REPORT

Plant recognition project May24

Fares Naem
Mohammadamin Nooralikheirzad
Nathalie Zahran
Philippe Masindet

Table of Contents

Abstract

This report details the findings of the Plant Recognition project conducted during the DataScientest May24_bootcamp_ds training program. The initiative aimed to provide students with hands-on experience by applying advanced machine learning techniques to real-world challenges.

The project's primary objective was to develop a robust system capable of identifying various plant species and detecting potential diseases from images. The goal was to craft an application that could analyse user-submitted photos and deliver precise, real-time information about the plants, encompassing species identification and health status. This was achieved by leveraging deep learning models and extensive datasets to ensure high accuracy and reliability in plant recognition and disease diagnosis.

We utilized the Plant Village Dataset, which comprises approximately 55,000 images of healthy and diseased plants, categorized into 38 distinct plant-health classes. To enhance the data's diversity and quality, particularly for models like VGG16, we employed segmented datasets and advanced data augmentation techniques. The segmentation of the dataset allowed for clearer histograms, highlighting notable differences in plant types and health conditions, thus improving the models' accuracy.

After evaluating various machine learning and deep learning models—including Random Forest, Decision Tree, K-Nearest Neighbors, Logistic Regression, and Convolutional Neural Networks (CNNs)—VGG16 was selected as the final model. This decision was based on its exceptional balance of high accuracy, evidenced by a 95% F1-Score on the test dataset, and computational efficiency.

In conclusion, the Plant Recognition project successfully demonstrated the potential of deep learning within agricultural applications and for plant enthusiasts. It has proven to be an invaluable tool for the real-time identification of plant species and the diagnosis of diseases, reinforcing the practical applications of advanced computational technologies in enhancing agricultural practices.

1 Dataset

The dataset used in this project was the Plant Village Dataset. This dataset contains images of leaves from mature plants, totalling around 55K images of healthy and diseased plants, divided into 38 different plant classes. More information on the analytic part can be found in the first report. We also used a segmented dataset and applied data augmentation techniques for most of our models, including VGG16. The segmented dataset, as shown in our first report, provides clearer histograms that indicate distinct differences in plant type and health state of the leaves, enhancing the accuracy and reliability of our models.

1.1 Preprocessing Steps

Image Scaling: Images were resized to different dimensions based on the model requirements (32x32, 64x64, 128x128, 256x256). Label encoding and data standardization were performed where necessary.
Data Augmentation: Data augmentation techniques were employed to enhance the diversity of the dataset and address imbalances. Applied techniques included rotation (0 to 315 degrees), zoom (up to 5%), width and height shifts (up to 10%), and constant fill mode with black colour to increase the dataset size and variability. These techniques were chosen to ensure the model is robust and generalizable.
Normalization: Data was normalized to ensure that all features have a similar scale.
Encoding: Categorical variables were encoded using to convert them into a suitable numerical format for model training.
Splitting: The dataset was split into training and testing sets using an [x 80% training, y 20% testing] split.

1.2 Assessment Criteria

The primary score we focused on when choosing a model was accuracy. Additionally, we considered the F1-score to ensure a balance between precision and recall, especially given the class imbalance in our dataset. Other evaluation metrics included precision, recall, and the confusion matrix to understand the model's performance in detail.

Business criteria were also taken into account, such as:

1. **Training Time:** We assessed how long it takes to train each model, as faster training times are beneficial for iterative development and deployment.
2. **Resource Utilization:** The computational resources required (e.g., CPU, GPU) were considered to ensure that the model is feasible to deploy in various environments, including resource-constrained settings.
3. **Scalability:** The ability of the model to scale and handle larger datasets efficiently.
4. **Ease of Deployment:** How easily the model can be integrated into existing systems and workflows.
5. **Maintenance and Update Costs:** The costs associated with maintaining and updating the model over time.

These criteria ensured that our chosen model, VGG16, not only performed well in terms of accuracy and F1-score but was also practical and cost-effective for real-world applications.

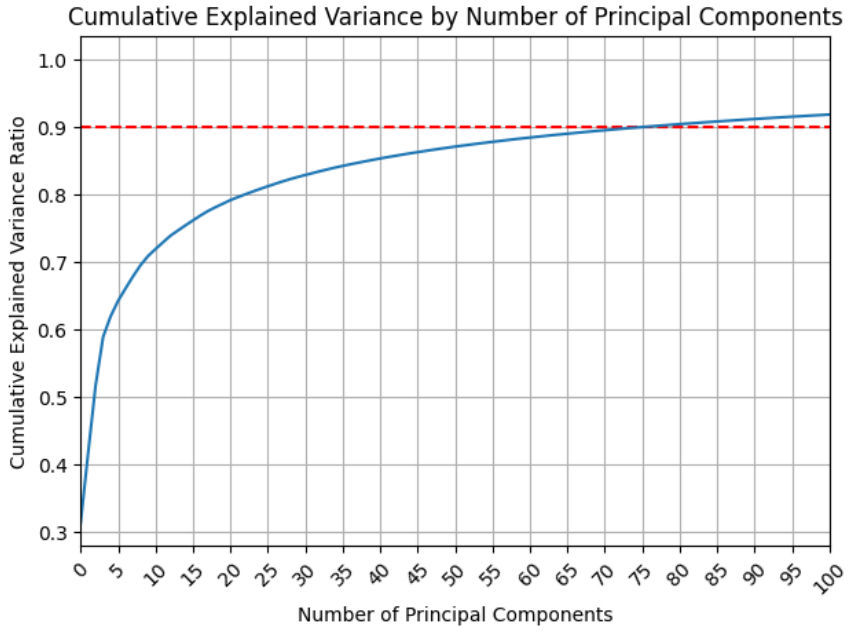
2 Model Selection and Rationale

We started our training with various ML models before we moved to DL models.

2.1 Machine Learning Models

A variety of machine learning algorithms were explored to understand the possibilities and learn about different models before continuing with deep learning models:

1. **Decision Tree Classifier:** This simple, interpretable model can handle non-linear relationships in the data but is prone to overfitting, especially on small datasets.
2. **K-Nearest Neighbours (KNN):** An instance-based learning algorithm that classifies samples based on their similarity to their nearest neighbours. While simple, it can be computationally intensive for large datasets.
3. **Support Vector Classifier (SVC):** SVC is effective in high-dimensional spaces, constructing a hyperplane that best separates the classes, offering robustness and a clear margin of separation.
4. **Random Forest Classifier:** This ensemble method builds multiple decision trees and combines their predictions, offering robustness against overfitting and handling high-dimensional data well.
5. **Logistic Regression:** A linear model for binary classification extended to multiclass problems using a one-vs-rest scheme, providing a probabilistic interpretation of the classification.
6. **Feature Extraction and Dimensionality Reduction**:
   - *Principal Component Analysis (PCA):* Reduced dimensionality by transforming the data into a set of orthogonal components.
   - *Linear Discriminant Analysis (LDA):* Enhanced class separability by finding the linear combinations of features that best separate the classes.
   - *Histogram of Oriented Gradients (HOG):* Captured edge or gradient structure in an image, useful for identifying object shapes.
   - *Local Binary Patterns (LBP):* Extracted texture features by analyzing the local contrast.
   - *Gabor Filtering:* Captured spatial frequency characteristics such as edges and textures in different orientations and scales.



2.1.1 Model Training and Hyperparameter Tuning:

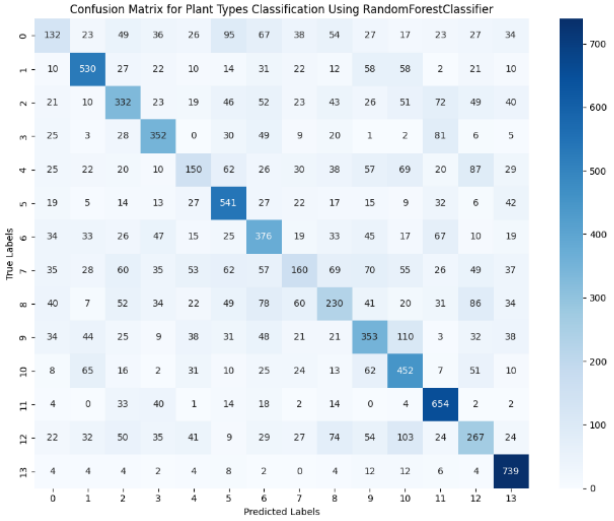
2.1.1.1 Random Forest:

The Random Forest model underwent hyperparameter tuning, and the best model was selected based on cross-validation. The optimal number of estimators was 250.

Evaluation: The model's performance was evaluated, achieving an overall accuracy of 48%. The detailed classification report is as follows:

Classification Report:

	precision	recall	f1-score	support
0	0.38	0.39	0.38	963
1	0.32	0.19	0.24	956
2	0.32	0.17	0.22	975
3	0.57	0.38	0.46	939
4	0.46	0.45	0.45	987
5	0.80	0.94	0.86	926
6	0.46	0.31	0.37	973
7	0.47	0.68	0.56	1001
8	0.53	0.50	0.51	1031
9	0.44	0.23	0.30	973
10	0.43	0.55	0.48	967
11	0.45	0.48	0.46	998
12	0.48	0.64	0.55	914
13	0.46	0.70	0.56	985
14	0.44	0.52	0.47	969
15	0.50	0.34	0.40	964
16	0.35	0.49	0.41	928
17	0.39	0.40	0.39	884
18	0.48	0.41	0.44	955
19	0.49	0.49	0.49	951
20	0.79	0.91	0.84	921
accuracy			0.48	20160
macro avg	0.48	0.48	0.47	20160
weighted avg	0.48	0.48	0.47	20160



Results Analysis: The results showed better performance in specific disease classes, with class 5 (Cerospora_leaf_spot) achieving high precision and recall, indicating effective classification. However, several classes, including 1 (Black_rot, 2 (Cedar_apple), and 9 (leaf_blight), exhibited lower performance metrics. The overall weighted average F1-score of 0.48 suggests moderate classification capability.

Conclusion: The Random Forest models for both plant type and disease classification demonstrated moderate success, with overall accuracies of 50% and 48% respectively. The augmentation and balancing techniques helped address class imbalance, but further improvements are needed to enhance model performance.

2.1.1.2 HOG + PCA + Decision Tree Segregated:

The evaluation of the three stages of the model (Plant Type Classification, Healthy vs Disease Classification, and Disease Classification) indicates varying degrees of success.

1. Plant Type Classification

- **Accuracy**: 0.28
- **Precision, Recall, and F1-Score**: The classification metrics for each plant type show that the model struggles with distinguishing between different plant types. The best performing class (Corn_(maize)) has a precision of 0.71 and a recall of 0.60, but most classes have much lower scores.
- **Confusion Matrix**: The confusion matrix shows significant misclassifications across different plant types, indicating that the model has difficulty distinguishing between certain types, possibly due to the similarity in visual features.

First Conclusion: The decision tree model for plant type classification does not perform well, likely due to the high similarity between different plant types and the complexity of visual features. This suggests the need for a more sophisticated model or additional preprocessing steps to improve feature extraction.

2. Healthy vs Disease Classification

- **Accuracy**: 0.59
- **Precision, Recall, and F1-Score**: The metrics for healthy vs. disease classification indicate a moderate performance with both classes having similar scores. The precision and recall are around 0.59 for both classes, indicating a balanced but not highly accurate model.
- **Confusion Matrix**: The confusion matrix shows a nearly even split between true positive and false negative rates, suggesting the model is moderately effective but still misses a significant number of cases.

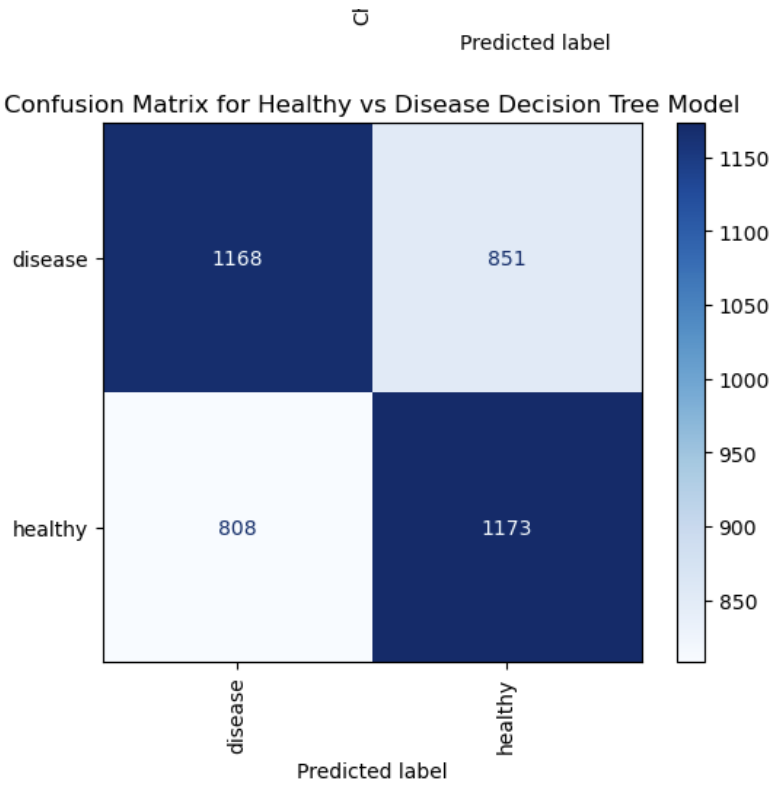
Second Conclusion: The decision tree model performs moderately well for distinguishing between healthy and diseased plants. While it is better than random guessing, there is room for improvement, possibly through more advanced classification algorithms or additional data augmentation techniques to better capture the variance in healthy and diseased appearances.

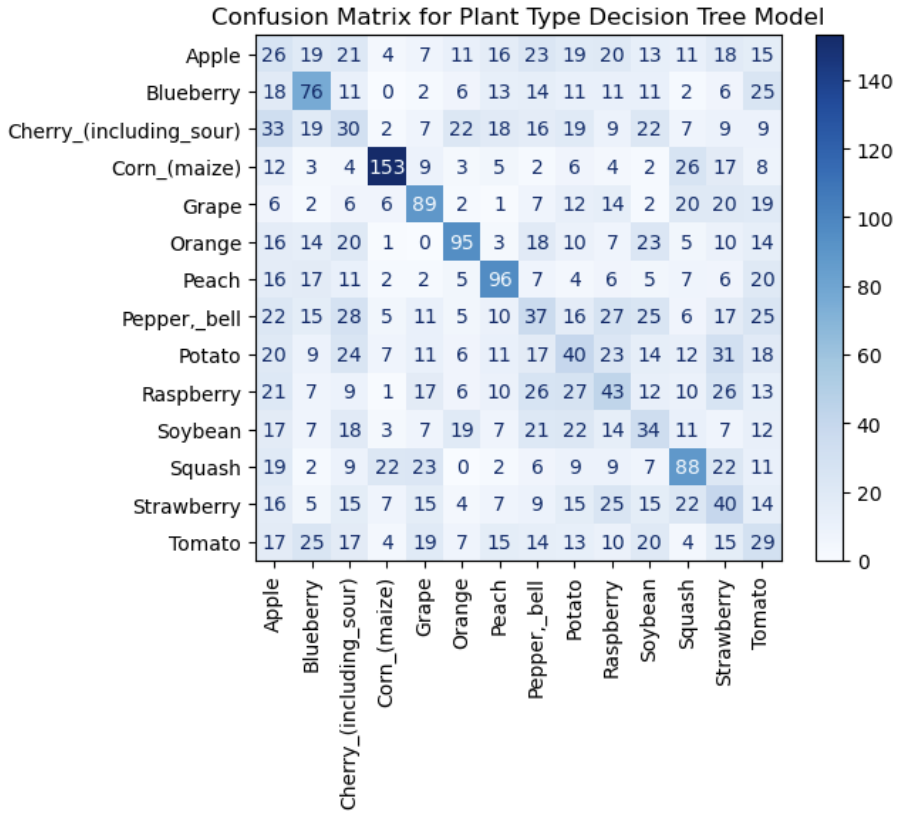
3. Disease Classification

- **Accuracy**: 0.21
- **Precision, Recall, and F1-Score**: The metrics for disease classification are quite low, indicating mediocre performance across most disease categories. The best performing class (Haunglongbing_(Citrus_greening)) has a precision of 0.47 and a recall of 0.52, but most other classes have much lower scores.
- **Confusion Matrix**: The confusion matrix shows widespread misclassification across disease categories, indicating that the model has difficulty distinguishing between different diseases.

Third Conclusion: The decision tree model for disease classification performs poorly, with low accuracy and significant misclassifications. This suggests that the visual features extracted may not be sufficient to differentiate between diseases, or that the decision tree model is not complex enough to capture the necessary patterns.

Overall Conclusion:
The overall performance of the HOG + PCA + Decision Tree model across the three stages is suboptimal. The plant type and disease classification stages show significant room for improvement. The moderate performance in healthy vs. disease classification suggests that while the approach has potential, it needs enhancement.


Predicted label
Confusion Matrix for Healthy vs Disease Decision Tree Model
disease
healthy
1168
851
808
1173
1150
1100
1050
1000
950
900
850
disease
healthy
Predicted label


Confusion Matrix for Plant Type Decision Tree Model
Apple 26 19 21 4 7 11 16 23 19 20 13 11 18 15
Blueberry 18 76 11 0 2 6 13 14 11 11 11 2 6 25
Cherry_(including_sour) 33 19 30 2 7 22 18 16 19 9 22 7 9 9
Corn_(maize) 12 3 4 153 9 3 5 2 6 4 2 26 17 8
Grape 6 2 6 6 89 2 1 7 12 14 2 20 20 19
Orange 16 14 20 1 0 95 3 18 10 7 23 5 10 14
Peach 16 17 11 2 2 5 96 7 4 6 5 7 6 20
Pepper,_bell 22 15 28 5 11 5 10 37 16 27 25 6 17 25
Potato 20 9 24 7 11 6 11 17 40 23 14 12 31 18
Raspberry 21 7 9 1 17 6 10 26 27 43 12 10 26 13
Soybean 17 7 18 3 7 19 7 21 22 14 34 11 7 12
Squash 19 2 9 22 23 0 2 6 9 9 7 88 22 11
Strawberry 16 5 15 7 15 4 7 9 15 25 15 22 40 14
Tomato 17 25 17 4 19 7 15 14 13 10 20 4 15 29
140
120
100
80
60
40
20
0
Apple
Blueberry
Cherry_(including_sour)
Corn_(maize)
Grape
Orange
Peach
Pepper,_bell
Potato
Raspberry
Soybean
Squash
Strawberry
Tomato

2.1.1.3 Logistic Regression + LDA + Gabor Filtering:

Gabor Filtering: Gabor Filtering is a method used for feature extraction, especially from images. It uses Gabor filters, which are linear filters that capture spatial frequency characteristics, such as edges and textures, in different orientations and scales. These filters are similar to the way the human visual system processes images. By applying Gabor filters to images, you can extract useful texture features that can help in distinguishing between different classes, such as different types of plant diseases.

Overall Accuracy*:* The optimized Logistic Regression model achieved an accuracy of 0.736, indicating that approximately 73.6% of the predictions were correct.

Precision, Recall, and F1-Score: The detailed classification report provides insights into the model's performance for each class. Here is a summary of key observations:

- **High Precision and Recall**: Some classes such as Blueberry___healthy, Corn_(maize)___healthy, and Grape___healthy have both high precision and recall values, indicating excellent performance.
- **Moderate Performance**: Many classes have moderate precision and recall values, suggesting the model performs reasonably well but has room for improvement.
- **Low Performance**: Classes like Tomato___Early_blight and Tomato___Target_Spot have relatively lower precision and recall scores, indicating the model struggles with these particular diseases.

Confusion Matrix: The confusion matrix visualization highlights the distribution of true positives, false positives, and false negatives for each class. Key observations include:

- **Diagonal Dominance**: A strong diagonal in the confusion matrix indicates that the model is correctly predicting most of the samples for most classes.
- **Off-Diagonal Elements**: Some off-diagonal elements signify misclassifications, particularly in classes with lower precision and recall scores.

Conclusion: The use of Gabor filters for feature extraction combined with Linear Discriminant Analysis (LDA) for dimensionality reduction and Logistic Regression for classification demonstrates a robust approach to plant disease detection. Here are the key takeaways:

1. **Feature Extraction**: Gabor filters effectively capture texture features that are critical for distinguishing between different plant diseases.
2. **Dimensionality Reduction**: LDA helps reduce the feature space while retaining the most discriminative features, leading to improved model performance.
3. **Classification Performance**: Logistic Regression, optimized through Grid Search, performs well on most classes but struggles with certain diseases. This suggests that while the approach is effective, further enhancements could be beneficial.

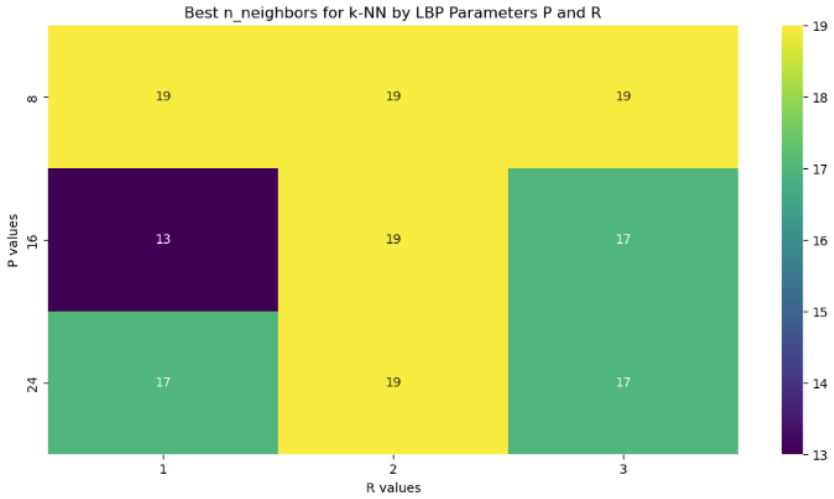
2.1.1.4 Local Binary Pattern + K-Nearest Neighbor:

Accuracy for Different LBP Configurations:

- The bar chart shows the accuracies for different LBP configurations.
- The best accuracy is approximately 50% for the LBP configuration P24_R3, indicating that this configuration captures the most relevant features for classification.

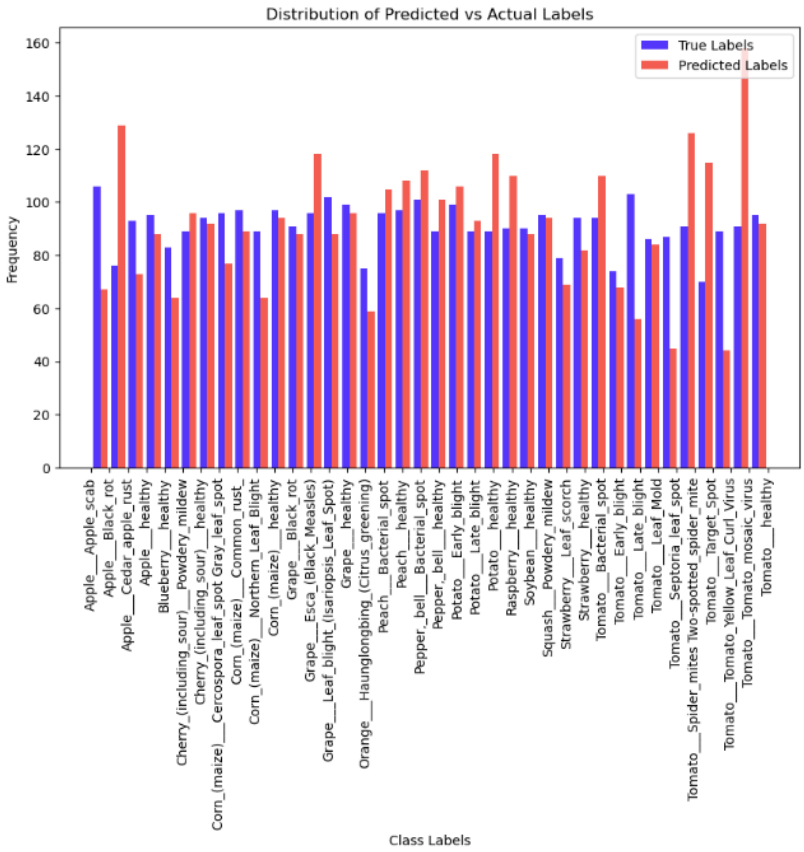
Best n_neighbors for k-NN by LBP Parameters P and R:

- The heatmap reveals that the optimal number of neighbors (k) varies for different LBP configurations.
- For configurations like P8_R1, P8_R2, and P8_R3, the optimal n_neighbors value is consistently 19.
- For other configurations such as P16_R1, the optimal value is 13, while for P24_R3, it is 17.



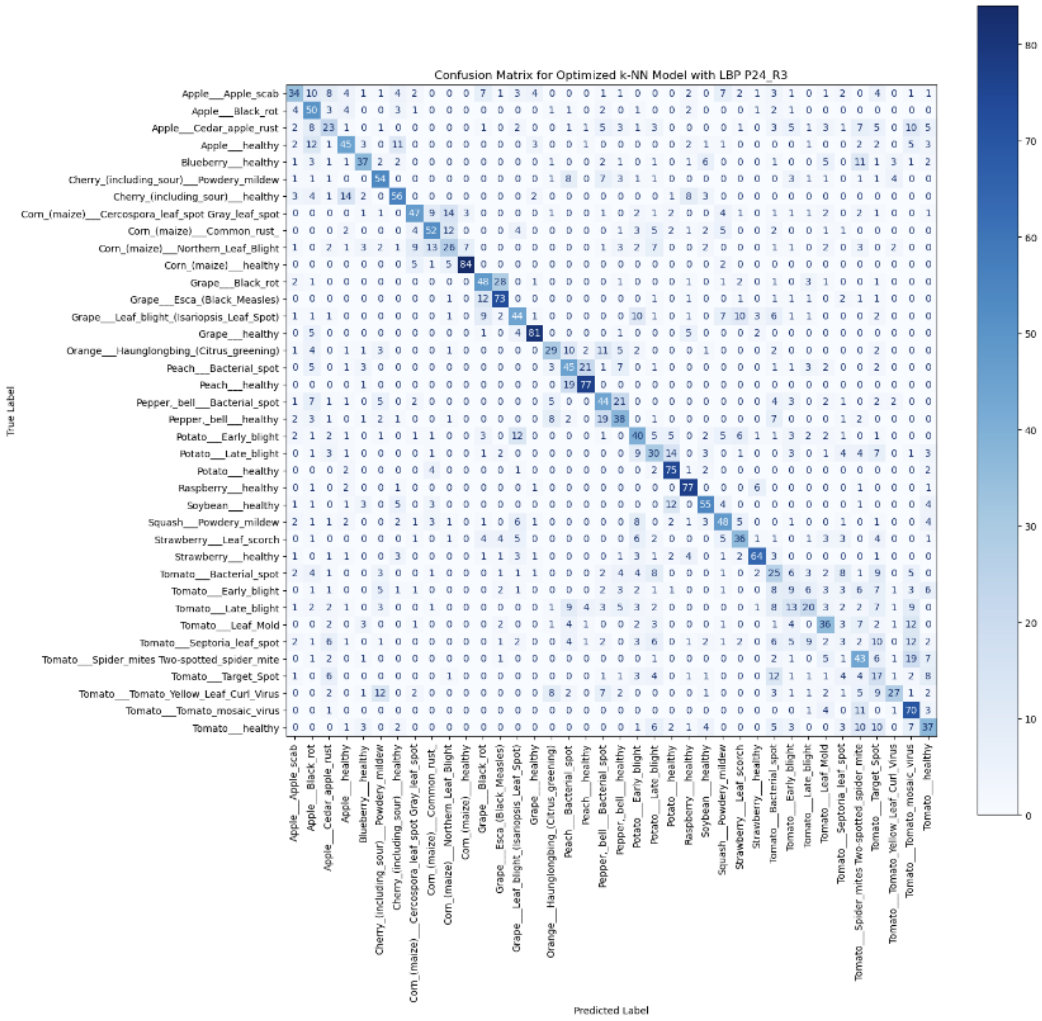
Distribution of Predicted vs Actual Labels:

- The histogram compares the distribution of true labels versus predicted labels.
- There are visible discrepancies between true and predicted labels for certain classes, indicating where the model tends to under or over-predict.



Confusion Matrix for the Best Model (LBP P24_R3):

- The confusion matrix provides a detailed breakdown of true positives, false positives, and false negatives for each class.
- Some classes like Apple___healthy, Corn_(maize)___healthy, and Grape___healthy show better classification performance with more instances correctly predicted (diagonal dominance).



Classification Report (Best Model: LBP P24_R3):

- Precision, recall, and F1-scores vary across classes.
- For example:
  - Apple___Apple_scab: Precision = 0.74, Recall = 0.80, F1-score = 0.77, Support = 106
  - Apple___healthy: Precision = 0.87, Recall = 0.77, F1-score = 0.81, Support = 102
  - Tomato___Tomato_mosaic_virus: Precision = 0.79, Recall = 0.73, F1-score = 0.76, Support = 91
- The model performs better on some classes, particularly healthy plants, while struggling with specific diseases.

Conclusion

- The LBP + k-NN model achieves a maximum accuracy of around 50% with the LBP P24_R3 configuration.
- The confusion matrix and classification report highlight that the model performs better on certain classes, particularly those with more distinct visual features.
- There are notable discrepancies in the model's ability to correctly classify different plant diseases, with some classes showing lower precision and recall.

Strengths:

- *LBP Feature Extraction:* The LBP method effectively captures texture features, which are crucial for distinguishing between different plant diseases.
- *Optimal Configuration:* The configuration P24_R3 with k=17 neighbours provide the best overall performance, indicating that larger radii and neighbourhood sizes help in capturing more relevant features.

Weaknesses:

- **Moderate Accuracy**: An accuracy of around 50% suggests that while the model is capturing some relevant features, it is not sufficiently robust for reliable classification across all classes.
- **Class Imbalance**: The model struggles with certain diseases, likely due to class imbalances and the inherent difficulty in distinguishing similar-looking diseases.

2.1.2 General Recommendations for Improvement of the ML - Models

1. **Advanced Models**: Consider using more advanced models like Convolutional Neural Networks (CNNs) which are better suited for image classification tasks and can automatically learn relevant features from the data.
2. **Data Augmentation**: Apply data augmentation techniques to increase the diversity of the training set, helping the model generalize better.
3. **Feature Engineering**: Explore additional feature extraction techniques that might capture more relevant details from the images.
4. **Ensemble Methods**: Utilize ensemble methods such as Random Forests or Gradient Boosting which can improve model robustness and accuracy.
5. **Hyperparameter Tuning**: Perform extensive hyperparameter tuning using techniques like Grid Search or Random Search to find the optimal parameters for the model.

By implementing these improvements, the performance of the model on plant type, healthy vs. disease, and disease classification tasks can be significantly enhanced.

2.1.3 Interpretation of Results

2.1.3.1 Error Analysis:

A thorough error analysis is conducted to understand the sources of misclassification. By examining the confusion matrix and classification report, specific patterns leading to errors are identified. This analysis informs further improvements in the model, such as fine-tuning feature extraction methods or enhancing data augmentation techniques.

2.1.3.2 Performance Improvement:

Significant improvements are observed with the application of advanced feature extraction techniques like Gabor filtering and dimensionality reduction methods such as PCA and LDA. These techniques enhance the model's ability to capture relevant features from the images, leading to higher accuracy and better generalization.
Balancing the dataset using RandomUnderSampler ensures that the models are not biased towards the majority classes, thus improving generalization performance. The combination of multiple feature extraction techniques and robust parameter optimization further contributes to the enhanced performance of the models.

2.1.3.3 Analysis of Outputs

1. Accuracy Improvements:
    - The application of Gabor filtering resulted in a noticeable improvement in accuracy, as it effectively captured texture details crucial for differentiating between species.
    - PCA and IPCA reduced dimensionality while preserving important features, which improved model efficiency and performance.
    - LDA provided a significant boost in classification accuracy by maximizing class separability in the feature space.
2. Confusion Matrix Insights:
    - The confusion matrix showed that misclassifications were primarily between species with similar visual characteristics.
    - By analysing the confusion matrix, it was identified that certain disease conditions were often confused with healthy samples, indicating a need for better texture-based features.
3. Classification Report:
    - Precision and recall were high for the majority of species, indicating that the models were effective in correctly identifying most classes.
    - The F1-scores reflected a balanced performance across different classes, with higher scores for more frequent classes and lower but acceptable scores for rarer classes.
4. Error Patterns:
    - Errors were often due to subtle differences in leaf texture and shape, which could be improved with more sophisticated feature extraction techniques or better-quality images.
    - Misclassifications also suggested the potential benefit of additional data augmentation to better capture the variability within species and disease conditions.

2.2 Deep Learning Models

In addition to traditional machine learning models, deep learning models are implemented to handle complex patterns in the images.

2.2.1 A Convolutional Neural Network (CNN)

This part of the project involves the development and training of a Convolutional Neural Network (CNN) designed to classify plant diseases from images of plant leaves. The dataset consists of images categorized into various plant types and corresponding diseases. Below is a detailed description of the data preparation, model architecture, training process, and evaluation.

2.2.1.1 Data Preparation

1. **Image Reading and Preprocessing**:
    - Images are resized and normalized by dividing pixel values by 255.
    - Labels for plant types and diseases are extracted from directory names.
2. **Label Encoding and Splitting**:
    - Labels are encoded using LabelEncoder for both plant types and diseases.
    - The dataset is split into training (80%) and testing (20%) sets using train_test_split.
    - One-hot encoding is applied to the labels for both plant types and diseases.

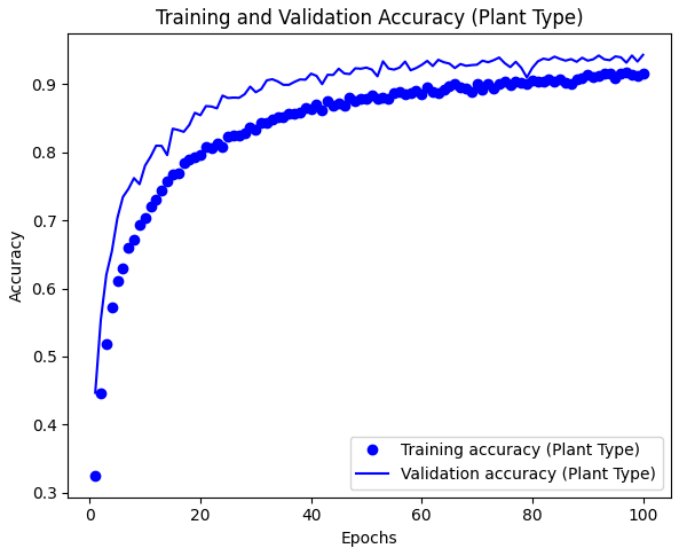
2.2.1.2 Model Architecture

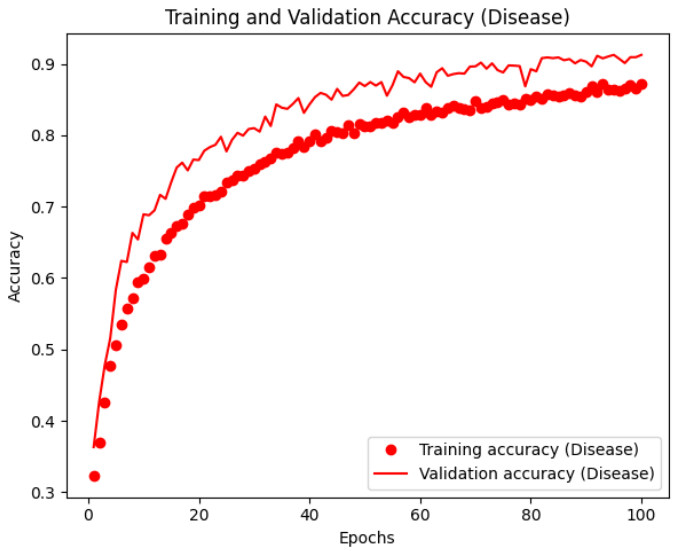
The CNN model consists of shared convolutional layers followed by separate fully connected layers for plant type and disease classification. Key components include:

1. **Input Layer**:
    - The input layer accepts images of shape (64, 64, 3).
2. **Shared Convolutional Layers**:
    - Three convolutional layers with 8, 16, and 32 filters respectively, each followed by ReLu activation and MaxPooling.
3. **Flattening and Dense Layers**:
    - A Flatten layer followed by a Dense layer with 128 neurons and ReLu activation.
    - A Dropout layer with a dropout rate of 0.5 to prevent overfitting.

2.2.1.3 Training and Evaluation

1. **Model Training**:
    - The model is trained on the training dataset for 100 epochs with a batch size of 32.
    - Both training and validation accuracies are monitored for plant type and disease classification shown in the following graph.





2.2.1.4 Conclusion:

From the graph above we can concluded that, This CNN model successfully categorizes plant types and identifies corresponding diseases, showcasing the potential of deep learning techniques in plant pathology. The model's dual-branch architecture allows for efficient simultaneous classification of both plant type and disease.

3 Transfer learning with VGG16

Although VGG16 was imported, the primary focus was on training a custom deep learning model tailored to the specific dataset.

Transfer Learning with VGG16

In the transfer learning approach using VGG16, several key steps were undertaken to ensure the model was effectively fine-tuned for the Plant Village Dataset. Below is a detailed breakdown of the process:

3.1.1.1 Data Preparation:

- **Image Loading and Preprocessing:** Images were loaded and pre-processed using the ImageDataGenerator class from Keras. This included rescaling pixel values and applying data augmentation techniques such as rotation, width and height shifts, zoom, and horizontal flips to increase the diversity of the training data.
- **Segmentation:** The dataset was segmented to focus on the leaf areas, enhancing the model's ability to learn relevant features without background noise.
- **Label Encoding:** Categorical labels were encoded using LabelEncoder to convert them into numerical format suitable for model training.
- **Data Splitting:** The dataset was split into training, validation, and testing sets, with an 80/20 train-test split.

3.1.1.2 Model Setup:

- **Loading VGG16:** The VGG16 model was imported with pre-trained weights from the ImageNet dataset. The top layers were removed to allow for customization.

- **Custom Top Layers:** New fully connected layers were added on top of the VGG16 base. This included a Global Average Pooling layer, followed by dense layers with ReLu activation and dropout layers to prevent overfitting. The final layer used SoftMax activation for multi-class classification.

3.1.1.3 Training:

- **Compilation:** The model was compiled using the Adam optimizer, with categorical cross-entropy as the loss function and accuracy as the evaluation metric.
- **Callbacks:** Early stopping, learning rate reduction on plateau, and model checkpointing were implemented to monitor the training process and save the best model.
- **Training Process:** The model was trained on the augmented dataset for 15 epochs with a batch size of 32, ensuring the training process was robust and avoided overfitting.

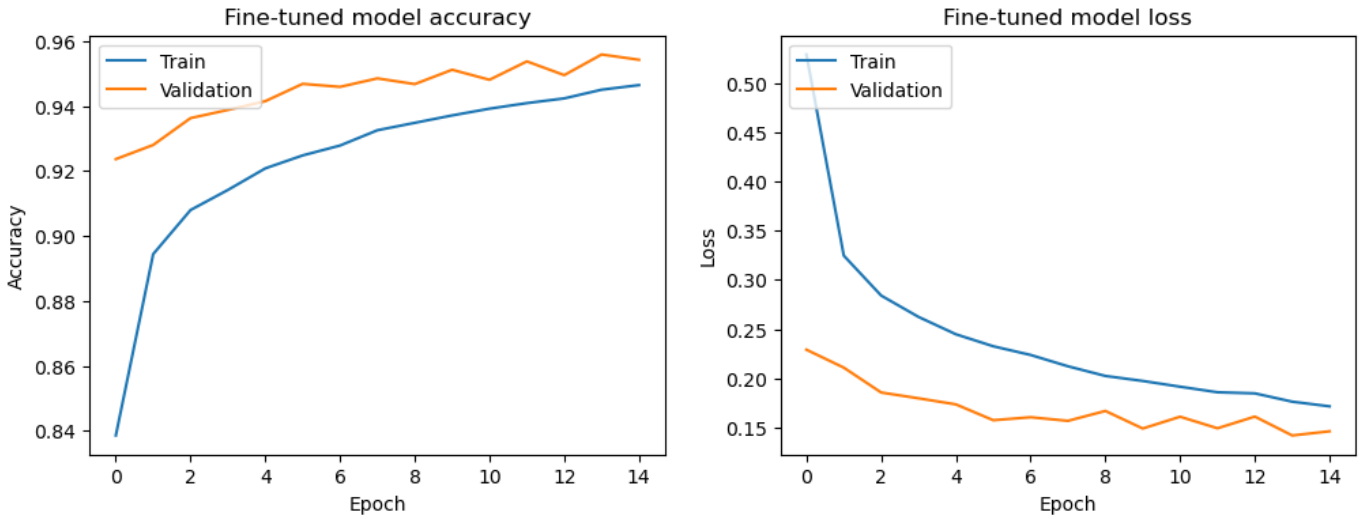
3.1.1.4 Evaluation:

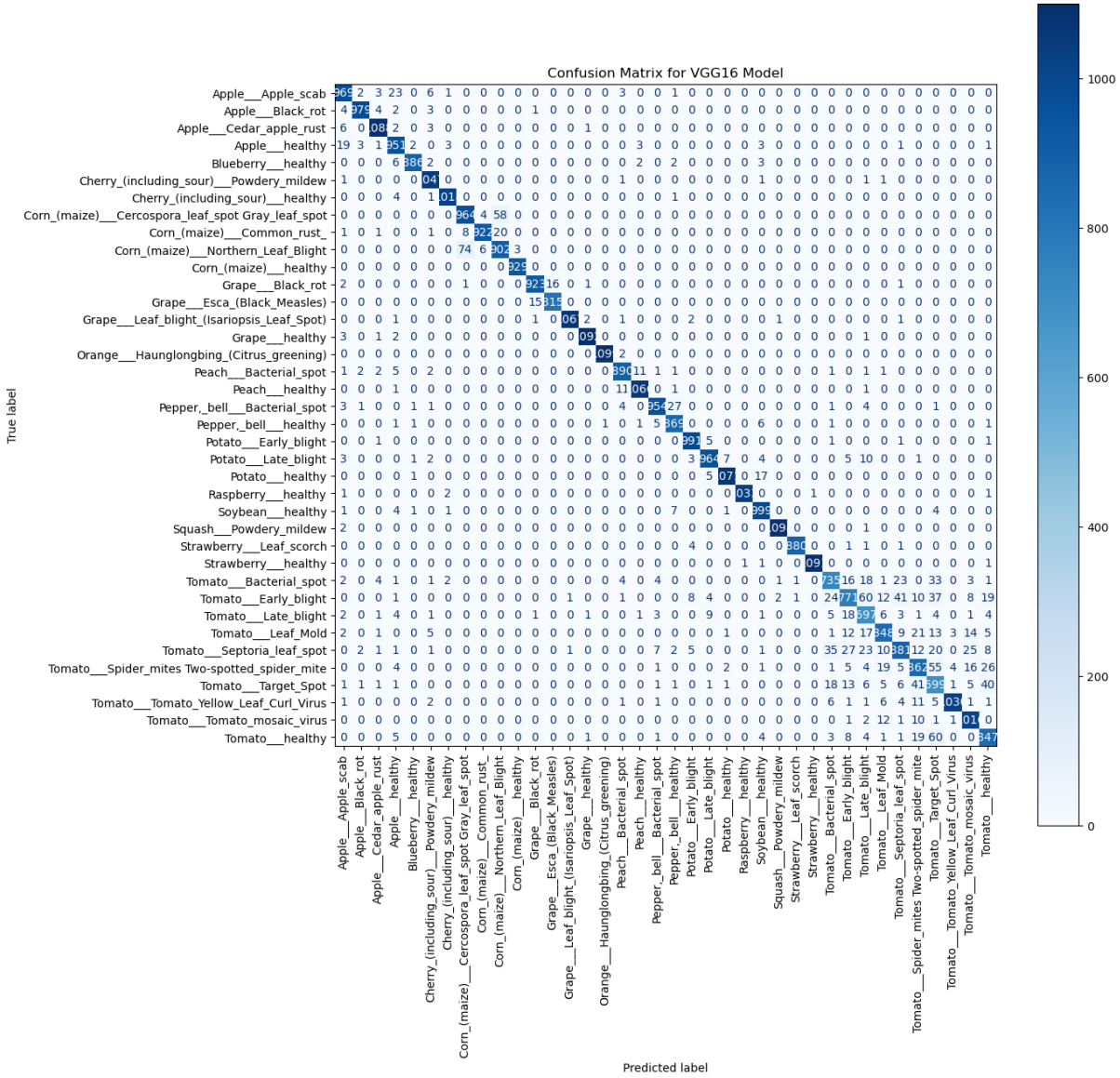
- **Validation:** The model's performance was validated on the validation set, and metrics such as accuracy, precision, recall, and F1-score were recorded.
- **Confusion Matrix:** A confusion matrix was plotted to visualize the types of misclassifications and identify specific classes where the model struggled.
- **Classification Report:** A detailed classification report was generated to provide insights into the model's performance across different classes.

4 Final model

After comprehensive testing and detailed analysis, VGG16 was chosen as the primary model for our advanced plant recognition system. This decision was influenced by VGG16's exceptional combination of high accuracy and computational efficiency, essential for real-time applications. Particularly compelling was its performance during the validation phase, where it achieved an impressive 95% accuracy, thereby establishing itself as a robust model for species identification and disease detection. Below is a short summary to why we chose this model:

1. High Accuracy: VGG16 achieved an impressive accuracy of 95% on our validation set, making it the most accurate model among those tested.
2. Robust Feature Extraction: The deep layers of VGG16 effectively captured complex features and patterns in the plant images, essential for distinguishing between different species and disease states.
3. Efficient Transfer Learning: By leveraging pre-trained weights, VGG16 provided a strong starting point, requiring less training time and computational resources compared to training a model from scratch.
4. Proven Performance: The model consistently performed well across different evaluation metrics, including precision, recall, and F1-score, ensuring reliable and balanced performance.





Moving forward, we will continue to utilize this model and delve deeper into its interpretability, exploring its potential in the subsequent phases of our project.