loss curve, as depicted in the first image, presents the progression of both training loss (loss) and validation loss (val_loss) across 10 epochs. At the start of the training process, the model exhibits a high loss value, which reflects a poor initial fit to the data. However, with continued training, both training and validation losses show a significant downward trend, indicating effective learning and improvement in model performance.

Throughout the training, the validation loss closely follows the training loss, suggesting that the model is generalizing well to unseen data. By the 10th epoch, both losses stabilize at values approaching zero, which is a strong indication of successful convergence and minimal overfitting.

Although there are minor fluctuations in the validation loss, these can be attributed to natural variance in the validation data and are not indicative of serious generalization issues. The convergence of the two loss curves supports the conclusion that the model is well-trained and capable of accurately learning from the input data without overfitting.

In summary, the loss curve effectively demonstrates that the model achieves both efficient learning and strong generalization, making it suitable for practical applications in plant disease detection.

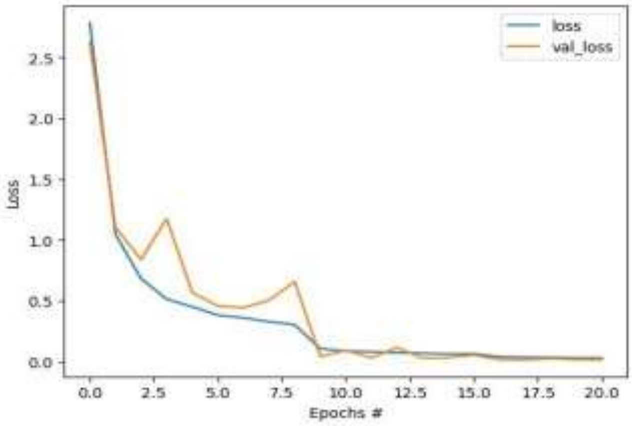


Figure 5. Training and Validation Loss

Table 2: loss Results Over Epochs

Epoch Range	Training Loss	Validation Loss	Observations
0 - 2	Very high (Above 2.5)	Very high (Above 2.5)	The model starts with a poor fit and high loss.
2 - 5	Rapid decrease	Rapid decrease with fluctuations	The model learns quickly, but validation loss fluctuates slightly.
5 - 10	Gradual decrease	Slight fluctuations	Training loss continues to decline, and validation loss stabilizes.
10 - 20	Gradual decrease	Near zero (~0.1)	Loss stabilizes, indicating good model convergence with minimal overfitting.

Validation and Testing

To ensure that a deep learning model for plant leaf disease detection performs reliably and accurately, it is essential to implement effective validation and testing strategies. These processes not only evaluate model performance but also help confirm its ability to generalize to new, unseen data.

A widely adopted method is k-fold cross-validation, where the dataset is divided into k equal-sized subsets or "folds." In each iteration, the model is trained on k–1 folds and validated on the remaining fold. This process is repeated k times, with each fold serving as the validation set once. The final performance metrics are averaged over all k iterations, providing a more robust and generalized assessment of the model's performance. This technique helps minimize the risk of overfitting to a particular subset of data.

In addition to cross-validation, another commonly used strategy involves dividing the dataset into three distinct subsets: training, validation, and testing sets. The validation set is used to monitor the model's performance during training and to fine-tune hyperparameters, such as learning rate, number of layers, and batch size. This ensures the model remains adaptable and avoids overfitting before final evaluation.

Once the optimal model configuration is identified through validation, it is evaluated using a dedicated test set. The test dataset provides an unbiased measure of the model's real-world performance, helping assess its effectiveness across different environmental conditions, plant species, and disease types.

In summary, the combination of cross-validation, hyperparameter tuning, and final testing ensures that the model is both accurate and generalizable, making it suitable for practical agricultural applications.

Confusion Matrix

To ensure that a deep learning model for plant leaf disease detection performs reliably and accurately, it is essential to implement effective validation and testing strategies. These processes not only evaluate model performance but also help confirm its ability to generalize to new, unseen data.

A widely adopted method is k-fold cross-validation, where the dataset is divided into k equal-sized subsets or "folds." In each iteration, the model is trained on k–1 folds and validated on the remaining fold. This process is repeated k times, with each fold serving as the validation set once. The final performance metrics are averaged over all k iterations, providing a more robust and generalized assessment of the model's performance. This technique helps minimize the risk of overfitting to a particular subset of data.

In addition to cross-validation, another commonly used strategy involves dividing the dataset into three distinct subsets: training, validation, and testing sets. The validation set is used to monitor the model's performance during training and to fine-tune hyperparameters, such as learning rate, number of layers, and batch size. This ensures the model remains adaptable and avoids overfitting before final evaluation.

Once the optimal model configuration is identified through validation, it is evaluated using a dedicated test set. The test dataset provides an unbiased measure of the model's real-world performance, helping assess its effectiveness across different environmental conditions, plant species, and disease types.

In summary, the combination of cross-validation, hyperparameter tuning, and final testing ensures that the model is both accurate and generalizable, making it suitable for practical agricultural applications.

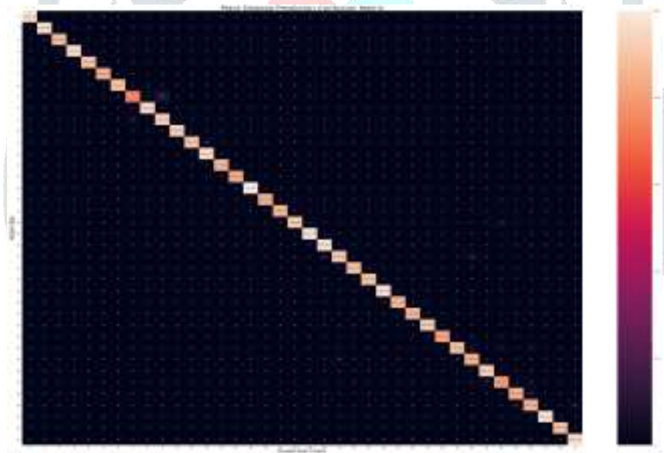


Figure 6. Confusion Matrix

V. Conclusion

The application of machine learning and OpenCV for plant disease detection presents a powerful and efficient solution to early disease identification in crops. By combining advanced image processing techniques with machine learning algorithms, this system significantly enhances diagnostic accuracy while minimizing human effort. The ability to detect plant diseases at an early stage allows for timely intervention, which can ultimately lead to improved crop yields and enhanced food security.

The integration of deep learning models, particularly Convolutional Neural Networks (CNNs), further strengthens the system's capability to classify and detect diseases with high precision. These models excel at automatically extracting features from plant leaf images, reducing the need for manual feature engineering and enhancing the overall performance of the disease detection system.

However, there are still challenges that need to be addressed. Issues related to dataset quality, such as the need for larger, more diverse datasets to account for varying environmental conditions and plant species, remain a key consideration. Additionally, computational requirements for training and deploying deep learning models can be substantial, posing challenges for real-time implementation, especially in resource-constrained settings.

Despite these challenges, ongoing advancements in artificial intelligence (AI) and computer vision are continuously improving the effectiveness and efficiency of plant disease detection systems. Future

enhancements, such as real-time disease detection, mobile-based applications, and the incorporation of more diverse and comprehensive datasets, hold the potential to further improve accessibility and reliability, making this technology an invaluable tool in modern agriculture.
In conclusion, the fusion of machine learning, image processing, and deep learning holds great promise in revolutionizing plant disease management, offering farmers a timely and accurate method for disease detection and intervention.

VI. REFERENCE

[1]Saeed, A., Abdel-Aziz, A. A., Mossad, A., Abdelhamid, M. A., Alkhaled, A. Y., & Mayhoub, M. (2023). Smart detection of tomato leaf diseases using transfer learning-based convolutional neural networks. Agriculture, 13(1), 139.

[2]Ahmad, A., Saraswat, D., & El Gamal, A. (2023). A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools. Smart Agricultural Technology, 3, 100083.

[3] Ullah, Z., Alsubaie, N., Jamjoom, M., Alajmani, S. H., & Saleem, F. (2023). EffiMob-Net: A deep learning-based hybrid model for detection and identification of tomato diseases using leaf images. Agriculture, 13(3), 737.

[4]Moupojou, E., Tagne, A., Retraint, F., Tadonkemwa, A., Wilfried, D., Tapamo, H., & Nkenlifack, M. (2023). FieldPlant: A dataset of field plant images for plant disease detection and classification with deep learning. IEEE Access, 11, 35398-35410.

[5]Roy, K., Chaudhuri, S. S., Frnda, J., Bandopadhyay, S., Ray, I. J., Banerjee, S., & Nedoma, J. (2023). Detection of tomato leaf diseases for agro-based industries using novel PCA DeepNet. IEEE Access, 11, 14983-15001.

[6]Shoaib, M., Shah, B., Ei-Sappagh, S., Ali, A., Ullah, A., Alenezi, F., ... & Ali, F. (2023). An advanced deep learning models-based plant disease detection: A review of recent research. Frontiers in Plant Science, 14, 1158933.

[7]Chin, R., Catal, C., & Kassahun, A. (2023). Plant disease detection using drones in precision agriculture. Precision Agriculture, 24(5), 1663-1682.

[8]Shi, T., Liu, Y., Zheng, X., Hu, K., Huang, H., Liu, H., & Huang, H. (2023). Recent advances in plant disease severity assessment using convolutional neural networks. Scientific Reports, 13(1), 2336.

[9]Aggarwal, M., Khullar, V., Goyal, N., Alammari, A., Albahar, M. A., & Singh, A. (2023). Lightweight federated learning for rice leaf disease classification using non independent and identically distributed images. Sustainability, 15(16), 12149..

VII. LIST OF FIGURE

VIII. LIST OF TABLE