**Data Science Technology for OpenCV**

 **Apurva Bhavsar** **¹, Gautam Mehetre** **², Vedashree Mainkar³, Atharva Godse⁴, Nikhil Jadhav⁵**

1Assistant Professor, Department of Artificial Intelligence & Data Science, Guru Gobind Singh College of Engineering & Research Center, Nashik, Maharashtra, India

**²**Student, Department of Artificial Intelligence & Data Science, Guru Gobind Singh College of Engineering & Research Center, Nashik, Maharashtra, India

**³**Student, Department of Artificial Intelligence & Data Science, Guru Gobind Singh College of Engineering & Research Center, Nashik, Maharashtra, India

**⁴**Student, Department of Artificial Intelligence & Data Science, Guru Gobind Singh College of Engineering & Research Center, Nashik, Maharashtra, India

**⁵**Student, Department of Artificial Intelligence & Data Science, Guru Gobind Singh College of Engineering & Research Center, Nashik, Maharashtra, India

**ABSTRACT**

Timely detection of plant diseases is essential for maintaining crop health and ensuring food security. This study presents a Plant Disease Recognition System using Deep Learning, specifically a Convolutional Neural Network (CNN), to identify plant diseases from leaf images. The system utilizes a publicly available Plant Diseases Dataset consisting of RGB images from 38 classes of healthy and diseased leaves. Data preprocessing techniques, including resizing, normalization, and data augmentation, were applied to improve model performance and generalization The CNN model was trained using the Adam optimizer and categorical cross-entropy loss, achieving a training accuracy of 98.1% and a validation accuracy of 95.4%. Performance was further evaluated using precision, recall, F1-score, and a confusion matrix. The model was deployed via Streamlit, providing a user-friendly web interface where users can upload images, receive disease predictions, and access recommended treatments.The proposed system offers an efficient, accessible, and scalable solution for early plant disease detection, supporting farmers and agricultural experts in timely decision-making. Future enhancements may include dataset expansion, camera integration, and multilingual support to improve usability in diverse agricultural environments.

**Keywords**: Plant Disease Detection, CNN, Deep Learning, Streamlit, Image Classification, Agriculture

**1 INTRODUCTION**

Agriculture plays a vital role in sustaining economies and food systems worldwide. One of the major challenges faced by the agricultural sector is the timely identification and management of plant diseases, which, if left undetected, can lead to significant losses in crop yield and quality. Conventional disease detection methods rely on manual inspection by agricultural experts, which can be time-consuming, labor-intensive, and subject to human error, especially in large-scale farming environments.

With the advancement of Artificial Intelligence (AI) and Deep Learning (DL) techniques, image-based disease recognition systems have gained popularity for their ability to automatically analyze visual patterns and classify plant health conditions with high accuracy. Among these, Convolutional Neural Networks (CNNs) have demonstrated remarkable success in image classification and feature extraction tasks due to their capacity to learn complex hierarchical features from raw image data.

This research proposes a Plant Disease Recognition System utilizing CNN models for early and accurate identification of plant diseases from leaf images. The system is trained on a publicly available plant disease dataset and deployed using Streamlit, a Python-based web application framework. The developed system offers a user-friendly interface, making it accessible to both farmers and agricultural practitioners for real-time disease detection and management recommendations.

**2 METHODOLOGY**

In this study, Flower Image Classification is carried out using Deep Learning techniques in the form of Convolutional Neural Networks (CNNs) combined with Transfer Learning. CNNs have proven highly effective in image classification tasks due to their ability to automatically extract complex visual features from images. To enhance the performance and training efficiency, a pre-trained MobileNetV2 model was integrated as the feature extractor, followed by custom classification layers. This method significantly improves the accuracy of flower classification, providing a lightweight and reliable solution suitable for real-time applications via a web-based interface.

**2.1 Dataset Description**

For this research, the publicly available TensorFlow Flowers Dataset was used, which is hosted on TensorFlow Datasets. This dataset contains a comprehensive collection of high-quality RGB images from five distinct flower species: Daisy, Dandelion, Roses, Sunflowers, and Tulips. The dataset was divided into two parts: 80% for training and 20% for testing, ensuring effective generalization and evaluation of the model’s performance. The images were provided in various resolutions, which required preprocessing before being fed into the CNN model.

**Figure 1: Samples of Dataset**

**2.2 Data Preprocessing**

To enhance model accuracy and ensure efficient learning, essential data preprocessing steps were performed on the flower images before training the CNN model:

**1. Resizing & Normalization:**

* All images were resized to 224 × 224 pixels to maintain uniform input dimensions for the MobileNetV2 architecture.
* Pixel values were normalized to the range [0, 1] to stabilize and speed up the training process.

**2. Data Augmentation:**

* To increase dataset variability and prevent overfitting, several augmentation techniques were applied, including:
	+ Horizontal flipping
	+ Random rotations (up to 20°)
	+ Zooming (up to 20%)
	+ Brightness adjustments

**3. Dataset Splitting:**

* The dataset was divided into Training (80%) and Testing (20%) sets. The training set was shuffled and batched, while the test set was used solely for final model evaluation to assess accuracy and generalization.

**2.3 Model Compilation & Training**

The CNN model was built by integrating the MobileNetV2 pre-trained on the ImageNet dataset as the base model, excluding its top layers. It was followed by a Global Average Pooling layer, two Dense layers, and Dropout layers for regularization. The MobileNetV2 base model was frozen to retain its pre-learned features, while the custom layers were trained on the flower dataset.

The model was compiled using the Adam optimizer with a learning rate of 0.0001, which dynamically adapts the learning rate during training to ensure faster convergence. The Sparse Categorical Cross-Entropy loss function was used, as the problem involves multi-class classification.

A batch size of 32 was chosen to balance between memory efficiency and computational speed. The model was trained for 20 epochs, with early stopping implemented to halt training if no improvement was observed in validation loss, preventing unnecessary computation and overfitting.

**3 MODELING AND ANALYSIS**

**Figure 2: Model Architecture**

This section outlines the architectural design, training procedure, evaluation metrics, and performance analysis of the proposed deep learning model for plant species classification using flower images.

**3.1 Model Architecture**

The model architecture is built upon a pre-trained MobileNetV2 as a feature extractor, leveraging the power of transfer learning for effective image classification. The pipeline starts with image augmentation layers to increase the diversity of training data and improve generalization.

The core stages of the model are:

* **Input Layer**: Receives RGB flower images resized to 224×224 pixels.
* **Data Augmentation**: Includes random horizontal flipping and rotation for enhancing model robustness.
* **Feature Extraction Layer**: Utilizes MobileNetV2 pretrained on the ImageNet dataset with frozen base layers to retain previously learned features.
* **Global Average Pooling**: Reduces spatial dimensions by computing the average of each feature map.
* **Fully Connected Dense Layers**: Includes a 128-unit Dense layer with ReLU activation, followed by a Dropout layer for regularization.
* **Output Layer**: Consists of a Dense layer with Softmax activation to classify images into five classes: *Daisy, Dandelion, Roses, Sunflowers, and Tulips*.

This architecture allows efficient learning while mitigating overfitting and reducing training time.

**3.2 Training & Testing**

**3.2.1 Training Parameters**

The model was compiled and trained using the following hyperparameters:

* **Batch Size**: 32
* **Epochs**: 20
* **Optimizer**: Adam (learning rate = 0.0001)
* **Loss Function**: Sparse Categorical Cross-Entropy
* **Activation Functions**: ReLU in Dense layers and Softmax for final classification

**3.2.2 Training and Testing Process**

* **Feature Learning**:
The MobileNetV2 model extracts high-level spatial and semantic features from flower images.
* **Backpropagation and Optimization**:
Loss was computed using sparse categorical cross-entropy and minimized via the Adam optimizer. Weights were iteratively updated using backpropagation.
* **Validation and Early Stopping**:
A validation set monitored model generalization during training. EarlyStopping callback was employed to halt training when the validation loss plateaued, preventing overfitting.
* **Testing**:
Final model performance was evaluated on an unseen test set. Metrics such as accuracy, precision, recall, F1-score, and a confusion matrix were used to quantify predictive ability.

**3.3 Performance Evaluation Metrics**

To assess the model's effectiveness, the following metrics were computed:

**1.Accuracy (ACC):**

Proportion of correctly classified images out of total predictions.

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

**2.Precision:**
Determines how many positively classified images are actually correct:

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

**3.Recall:**
Measures the model’s ability to correctly detect positive (target) classes:

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

**4.Confusion Matrix Components**

* **TP (True Positive)**: Correctly predicted positive cases
* **TN (True Negative)**: Correctly predicted negative cases
* **FP (False Positive)**: Incorrectly predicted positive cases
* **FN (False Negative)**: Incorrectly predicted negative cases

**4 RESULTS AND DISCUSSION**

**4.1 Results**

This is the home page of the application. It includes:

* A title: **"PLANT RECOGNITION SYSTEM"**.
* A representative image of a diseased plant.
* A welcome message with a short description of the app’s purpose.
* A **"How It Works"** section with the following steps:
	1. Upload Image
	2. Analysis by the system
	3. Display of results and recommendations

This screen sets the context and provides clear instructions for users when they visit the app.

**Figure 3: User Interface – Input**

* A results section displaying:
	+ The **model prediction** (e.g., *Cherry (including sour) - Powdery Mildew*).
	+ A **recommended treatment** (e.g., *Neem oil spray*).
	+ **Dosage and application instructions** for the recommended medication.

**4.2 Discussion**

This research presents a Plant Disease Recognition System utilizing Convolutional Neural Networks (CNNs) for identifying diseases from leaf images. The CNN model effectively extracts visual features through multiple convolutional, activation (ReLU), and pooling layers, and classifies the images in the output layer. The system accurately predicted plant diseases such as Powdery Mildew in cherry leaves, and provided relevant treatment recommendations like neem oil spray, complete with dosage and application guidelines.

The model was successfully deployed using Streamlit, providing a lightweight, interactive, and user-friendly web-based interface. Users can easily upload plant images, initiate predictions, and view results, making the application accessible for both technical and non-technical users. The clear, intuitive layout and the responsiveness of the dashboard improve usability, while Streamlit’s open-source, lightweight nature allows for easy deployment both locally and on the cloud.

Despite the promising results, there remains scope for improvement. Expanding the dataset to include more plant species and a wider range of diseases would enhance the model’s accuracy and generalization capability. Additionally, integrating features such as camera-based image capture, offline accessibility, and multilingual support could further improve the system’s practical usability in real-world agricultural settings.

Overall, this system demonstrates a practical, effective solution for early plant disease detection, encouraging timely intervention, improved crop health management, and ultimately supporting sustainable agriculture practices.

**5 CONCLUSION**

In this research, a Plant Disease Detection System was developed using Convolutional Neural Networks (CNNs) to identify plant diseases through leaf images. The model was trained on a publicly available dataset consisting of 38 different plant disease classes and healthy plant images. By applying multiple convolutional, activation, and pooling layers, the CNN effectively extracted critical visual features and achieved high classification accuracy.

The system demonstrated excellent performance in predicting diseases such as *Powdery Mildew* in *cherry leaves*, and suggested suitable treatments along with dosage and application instructions. With an intuitive and interactive web interface built using Streamlit, the application ensures ease of use for both technical and non-technical users, enabling quick and accurate plant disease diagnosis.

The research highlights the potential of deep learning-based solutions in the agricultural sector by promoting early disease detection, timely treatment, and improved crop management. This contributes to reducing crop losses, increasing yield, and enhancing sustainable farming practices.

Future improvements to the system could include:

* Expanding the dataset with more plant species and disease categories.
* Integrating real-time image capture using mobile or embedded devices.
* Providing offline access and multilingual support to benefit farmers in remote regions.

Overall, this project establishes a practical, efficient, and scalable framework for plant disease detection, offering valuable support to farmers, agronomists, and agricultural researchers worldwide.

**6 REFERENCES**

[1] Mohanty, S.P., Hughes, D.P., Salathé, M., "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science*, 2016, 7(1419).

[2] Ferentinos, K.P., "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, 2018, 145.

[3] Kingma, D.P., Ba, J., "Adam: A Method for Stochastic Optimization," *International Conference on Learning Representations (ICLR)*, 2014.

[4] Chollet, F., "Deep Learning with Python," *Manning Publications*, 2017.

[5] Brownlee, J., "Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python," *Machine Learning Mastery*, 2019.

[6] Goodfellow, I., Bengio, Y., Courville, A., "Deep Learning," *MIT Press*, 2016.

[7] Aggarwal, C.C., "Neural Networks and Deep Learning: A Textbook," *Springer*, 2018.

[8] Zhang, S., Wu, X., You, Z., "Leaf image based cucumber disease recognition using sparse representation classification," *Computers and Electronics in Agriculture*, 2017, 134.

[9] Too, E.C., Yujian, L., Njuki, S., Yingchun, L., "A comparative study of fine-tuning deep learning models for plant disease identification," *Computers and Electronics in Agriculture*, 2019, 161.

[10] Brahimi, M., Arsenovic, M., Laraba, S., Sladojevic, S., Boukhalfa, K., Moussaoui, A., "Deep learning for plant diseases: detection and saliency map visualisation," *Human and Machine Learning*, 2018.

[11] Kaggle, "New Plant Diseases Dataset," *Kaggle Datasets*, 2021.