Plant Disease Classification using Advanced Machine Learning Techniques

Aryan Shetty
Department of Computer Science
University of Illinois Chicago
Chicago, USA
ashet31@uic.edu

Niket Pathak
Department of Computer Science
University of Illinois Chicago
Chicago, USA
npath5@uic.edu

Om Sali
Department of Computer Science
University of Illinois Chicago
Chicago, USA
osali4@uic.edu

Soureesh Dalal
Department of Computer Science
University of Illinois Chicago
Chicago, USA
sdala@uic.edu

Varun Phanindra Shrivathsa
Department of Computer Science
University of Illinois Chicago
Chicago, USA
vphan@uic.edu

I. Introduction

Effective plant disease management is crucial for reducing crop loss and maintaining food security. Most of the traditional techniques for the detection of diseases are time-consuming and often restricted by the expertise a diagnostician needs to conduct an effective diagnosis. This paper presents a machine learning-based model for plant illness classification using images of damaged and healthy leaves from different crops like tomatoes, potatoes, and grapes. Using a dataset of 38 distinct disease classes, this work will explore several architectures: a Custom CNN, ResNet-50, VGG, SVM, and Random Forest all with the aim of choosing the best accurate and efficient one. The methodology, through preprocessing approaches like data augmentation and transfer learning, tries to improve model robustness and generalization. It thus opens up a realistic path toward a viable and scalable solution for plant disease prediction.

II. Problem Statement

Plant diseases pose a serious threat to agricultural productivity and incur significant economic losses while creating negative impacts on food security. Traditional plant disease detection depends mostly on manual observation, involving expert knowledge that is time-consuming and may be inconsistent.

This project evaluates and compares several machine learning models for plant disease classification. We consider a set of approaches using Random Forest, Support Vector Machines (SVM), CNNs with an architecture developed in this research, and also pre-trained deep learning architectures like VGG and ResNet. The presented paper examines each model's performances concerning the criteria of accuracy, robustness, and computational efficiency in a comparative manner.

The project is aimed at finding the best machine learning approach for the early detection of plant diseases through intensive comparison and evaluation. The different models are applied in the research for an understanding of the strengths and weaknesses inherent in each algorithm in determining the best model for practical applications in agriculture. Contributions to the knowledge of how machine learning techniques can be applied to further improve disease detection and reduce crop loss are therefore expected from the findings of the project.

III. Objective

In this project, the we have proposed a high-accuracy system for identifying plant diseases using Convolutional Neural Networks, Support Vector Machines, Random Forest, VGG16 , and Res Net models. This project will overcome the disadvantages of the traditional detection methods, which were often manual, time-consuming, and expert-dependent, considering the high impact that plant diseases have on agricultural productivity, by allowing for quick detection using ML/DL. The project initiated the application of an open plant disease dataset, pre-processing a wide variety of images of plants, including both healthy and sick ones, to construct a quality input pipeline that raised the model's training and prediction accuracies. To determine which models were most appropriate for scalable illness prediction, a comparative analysis of each model was carried out to evaluate its performance in terms of accuracy, efficiency, precision, recall, and F1-Score . By offering early disease diagnosis, this technology ultimately sought to improve precision agriculture by preserving crop health, lowering yield losses, and encouraging sustainable farming methods.

IV. Backgrounds Study

1) Plant disease detection has been paid with the use of machine learning and deep learning techniques. Datasets like PlantVillage, PlantCLEF, and AGRONOMI-Net

provide RGB, infrared, and thermal images for the accurate classification of plant diseases. High accuracy in disease identification can be achieved by CNNs through feature extraction for the purpose of disease identification. Real-time models such as YOLO and SSD have been effective in localization. Techniques such as Mask R-CNN improve lesion segmentation up to 97.24% in accuracy, and Random Forest (RF) reduces overfitting through ensemble learning. Transfer learning and data augmentation are imperative for generalizability, especially in the case of small or imbalanced datasets.[1]

2) Deep learning methods, particularly CNNs, have automated feature extraction, surpassing traditional approaches in accuracy. However, challenges like data scarcity in agriculture remain. Data augmentation and Generative Adversarial Networks (GANs) are being utilized to increase data diversity. Emerging techniques, such as few-shot learning, enhance accuracy with limited data, while hyperspectral imaging offers potential for early disease detection. Further research is needed to improve model robustness, interpretability, and adaptability to different environments. [2]

3) Machine learning techniques like SVM, RF, ANN, and CNNs have strong potential for the detection of diseases and pests in crops. These techniques are classified into data acquisition, preprocessing, and classification. Although ML helps in precision agriculture, issues related to handling variability in field data and computational demands pose a challenge. Improvement in model scalability, robustness, and applicability is under research, with the integration of IoT and data augmentation to handle such challenges. [3]

4) Artificial intelligence has revolutionized the prediction and diagnosis of plant diseases in order to improve crop quality and yield. AI-based automated detection methods, mostly ML and DL models, compensate for the drawbacks of the traditional methods, which are often time-consuming and laborious. These innovations enable faster interventions with a view to reducing productivity losses in agriculture. [4]

5) In improving the accuracy and efficiency of plant disease detection, classification algorithms such as SVM, neural networks, and fuzzy logic play a very important role. For example, BPNNs use color and texture cues to classify diseases. Moreover, Random Forest classifiers, when combined with feature descriptors such as Haralick texture and Hu moments, outperform alternatives like Naïve Bayes. These methods underline the importance of tailored feature extraction strategies in ML-based plant disease detection. [5]

6) CNNs have also been successfully applied in the classification of plant diseases in crops such as apple and tomato with up to 88.7% accuracy using optimization techniques such as dropout layers. These models are reliable for disease diagnosis, yet their large model size inhibits them from being applied in embedded applications. The future in this aspect would be looking at alternative architectures and more extensive datasets for better efficiency and accuracy. [6]

V. Data

The dataset of plant leaf images used in this work is openly accessible and was primarily produced for disease detection and classification purposes. About 87,000 high-resolution RGB images of crop leaves make up the dataset; each image has been appropriately labeled and categorized into 38 groups.

The dataset, which is well-known in the domains of machine learning and plant pathology, was acquired from Kaggle. To provide clarity and consistency across the collection, each photograph is tagged with the crop species and health status. It is more suited for thorough model training and assessment since it incorporates a variety of crop types and diseases.

The dataset's breadth and structure allow it to cover a broad range of plant conditions, which makes it ideal for training machine learning models to precisely detect and categorize diseases. The precise classification and superb annotations aid in the development of models that are able to discern minute visual distinctions between healthy and unhealthy plants. This feature is especially important in real-world agricultural applications where accurate disease identification can have a big impact on crop productivity and management strategies.

The dataset, which covers a broad spectrum of crop species and disease types, offers a rich backdrop for research on plant health management and precision agriculture. Because of this variability, models can be applied to a range of agricultural contexts and generalized.

A. Data Cleaning and Handling Missing Values

The following techniques were applied to clean the data and handle any missing values:

1) Image Resizing and Normalization: All images were resized according to the model to ensure consistent input dimensions for the machine learning models. This standardization guarantees uniform processing of all images. Additionally, each pixel value was rescaled to a range of [0, 1] by dividing by 255, which normalizes the data and speeds up model convergence, ultimately enhancing performance.

2) Error Handling During Image Loading: During the image loading process, any images that could not be loaded due to corruption or other issues were skipped, ensuring that only valid images were included in the dataset.

3) Bootstrap Sampling: To address dataset imbalance and enhance model generalization, bootstrap sampling was employed. This technique involves randomly resampling the dataset with replacement to generate a new dataset of the same size, which helps ensure resilience during the training process.

4) Outlier Identification and Management: Outliers were implicitly managed during the preprocessing stage. Images with anomalies such as incorrect tagging or corruption were excluded from the dataset, ensuring the integrity of the data used for model training.

5) Dataset Augmentation: Various augmentation techniques, such as horizontal flipping, shearing, and zooming, were applied to generate synthetic variants of the original images. This increased the diversity of the dataset, reduced the impact of outliers, and helped the model become more robust by learning from varied examples.

B. Feature Engineering

1) Histogram of Oriented Gradients (HOG): To extract texture features, Histogram of Oriented Gradients (HOG) was used. This technique captures edge orientations and gradients, which are particularly useful for detecting disease-specific patterns, such as leaf spots or discoloration.

2) Label Encoding: Crop and disease labels were encoded into numerical categories to ensure compatibility with machine learning algorithms. This encoding process allows the algorithms to effectively handle and classify the labels during training.

3) Augmentation: Techniques like zooming, shearing, and horizontal flipping were applied as part of the dataset augmentation process to improve the model's robustness and ability to generalize to new, unseen data.

Attributes	No. of images
Tomato___Late_blight	1851
Tomato___healthy	1926
Grape___healthy	1692
Orange___Haunglongbing_(Citrus_greening	2010
Soybean___healthy	2022
Squash___Powdery_mildew	1736
Potato___healthy	1824
Corn_(maize)___Northern_Leaf_Blight	1908
Tomato___Early_blight	1920
Tomato___Septoria_leaf_spot	1745
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot	1642
Strawberry___Leaf_scorch	1774
Peach___healthy	1728
Apple___Apple_scab	2016
Tomato___Tomato_Yellow_Leaf_Curl_Virus	1961

Fig. 1. Description of Attributes

VI. Experimental Design

A. Data Partitioning

The dataset was split into training (80%) and validation (20%) sets using stratified sampling to ensure balanced class representation.

Fig. 2. tomato_late_blight

B. Model Selection

To find the best method for classifying plant diseases, a number of machine learning models were trained and assessed, including Support Vector Machines (SVM), Random Forest (RF), Custom Convolutional Neural Networks (CNN), VGG16, and ResNet. Every model was designed to manage categorical labels and picture data.

C. Evaluation Metrics

To offer a thorough study of each model's classification capacity, conventional measures such as accuracy, precision, recall, and F1-Score were used to evaluate model performance.

D. Cross-Validation

For traditional models like RF, K-fold cross-validation ($k = 5$) was used to reduce overfitting and guarantee accurate performance estimations.

E. Hyperparameter Tuning

Hyperparameters for each model, such as the number of estimators for RF and the kernel type for SVM, were optimized using grid search and randomized search approaches. Learning rate, dropout rates, and the number of convolutional layers were among the architecture-specific parameters that were adjusted for CNNs.

F. Augmentation Pipeline

To increase the resilience of CNN models, data augmentation was incorporated into the training process by adding variables such as flips and rotations.

G. Comparative Analysis

To identify the most accurate and effective model, the final outcomes from each model were compared. The capacity of models to generalize across various plant species and disease types was given particular attention.

VII. Methodology

In this work, a supervised learning strategy has been applied to the challenge of plant disease classification. This approach was followed due to the availability of labeled datasets, something very necessary in training the model to identify healthy and diseased plants through their appearance. Supervised learning helped the model learn all those minute patterns

associated with any particular disease for proper classification of new, unseen data.

A. Custom CNN Model

In the custom CNN, a supervised learning framework has been used that uses labeled image data of plant leaves categorized by their respective disease classes. This was one of the ideal approach for the problem at hand, as it make sure that the model learns features tied to various diseases and classifies unseen data with high accuracy.

1) Model Design: The architecture of the custom CNN was carefully constructed to balance feature extraction and computational efficiency. The model includes the following layers, and the input images were resized to a target size of 150×150 pixels to standardize the input dimensions:

a) Convolutional Layers: Five convolutional layers were used, with filter sizes progressively increasing (32, 64, 128, 256, 512). Each layer uses a kernel size of 3×3 and ReLU activation to capture spatial patterns in the input images.

b) Pooling Layers: MaxPooling layers were inserted after each convolutional layer to reduce spatial dimensions of the image while retaining key features.

c) Global Average Pooling: A GlobalAveragePooling2D layer was used to condense the feature maps into a vector representation, minimizing overfitting.

d) Fully Connected Layers: A dense layer with 512 neurons along with Batch-Normalization is used to to stabilize model learning. A final dense layer with 38 neurons one for each class of disease and healthy leaf, matching the number of output classes, uses a softmax activation function for multi-class classification.

e) Regularization: Dropout layers were inserted to reduce overfitting of model by randomly deactivating 50% of neurons during training.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_1 (Conv2D)	(None, 72, 72, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_2 (Conv2D)	(None, 34, 34, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 128)	0
conv2d_3 (Conv2D)	(None, 15, 15, 256)	295,168
max_pooling2d_3 (MaxPooling2D)	(None, 7, 7, 256)	0
conv2d_4 (Conv2D)	(None, 5, 5, 512)	1,180,160
max_pooling2d_4 (MaxPooling2D)	(None, 2, 2, 512)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 512)	0
dense (Dense)	(None, 512)	262,656
batch_normalization (BatchNormalization)	(None, 512)	2,048
activation (Activation)	(None, 512)	0
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 38)	19,494

Total params: 1,852,774 (7.07 MB)
Trainable params: 1,851,750 (7.06 MB)
Non-trainable params: 1,024 (4.00 KB)

Fig. 3. CNN_Model_Architecture

2) Training and Optimization: The model was compiled with the following settings:

- **Optimizer:** Adam optimization algorithm was used because of its adaptive learning rate and efficiency in computation .
- **Loss Function:** Categorical cross-entropy was used, which is an appropriate choice for multi-class classification tasks.
- **Metrics:** Accuracy, precision, and recall were calculated to evaluate performance.

The model was trained for 10 epochs with a batch size of 128. Additional methods such as EarlyStopping and ReduceLROnPlateau callbacks were utilized to prevent overfittingof the model and adapt the learning rate based on validation loss.

3) Evaluation: To assess the model's performance:

- Training and validation accuracy / loss was monitored and plotted across epochs.
- The model was evaluated on a hold-out test set(unseen test data).

B. VGG16 Model

For the VGG16 Model, we implemented the supervised learning framework with the transfer learning methodology. The reason for this approach was the availability of labeled data—pictures of plant leaves annotated with their respective disease classes—which makes supervised learning most fitting. We decided to perform transfer learning to fully exploit the pre-existing knowledge encapsulated within VGG16, which has shown considerable effectiveness in image recognition from prior experiences with large-scale image datasets.

1) Model Design: The VGG16 model pre-trained on the ImageNet dataset was used as the base model. The top layers were removed, and a custom classifier was added:

- A Flatten layer is used to flatten the feature maps into a one-dimensional vector.
- A fully connected Dense layer with 38 units, matching the number of disease classes, uses a softmax activation function for multi-class classification.
- The base layers of VGG16 were frozen, preserving the feature extraction capabilities of the pre-trained network while training only the added layers.

2) Training and Optimization: The model was compiled with the following settings:

- **Optimizer:** Adam optimization algorithm was chosen for its adaptive learning rate and computational efficiency.
- **Loss Function:** Categorical cross-entropy was used, as it is appropriate for multi-class classification tasks.
- **Metrics:** Accuracy was used to evaluate model performance.

The model was trained for 5 epochs with a batch size of 128. Validation was performed on a hold-out validation set to monitor the model's performance and ensure it did not overfit.

3) Evaluation: To assess the model's performance:

- Training and validation accuracy/loss were monitored and plotted across epochs to visualize model performance.
- A test image was finally shown to the trained model for prediction, demonstrating its ability to classify plant diseases.

C. SVM

The goal is to classify images of plants for disease detection using supervised learning. Since labeled data is available—healthy or diseased classes—this approach is ideal for training models like Support Vector Machines (SVM) to recognize patterns and make accurate predictions of the labels.

1) Model Design: We choose an SVM with a Radial Basis Function (RBF) kernel as it can work in high dimensional non-linear feature space efficiently. The RBF kernel enhances pattern recognition in complex data.

2) Training and Optimization: The model was compiled with the following settings:

- **Data Preprocessing:**
  - Images resized to 128×128 pixels for uniformity.
  - Flattened into 1D arrays for use with SVM.
  - Pixel values normalized to a range of [0, 1].
- **Advanced feature extraction:** Histogram of Oriented Gradients (HOG) captures image texture and structure by analyzing gradient distributions. Key steps include gradient computation, spatial binning, block normalization, and feature vector construction. HOG provides compact and effective image representations, ideal for classification tasks.
- **Data Spilt:** 80% of the train data is used for training (i.e. 56,251 images) and 20% for validation (i.e. 14,062 images).

3) Evaluation: To assess the model's performance:

- After training, the model was tested on another dataset of 17,572 images to validate its performance in classifying plant images it had never seen before. This showed how the model was accurate in predicting whether a plant is healthy or diseased based on the features learned.

D. ResNet

Next, we chose ResNet-18, a deep learning model known for addressing challenges like vanishing gradients in deep neural networks through residual connections. ResNet-18 is a well-established architecture for image classification tasks and comes pre-trained on the ImageNet dataset, which helps achieve faster convergence and better feature extraction.

1) Steps of the ResNet-18 Approach:

- **Input Layer:** The input images are resized to 128x128 pixels and normalized for uniformity.
- **Feature Extraction:** The model leverages convolutional layers to extract low-level and high-level features such as edges, textures, and patterns. Residual Blocks allow skip connections to bypass one or more layers, learning residuals instead of direct transformations.
- **Global Average Pooling (GAP):** Reduces the feature map size to a single vector per class, simplifying the classification process.
- **Fully Connected Layer:** The pre-trained fully connected layer is replaced to output probabilities for 38 plant disease classes.
- **Output:** The model predicts a given input image's most probable disease class.

2) Why ResNet-18?:

- **Robustness:** ResNet-18 is designed to handle deep architectures effectively and prevent vanishing gradients, which is crucial for large datasets.
- **Pretrained Weights:** The model generalizes more quickly when pre-trained ImageNet weights are used since they transmit knowledge from a much larger dataset.
- **Alignment with Problem:** Plant disease classification involves identifying fine-grained features, which ResNet-18 can effectively capture.

3) Implementation Design: To ensure an effective and systematic workflow, the following steps were taken:

- **Training and Evaluation:**
  - Training: The model was trained over 10 epochs using a batch size of 64.
  - Loss Function: CrossEntropyLoss was used to compute classification error.
  - Optimizer: Adam optimizer was employed for efficient gradient updates.
  - Learning Rate Scaling: The learning rate was adjusted dynamically using a scheduler.
  - Validation: Validation accuracy and loss were monitored after each epoch to evaluate performance.
- **Testing:** We used the test dataset for evaluation of the model's performance and metrics like accuracy, recall, precision, and F1-score were reported. For this a confusion matrix was plotted to analyze the classification performance for every class.

E. Random Forest

We have implemented a Random Forest classifier to classify plant leaf images into two categories: Healthy and Unhealthy. Using 100 decision trees, each decision tree is trained on a bootstrapped sample, which involves using samples from the dataset with replacement. It uses majority voting to aggregate predictions, assigning the final class label to the category with the highest votes. This method helps reduce variance and improve overall accuracy.

- **Training:**
  - Each tree in the forest independently makes predictions for the given input data.
  - Bootstrapping involves sampling the data with replacement, so that each decision tree sees a slightly different version of the dataset.

- By combining the results of multiple decision trees, it reduces the overfitting of individual data points and adapts to better match unseen information.

- **Evaluation:**
  - OOB Score(Out-of-Bag) Accuracy - The samples that were left out during bootstrapping process is used to check the accuracy of the model.
  - After this phase, a newer unseen dataset is used for the final phase of the testing.

VIII. Result, Analysis and Findings

Model	Accuracy	Precision	Recall	F1-Score
ResNet	98.28%	0.98	0.98	0.98
CNN	94.12%	0.94	0.93	0.935
VGG16	93.86%	0.03	0.03	0.03
Random Forest	70.85%	0.71	0.71	0.71
SVM	72.23%	0.71	0.72	0.71

TABLE I
Performance Comparison of Machine Learning Models

A. *Performance Comparison Across Models:*

ResNet achieves the highest performance across all metrics, with an Accuracy of 98.28%, Precision of 0.98, Recall of 0.98, and F1-Score of 0.98, indicating its superior capability to correctly classify the majority of both positive and negative instances.CNN performs second best with an Accuracy of 94.12%, Precision of 0.94, Recall of 0.93, and F1-Score of 0.935, making it a highly reliable model. However, it is slightly less robust compared to ResNet.VGG16, despite having an Accuracy of 93.86%, demonstrates poor class-specific performance with a Precision, Recall, and F1-Score of 0.03 each, suggesting potential issues in model training or evaluation that may have caused this result.Random Forest and SVM deliver significantly lower performance compared to deep learning models, with Accuracy values of 70.85% and 72.23%, respectively. Both models have similar Precision and Recall (around 0.71 and 0.72), with F1-Scores of 0.71, reflecting their limited capability in handling the complexity of the dataset.

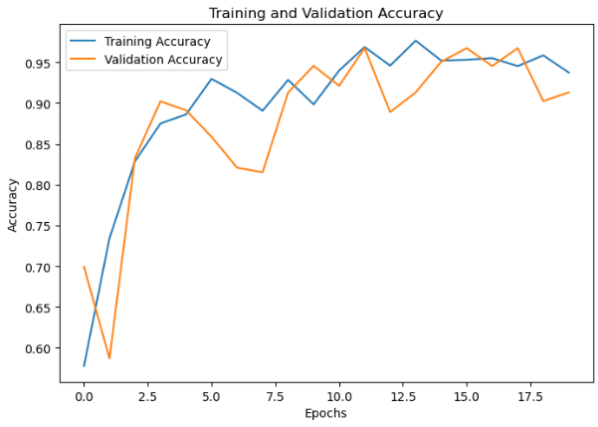

Fig. 4. CNN Accuracy

Fig. 5. CNN Validation Loss

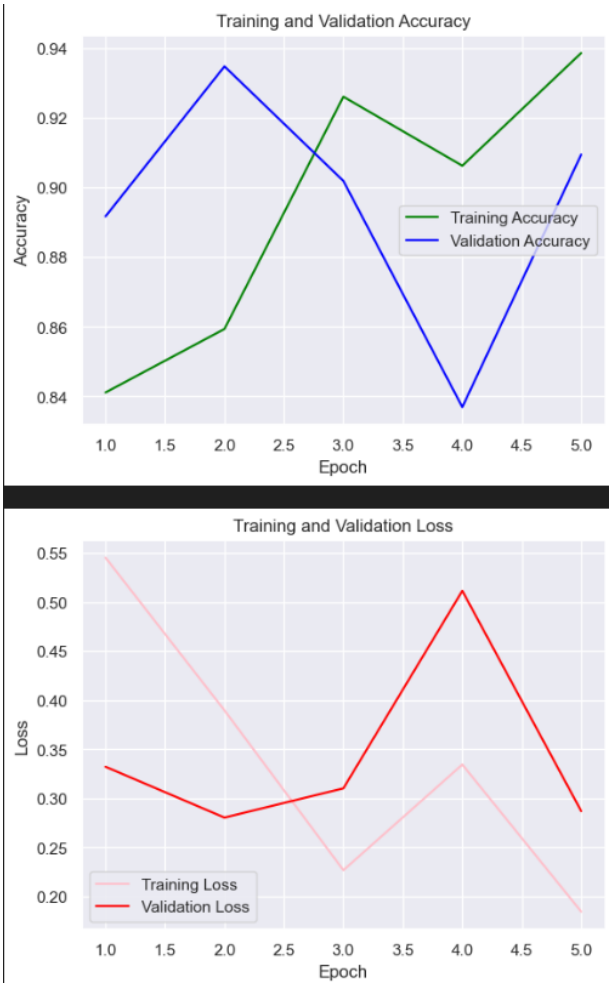

Fig. 6. VGG16 Accuray and Validation Loss

B. *Trade-offs Between Metrics:*

The high Precision and Recall for ResNet, CNN suggest these models balance minimizing false positives and false negatives effectively, making them suitable for applications where both errors are critical. Random Forest and SVM show

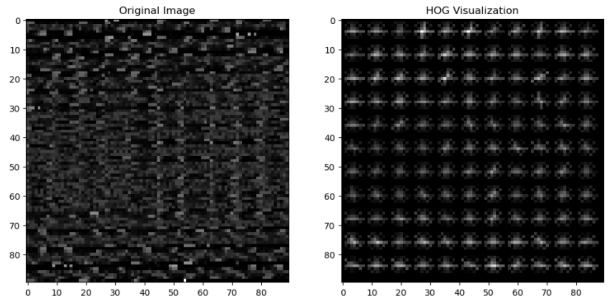

Fig. 7. SVM HOG Extraction

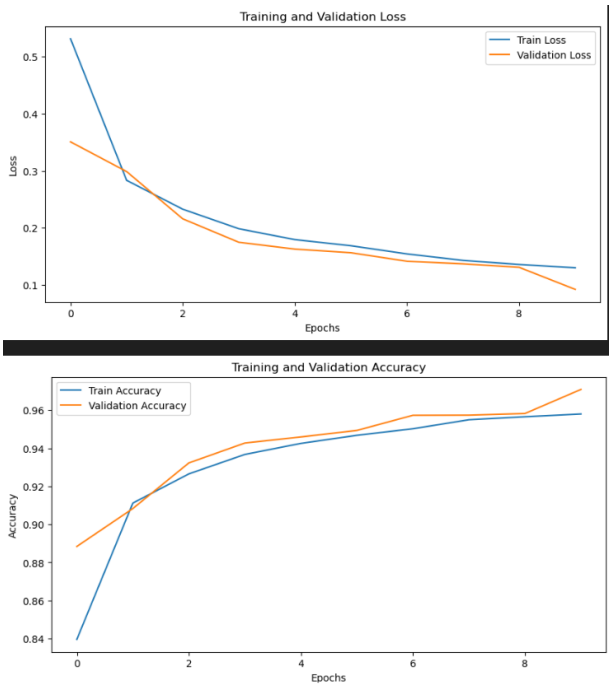

Fig. 8. ResNet Accuray and Validation loss

balanced but low Precision and Recall, highlighting difficulty in distinguishing classes effectively. VGG16 has high accuracy but low precision and recall which needs to be analyzed

C. Deep Learning vs. Traditional Machine Learning:

The results clearly indicate that deep learning models (CNN, VGG16, ResNet) outperform traditional machine learning models (Random Forest, SVM). This is likely due to the ability of deep learning architectures to learn hierarchical and complex feature representations from the data.

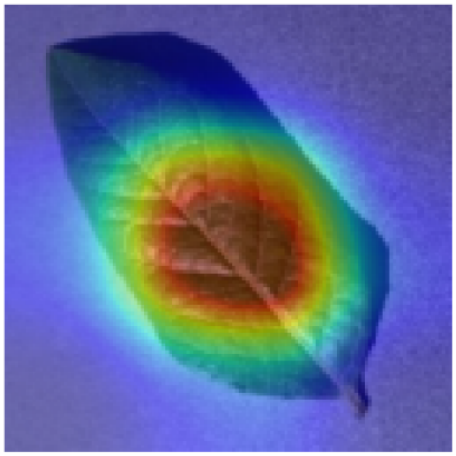

Fig. 9. Heat map using GradCA

IX. Discussion

A. Challenges and Solutions

1) Data Imbalance: Predictions were skewed because some disease classes had fewer samples. To solve this we used data augmentation to create more samples and balance the collection.

2) Deep Model Overfitting: On the training set, models such as CNN and VGG16 showed overfitting. To avoid overfitting and enhance generalization, dropout, batch normalization, and early stopping were used.

3) High Dimensionality and Complexity: Random Forest and SVM struggled to handle the complexity of picture attributes. To tackle this we used dimensionality reduction techniques like SVM HOG features and Random Forest scaling.

B. Limitations

- **Computational Cost:** Training deep learning models like Custom CNN, ResNet, VGG16, SVM, and RF required significantly high computational resources and time.
- **Model Complexity:** The complexity of deep models makes them less interpretable than traditional models like Random Forest.
- **Generalization:** Random Forest and SVM showed limited ability to generalize for unseen test data.

C. Improvements and Outcomes

1) Transfer Learning:

- ResNet-18 and VGG16 utilized pre-trained weights, significantly improving convergence speed and accuracy.
- ResNet had the greatest F1-Score (0.98), and these models often beat conventional methods.

2) Explainability:

- Grad-CAM heatmaps were used with ResNet-18 to provide visual explanations, highlighting the regions contributing most to the model's predictions.

3) Validation of Traditional Models:

- Random Forest and SVM established a useful benchmark for comparison, despite their subpar performance relative to deep models.

D. Future Directions

1) Ensemble Methods: Combining deep and conventional model predictions could capitalize on each method's advantages.

2) Larger Dataset: Adding more varied samples to the dataset helps enhance generalization for all models.

3) Deployment: Optimization techniques will be investigated to enable deployment on mobile or edge devices for use in agricultural fields.

E. Novelty Statement

In this project, a multi-model comparative approach was employed, evaluating both traditional machine learning models (Random Forest and SVM) and deep learning architectures (CNN, VGG16, and ResNet-18) on the same plant disease classification dataset. This thorough comparison shows the relative benefits and drawbacks of several strategies addressing this challenging issue, in contrast to many previous research that just concentrate on standard or deep learning models. We have used custom-designed CNN instead of relying on pre-trained architectures. This approach allows greater flexibility in tailoring the model to the specific characteristics of the dataset, resulting in high performance and demonstrating the potential of custom deep learning solutions.

The project also uses Grad-CAM heatmaps for ResNet-18, which provide a novel explainability feature that allows users to see which aspects of the input photos were most important in making the predictions. By enabling end users, such as farmers or agronomists, to understand the categorization process, this improves model transparency and solves a significant deep-learning problem.

Finally, the project shows the versatility of conventional models with appropriate feature engineering by integrating HOG feature extraction for SVM, which is less frequently investigated in plant disease classification applications. These components work together to provide a comprehensive approach that distinguishes it from previous studies by fusing interpretability, model diversity, and technical rigor.

X. Group Contributions

A. Collaborative Efforts

The group worked both collaboratively and independently on the project at all times, ensuring comprehensive engagement in every aspect. Each member was involved in the discussions and contributed to the selection of methodologies and approaches. The team reviewed several research papers, collected and analyzed data, and tried different modeling techniques together to come up with the best results. All members contributed significantly to the documentation and analysis of results.

B. Individual Contributions

- **Varun:** Applied the Random Forest model by optimizing the algorithm and integrating it with the main framework of the project.
- **Om:** Focused on the development and optimization of the Support Vector Machine (SVM) model to make it robust regarding classification tasks.
- **Niket:** Designed and implemented a customized CNN model by modifying the architecture based on the dataset's special needs.
- **Soureesh:** Implemented the VGG architecture and modified it to adaptively work with feature extraction so that the performance on the dataset would improve.
- **Aryan:** Developed and fine-tuned the ResNet model for better performance on complex data with its residual learning framework.

XI. Conclusion

Plant disease classification is one of the big challenges that must be addressed for agriculture and food security in the world. This project studied different machine learning and deep learning techniques to propose effective solutions. Various models were implemented, including Random Forest, Support Vector Machines, custom Convolutional Neural Networks, and pre-trained architectures such as VGG and ResNet.

Advanced models of deep learning have outperformed, in the experiments, traditional methods such as Random Forest and SVM. Such models extracted high-order features from agricultural datasets with improved performance. Domain adaptation with transfer learning significantly boosted the results by utilizing the pre-trained models for similar datasets, reducing much of the computation at negligible loss in accuracy.

Valuable insights into complications within the current classification of plant diseases based on different projects in addressing challenges related to aspects such as computational efficiency, hyperparameter optimization, importance of transfer learning, and feature engineering toward a reliable, accurate outcome.

This project gave us real insights into how to solve a real-world problem step by step, starting from data preparation, model selection, evaluation, and refinement. Collaboration among members has played a great role in the exploration of different techniques to reach a unified and high-performance solution. In general, this work has achieved the goal of plant disease classification with high accuracy and reliability, hence providing a concrete framework for further research. The future directions could be the deployment of these models in real-world agricultural scenarios, lightweight architectures for mobile platforms, and incorporation of other data modalities such as soil characteristics and climatic conditions that may further improve the performance and applicability.

XII. Source Code and Dataset

The source code and dataset used in this research are publicly available and can be accessed using the links below:

- **Source Code:** https://github.com/Niket0702/Plant-Disease-Classification-using-Advanced-Machine-Learning-Techniques-.git
- **Dataset:** https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset

REFERENCES

[1] Shoaib M, Shah B, EI-Sappagh S, Ali A, Ullah A, Alenezi F, Gechev T, Hussain T and Ali F (2023) An advanced deep learning models-based plant disease detection: A review of recent research. Front. Plant Sci. 14:1158933. doi: 10.3389/fpls.2023.1158933

[2] L. Li, S. Zhang and B. Wang, "Plant Disease Detection and Classification by Deep Learning—A Review," in IEEE Access, vol. 9, pp. 56683-56698, 2021, doi: 10.1109/ACCESS.2021.3069646

[3] Liu, J., & Wang, X. (2021). Plant diseases and pests detection based on deep learning: a review. Plant Methods, 17(1). doi:10.1186/s13007-021-00722-9

[4] Jafar A, Bibi N, Naqvi RA, Sadeghi-Niaraki A and Jeong D (2024) Revolutionizing agriculture with artificial intelligence: plant disease detection methods, applications, and their limitations. Front. Plant Sci. 15:1356260. doi: 10.3389/fpls.2024.1356260

[5] S. Ramesh et al., "Plant Disease Detection Using Machine Learning," 2018 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C), Bangalore, India, 2018, pp. 41-45, doi: 10.1109/ICDI3C.2018.00017.

[6] Francis, M., & Deisy, C. (2019). Disease Detection and Classification in Agricultural Plants Using Convolutional Neural Networks — A Visual Understanding. 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN). doi:10.1109/spin.2019.8711701