**+ Comparative study of Feature Extraction with Machine learning Models vs CNN for plant Leaf Disease Classification**

**AN ASSIGNMENT REPORT**

***Submitted by***

**MADHUMITHA K**

**(CH.SC.U4AIE23027)**

**MOPURI RISHITHA**

**(CH.SC.U4AIE23030)**

**REENA RAO R M**

**(CH.SC.U4AIE23044)**

**BACHELOR OF TECHNOLOGY IN ARTIFICIAL INTELLIGENCE AND ENGINEERING**

## **Under the guidance of**

**DR. K DEEPAK**

**Submitted to**

**AMRITA VISHWA VIDYAPEETHAM**

**AMRITA SCHOOL OF COMPUTING**

**CHENNAI – 601103**

**September -2025**



**BONAFIDE CERTIFICATE**

This is Certify that this project report entitled **“Comparative study of Feature Extraction with Machine learning Models vs CNN for plant Leaf Disease Classification”** is the Bonafide work of **MADHUMITA.K(CH.SC.U4AIE23027), RISHITHA (CH.SC.U4AIE23030), REENARAO.R.M(CH.SC.U4AIE23044)** who carried out the assessment work under my supervision.

**SIGNATURE**

**DR. K DEEPAK**

Department of CSE-cys.

Assistant Professor, Dept. Of CSE-CYS.

Amrita School of Computing

Chennai

**INTERNAL EXAMINER**

**ABSTRACT**

This work focuses on image classification using different feature extraction techniques and machine learning models on a plant disease dataset. Initially, we applied classical feature extraction descriptors such as Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), and Edge detection (with a Sobel filter) on the dataset after preprocessing it. Then, we tested them with five machine learning classifiers like Logistic Regression, Decision Tree, Random Forest, KNN, and SVM. To explore deep learning, we tested our dataset with two approaches: an end-to-and Convolutional Neural Network (CNN) which is trained directly on the dataset, and VGG16, a pretrained deep learning model which is used as a feature extractor with ML classifiers. We noticed that an end-to-end CNN is achieving higher accuracy when compared with all other models. Again, we tested with Resnet50 that loads pre-trained ImageNet weights and excludes the final classification layer, allowing the model to focus on plant leaf disease classification, and it had achieved a better accuracy approximately 90 % than CNN.

**Keywords:** VGG16, HOG, CNN, LBP, KNN, SVM, Logistic Regression, Decision Tree, Random Forest

**ACKNOWLEDGEMENT**

This work would have been possible without the contribution of many people. It gives us immense pleasure to express my profound gratitude to our honorable Chancellor Sri Mata Amritanandamayi Devi, for her blessings and for being a source of inspiration. I am indebted to extend my gratitude to our Sampoojya Swami Vinayamritananda Puri, Administrative Director, and Shri. I B Manikantan, Campus Director for facilitating us all the facilities and extended support to gain valuable education and learning experience.

I register my special thanks to Dr. V. Jayakumar, Principal, for the support given to me in the successful conduct of this project. I wish to express my sincere gratitude to Dr. K Deepak, Project Coordinator, Assistant professor at CSE- CYS for their inspiring guidance, personal involvement and constant encouragement during the entire course of this work.

I am grateful to Review Panel Members and the entire faculty of the Department of Artificial Intelligence, for their constructive criticisms and valuable suggestions which have been a rich source to improve the quality of this work.

**MADHUMITA K**

**(CH.SC.U4AIE23027)**

**MOPURI RISHITHA**

**(CH.SC.U4AIE23030)**

**REENARAO R M**

**(CH.SC.U4AIE23044)**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **TITLE** | **PAGE NO** |
| Fig 1 | HOG feature visualization | 6 |
| Fig 2 | LBP image and its pattern using histogram | 6 |
| Fig 3 | edge detection using Sobel filter | 7 |
| Fig 4 | performance of HOG + SVM on each class | 8 |
| Fig 5 | confusion matrix for HOG+SVM | 9 |
| Fig 6 | performance of LBP + Random Forest on each class | 10 |
| Fig 7 | confusion matrix for LBP + Random Forest | 11 |
| Fig 8 | performance of EDGE detection +SVM in all classes | 12 |
| Fig 9 | Confusion matrix of Edge detection + SVM | 12 |
| Fig 10 | performance of all ML models vs Deep learning model (CNN) | 13 |
| Fig 11 | Training accuracy and validation accuracy of Resnet50 | 14 |
| Fig 12 | Training and validation loss of Resnet50 | 14 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO** | **TITLE** | **PAGE NO** |
| Table 1 | classification report of HOG among all classifiers | 7 |
| Table 2 | classification report of LBP among all classifiers | 8 |
| Table 3 | classification report of Edge detection among all classifiers | 9 |
| Table 4 | classification report of VGG16 among all classifiers | 10 |

**LIST OF SYMBOLS AND ABBREVATIONS**

|  |  |
| --- | --- |
| **SVM** | Support Vector Machine |
| **HOG** | Histogram of Oriented Gradients |
| **LBP** | Local Binary Patterns |
| **KNN** | K- nearest neighbor |
| **CNN** | Convolutional Neural Network |
| **ML** | Machine learning |
| **DL** | Deep Learning |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER**  **NO.** | **TITLE** | **PAGE NO** |
|  | Abstract | iii |
|  | Acknowledgement | iv |
|  | List of Figures | v |
|  | List of Tables | vi |
|  | List of Symbols and Abbreviations | vii |
| 1 | INTRODUCTION | 1 |
| 2 | LITERATURE SURVEY | 2 |
| 3 | METHODOLOGY | 5 |
| 4 | EXPERIMENTATION | 6 |
| 5 | RESULT AND ANALYSIS | 9 |
| 6 | CONCLUSION | 17 |
| 7 | FUTURE SCOPE | 18 |
| 6 | REFERENCES | 19 |
|  | APPENDIX | 20 |

**CHAPTER 1**

**INTRODUCTION**

Plant diseases are a major problem for farmers worldwide. They decrease crop yields, lower food quality, and cause economic losses. Early detection is crucial for it. It allows farmers to take effective control measures. Traditionally, farmers have inspected leaves by hand to spot symptoms. This approach is slow, expensive, and sometimes subjective, depending on human observation. With the evolution of computer vision and machine learning, new computerized approaches to disease diagnosis through digital photographs of leaves have been developed. These methods accelerate the process, reduce expenses, and increase precision.

Machine learning algorithms can analyze photos of leaves to detect patterns associated with particular diseases automatically, enabling farmers to make more informed decisions. In this work, we tried to develop a plant leaf disease classification system through two primary methods. The first method is conventional machine learning, where we preprocessed images to grayscale and employed feature extraction techniques such as HOG (Histogram of Oriented Gradients), LBP (Local Binary Patterns), and Edge Detection. We extracted features in training with different classifiers such as Logistic Regression, KNN, Decision Trees, Random Forest, and SVM.

The second method is based on deep learning using Convolutional Neural Networks (CNNs). Unlike traditional methods, CNNs process data in an end-to-end manner, learning the optimal features from raw images directly and classified as healthy or not healthy. This makes CNNs better suited for image-based applications like plant disease diagnosis. The objective of this assignment is to compare the performance of these two methods and decide on which method is better in classifying plant leaf diseases.

**CHAPTER 2**

**LITERATURE REVIEW**

**2.1 Significance of feature extraction in computer vision:**

The foundation of computer vision is widely acknowledged to be feature extraction. In the absence of it, raw pixel intensities are too high dimensional, redundant, and noise-sensitive to be of immediate value in categorization. This point has been reaffirmed in numerous surveys: Khalid et al. divided feature extraction techniques into three categories: learning (CNN, transformers), statistical (textures, co-occurrence), and structural (edges, forms). Salau and Jain emphasized how each technique developed to overcome the shortcomings of its forerunners.

The low processing cost and interpretability of the initial hand-written descriptors were valued but subsequently broke down under varying lighting conditions, scales, and orientations. CNN auto-learns features but at the cost of increased computational demands and lower transparency.

Feature extraction methods show that the choice of features strongly influences the classification. Thus, the most robust model should balance discriminative capabilities and computational efficiency. This is why we are comparing traditional, deep learning and hybrid models.

**2.2 Traditional Feature Extraction methods:**

**2.2.1 Histogram of Oriented Gradients (HOG):**

Histogram of Oriented Gradients (HOG), introduced by Dalal and Triggs (2005), in their research on human detection, where they demonstrated its superiority over earlier descriptors such as PCA-SIFT and Haar-like wavelets. It is one of the most influential feature extraction methods. It encodes the local intensity gradients into orientation histograms. The concept is to represent an image as a distribution of gradient orientation, calculated over local regions. It offers good results under lighting changes and minor deformations. It gained prominence in pedestrian detection, face recognition, traffic sign detection and industrial inspection.

In agriculture, leaf contour and lesion boundaries are captured by HOG. It is highly effective for shape relative tasks, but the results degrade under complex illumination or scale variations. To address these limitations, hybrid approaches with HOG are recommended for better results.

**Principal of HOG:**

HOG works with filters such as Sobel operators to compute the image gradients. The orientation and magnitude are quantized into fixed bins, commonly 9. The histogram of gradient directions is found by dividing the image into cells. Histograms are normalized at overlapping regions to ensure invariance to illumination, called blocks. These blocks are concatenated into a single vector, which can be used with classifiers such as SVM.

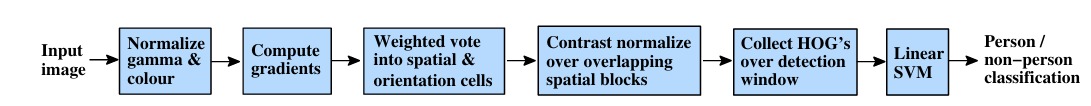


Fig an overview of feature extraction and object detection chain from their work.

Dalal and Triggs (2005) showed that scaled gradients along with normalization were required for robust detection. While their experiment was about human detection, it can also be applied to objects such as leaves. Thus, we chose HOG are one of our traditional feature extraction methods.

**Application of HOG in our Plant Dataset:**

HOG descriptors detect edges, venation and patterns caused by disease lesions. Smooth contours, uniform venation are found in healthy leaves while irregular textures, lesion boundaries, localized distortions etc. are found on diseased leaves. Even in gray scale representations HOG can be used effectively to distinguish between healthy and diseased leaves. It provides a baseline to compare more advanced techniques such as LBP, CNN and Sobel edge detection. We chose this method for its good reception to illumination changes and minor deformations, and its computational efficiency.

**2.2.2 Local Binary Patterns (LBP):**

Ojala et al. (2002) introduced LBP as a computationally effective descriptor to model textures. It represents local structures in the image by thresholding the neighboring intensity pixels relative to the center pixel with the result being binary patterns that capture well the surface micro textures. Commonly used in biometrics, material inspection, and medical imaging. In agriculture, it has been used to identify textural variations in leaf surfaces resulting from disease or nutritional deficiencies. The advantages are that it is simple and compatible, but the disadvantage is that it is very sensitive to noise and rotation changes, which restricts its use in unconstrained situations.

**2.2.3 Sobel edge detection:**

It is one of the earliest techniques for edge detection, which estimates intensities along horizontal and vertical directions. It is simple and efficient particularly on the resource limited systems. It has been used in medical image segmentation, industrial quality control and structural analysis. For my dataset, it is useful to obtain first order information about the lead venation or lesions. They are prone to noise and background interference.

CNN-Based Feature Extraction:

In computer vision, the introduction of convolutional neural networks, or CNNs, represented a paradigm shift. CNNs learn hierarchical representations of features automatically, from complex structures and semantic patterns in deeper layers to edges and textures in shadow layers, as opposed to handcrafted features.

Initially demonstrated to perform well in broad tasks such as digit recognition (LeCun et al., 1998), CNNs became widespread after AlexNet's landmark performance on ImageNet (Krizhevsky et al., 2012). Mohanty et al. (2016) demonstrated that CNNs learned on the PlantVillage dataset outperformed handcrafted methods in plant disease in 26 diseases and 14 crops.

Subsequent work made the CNN more interpretable and effective. Brahmi et al(2019) incorporated attention mechanisms to produce saliency maps, boosting the confidence of CNN predictions. Ahmed et al.(2021) concentrated on lightweight model like MobileNetV2, proving that high accuracy was achievable with compact model sizes for mobile deployment. CNN has a lot of advantages like better accuracy, flexibility, and the capacity to generalize across various conditions, while the sole trade-offs are computational complexity and interpretability.

**2.3 Hybrid Approaches: Integrating Handcrafts and Deep features:**

Studies have explored hybrid models that combine handcrafted features with deep representations. Research on rice leaf disease classification (2024) showed that HOG descriptors with Efficient Net increased accuracy from \~92%. The research further said that the hybrid model localized disease regions more effectively.

These studies show that handcrafted methods were no longer sufficient on their own, but can still compliment deep network. Efficient in fields with limited training data or high variation where using a single method would be sufficient.

Emerging Architectures: Convolution Transformer Hybrids

In recent years, there has been a shift beyond traditional convolutional neural networks (CNNs) towards more advanced models like vision transformers (ViTs) and hybrid convolution-transformer architectures. Transformers, which were originally created for natural language processing tasks, are particularly good at capturing long-range relationships in data. However, they don’t naturally include some of the useful characteristics of CNNs, such as a focus on local patterns and the ability to recognize features regardless of their position (translation invariance). To bridge this gap, hybrid models like CEFormer have emerged, aiming to blend the strengths of both methods: using convolutions to capture local features and attention mechanisms to understand broader, global patterns.

**2.4 Research gap:**

Despite significant progress, there are still many areas where research is lacking. In real-world field settings, where factors like changing lighting, object occlusion, and complex backgrounds can impact performance, many models that perform well on controlled datasets like Plant Village often perform poorly. Furthermore, although there are interpretability tools for CNNs, most of them are still qualitative in nature, and more quantitative techniques are obviously required to validate the subject matter of these models (e.g., saliency maps).

Furthermore, despite the promising outcomes of hybrid feature extraction pipelines, most of the research has focused on HOG (Histogram of Orientated Gradients), with less attention paid to other descriptors such as Sobel filters or LBP (Local Binary Patterns).

This study takes a more comprehensive approach to address these issues. It assesses the performance of HOG, LBP, and Sobel features, as well as embeddings from VGG16 and designed CNN models. The performance is assessed using various classifiers and an extensive array of metrics such as accuracy, precision, recall, F1-score, ROC-AUC, and Cohen's kappa. The models are further subjected to robustness in cases such as noise and rotation of images. Also, by making recent transformer and hybrid models part of the discussion, this paper not only fills existing loopholes but also is in line with the existing trends in the field offering valuable insight and broader perspective for future work.

**CHAPTER 3**

**METHODOLOGY**

1. Dataset Preparation: we had selected a plant village dataset from Kaggle. It consists of approximately 16,000 to 20,000 labeled images of healthy and diseased leaves from crop species such as tomato, potato, and bell pepper. Each image is categorized into different classes such as early bright, late bright and healthy etc. All these images are of good quality, lightning and with plain backgrounds that help to focus more on the leaf and easily minimizes the background noise. This makes the dataset is suitable for training and testing the machine learning models to recognize plant diseases. One challenge with this dataset is class imbalance as some diseases have significantly more samples than others. So, it requires preprocessing such as resizing, grayscale conversion, or feature extraction (HOG, LBP, VGG16, etc.)
2. We preprocessed the data using greyscale conversion. For feature extraction, we had chose classical ML models like HOG, LBP and Edges (Sobel) where HOG captures gradient/texture features, LBP encodes local texture patterns and Edge detection extracts leaf boundary and vein. We trained all these three models with the machine learning models classifiers (Log Regression, KNN, Decision Tree, Random Forest, SVM) and calculated evaluation metrics like accuracy, precision, recall, f1 score, Cohen kappa score, ROC\_AUC score and confusion matrix for all combinations.
3. Similarly, we had used an end-to-end CNN which is a deep learning model that gets images as input and gives the output without any steps like feature extraction and classification. In classical ML models, we will convert images to greyscale, then extract features using methods like HOG, LBP or edges, and last, we train a classifier. CNNs are more accurate for complex tasks like plant disease detection as the network automatically learns the features (like texture, color spots, or disease patterns) through its convolution layers and classifies them in the final layers.

**CHAPTER 4**

**EXPERIMENTATION**

1. We worked on a plant leaf disease dataset that contains images of healthy and as well as diseased leaves.
2. All images are resized to a fixed size of 64x64 pixels and converted to grayscale.
3. Normalization was applied to bring all pixel values in between 0 and 1 so that model will be more stable during training
4. After preprocessing, we extracted features from the images using three feature extraction techniques.

1) Hog (Histogram of Oriented Gradients): it captures the gradient and shape information of the leaves.

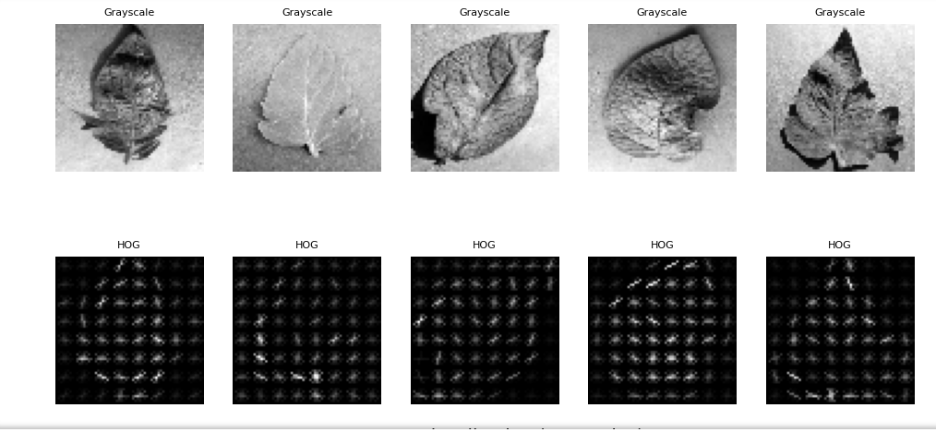


Fig1: HOG feature visualities

2) LBP (Local Binary Pattern): it was used to capture the spots in the leaf

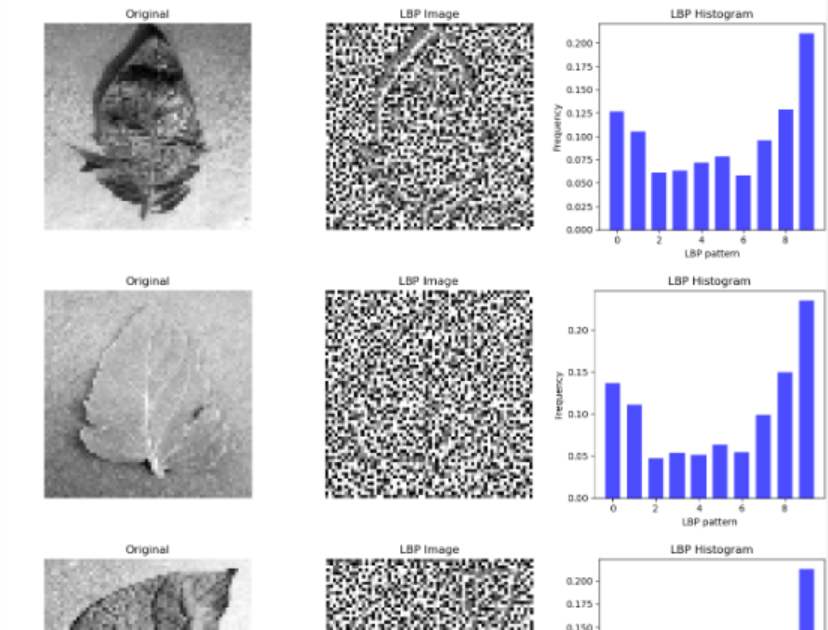


Fig2: LBP image and its pattern using histogram

3) Edge Detection (Sobel Filter): it was used to focus on the outline and boundary details of the leaf.

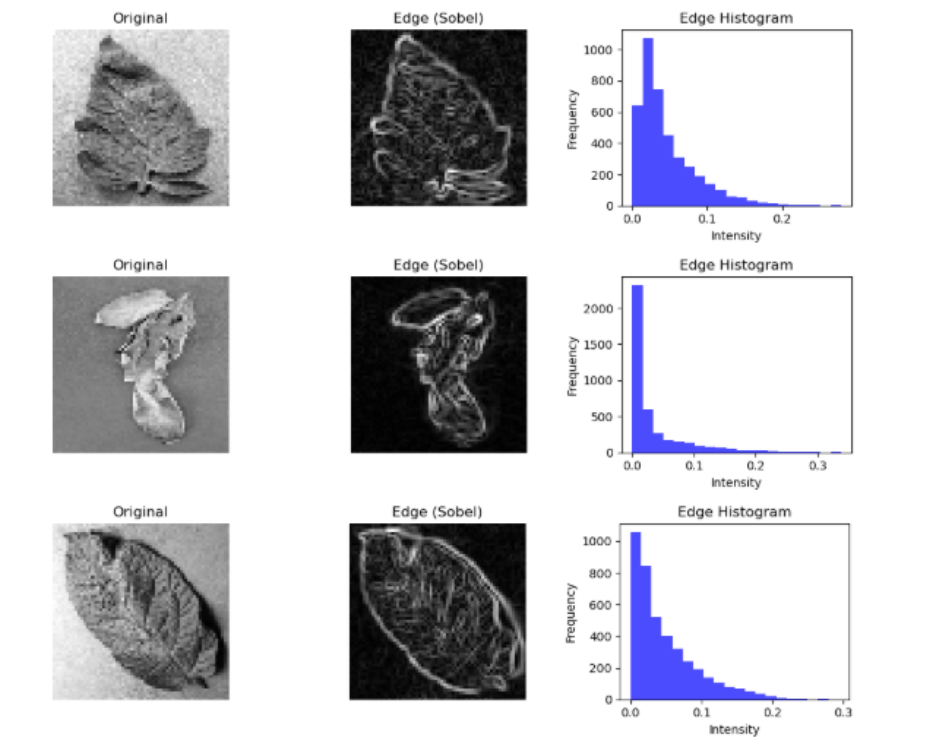


Fig3: edge detection using Sobel filter

1. These extracted features served as inputs to the machine learning models.
2. After extracting features, the dataset was divided into training and testing sets (80:20 split) and we trained different machine learning classifiers including Logistic Regression, Support Vector Machine, Decision Tree and k-Nearest Neighbours. Each classifier's performance was evaluated based on its performance in identifying different plant leaf diseases from the test set.
3. We used accuracy as the main metric. Confusion matrices were also analyzed to check which diseases were often misclassified.
4. Similarly, we implemented an end-to-end Convolutional Neural Network (CNN) and VGG-16. Unlike these feature extraction techniques, CNN automatically learns features from the raw images through convolutional and pooling layers. The fully connected layers of CNN performed the classification task.
5. By comparing all results (traditional mL vs DL models), deep learning model CNN gains more accuracy.

**CHAPTER 5**

**RESULTS AND ANALYSIS**

The experiments were done using three traditional feature extraction methods (HOG, LBP, Edge Detection) combined with the ML classifiers (Logistic Regression, KNN, Decision Tree, Random Forest, and SVM). The performance of each combination (feature extraction method+ ML classifier) was evaluated using performance metrics such as accuracy, precision, recall, F1-sore, Cohen’s kappa score, and ROC- AUC. Confusion matrices were also plotted to analyze class-wise performance

**HOG:**

Table1 : classification report of HOG among all classifiers

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **classifier** | **accuracy** | **precision** | **recall** | **f1** | **kappa** | **ROC-AUC** |
| LogReg | 0.490241 | 0.487898 | 0.4858 | 0.4837 | 0.4529 | 0.8951 |
| KNN | 0.4420 | 0.50499 | 0.4485 | 0.4191 | 0.4030 | 0.8068 |
| DecisionTree | 0.218140 | 0.2151 | 0.2151 | 0.2141 | 0.1613 | 0.5791 |
| RandomForest | 0.440873 | 0.4054 | 0.4277 | 0.4043 | 0.3996 | 0.8584 |
| SVM | 0.5912 | 0.5829 | 0.5846 | 0.5779 | 0.5613 | 0.9339 |

Among the classifiers, SVM achieved the best performance, while the decision tree is comparatively weaker. HOG features were effective in shape and gradient information of diseased leaves

SVM performs better because it handles high dimensional vector spaces like HOG while decision tree split data based on threshold of a single feature. It struggles with multiple correlated structures. SVM maximizes margin between classes making them robust for minute differences in patterns.

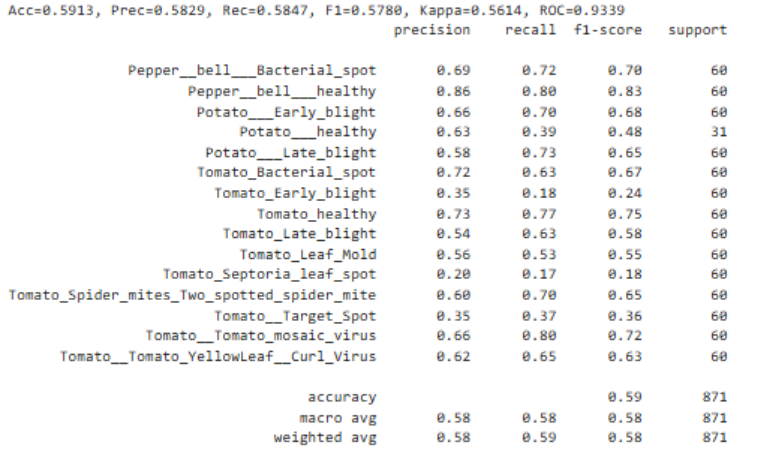


Fig 4: performace of HOG+svm on each class

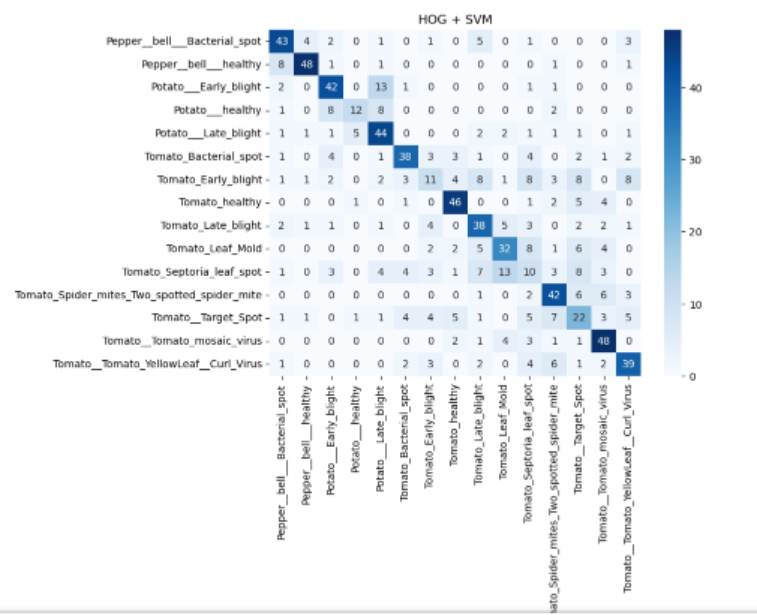


Fig 5: confusion matrix for HOG+SVM

**LBP**

Table 2: classification report of LBP among all classifiers

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **classifier** | **accuracy** | **precision** | **recall** | **f1** | **kappa** | **ROC-AUC** |
| LogReg | 0.1618 | 0.1729 | 0.1566 | 0.095 | 0.0998 | 0.7394 |
| KNN | 0.2342 | 0.2428 | 0.2297 | 0.227 | 0.1788 | 0.6709 |
| DecisionTree | 0.1997 | 0.2050 | 0.1995 | 0.2014 | 0.1413 | 0.5711 |
| RandomForest | 0.2870 | 0.2895 | 0.2840 | 0.2815 | 0.2349 | 0.7940 |
| SVM | 0.2606 | 0.2760 | 0.2522 | 0.2324 | 0.2059 | 0.8146 |

LBP is good at capturing small texture patterns, useful when leaves have spots. Random forest showed a better balance between accuracy, f1 score and kappa. LBP by itself is noisy so decision trees overfit. Random forest combines many trees and averages out the noise from individual trees, making it more stable.

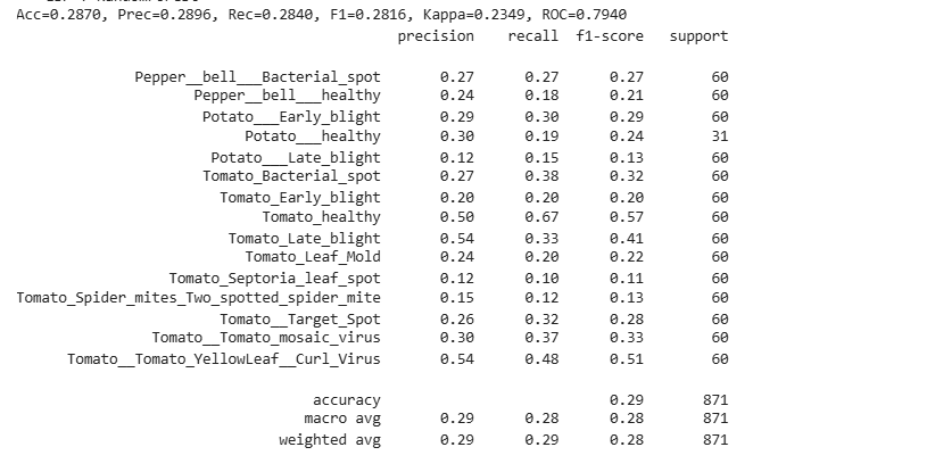


Fig 6: performance of LBP + Random Forest on each class

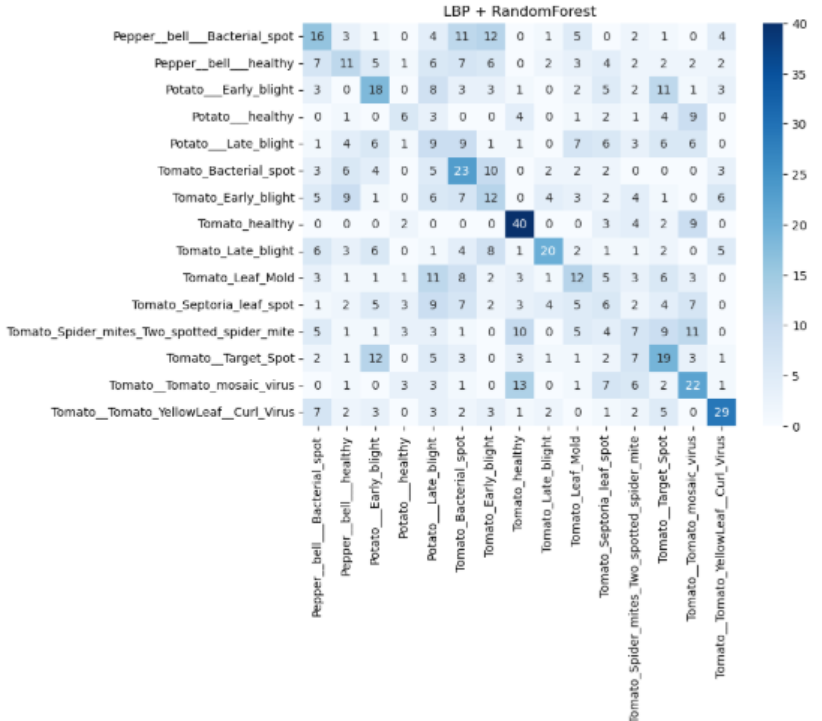


Fig 7: confusion matrix for LBP+Random Forest

**Edge:**

Table: classification report of Edge detection among all classifiers

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| classifier | accuracy | precision | recall | f1 | kappa | ROC-AUC |
| LogReg | 0.3960 | 0.3744 | 0.3885 | 0.3786 | 0.3521 | 0.8378 |
| KNN | 0.2032 | 0.2525 | 0.1966 | 0.1713 | 0.1448 | 0.6459 |
| DecisionTree | 0.1745 | 0.1727 | 0.1720 | 0.1717 | 0.1145 | 0.5564 |
| RandomForest | 0.4167 | 0.3750 | 0.4033 | 0.3766 | 0.3736 | 0.8247 |
| SVM | 0.4890 | 0.4954 | 0.4764 | 0.4598 | 0.4514 | 0.9009 |

Edge–based features are less effective compared to HOG and LBP as they only captured boundary information and ignores the detail textures. Accuarcy was lower , though SVM outperformed other classifiers.

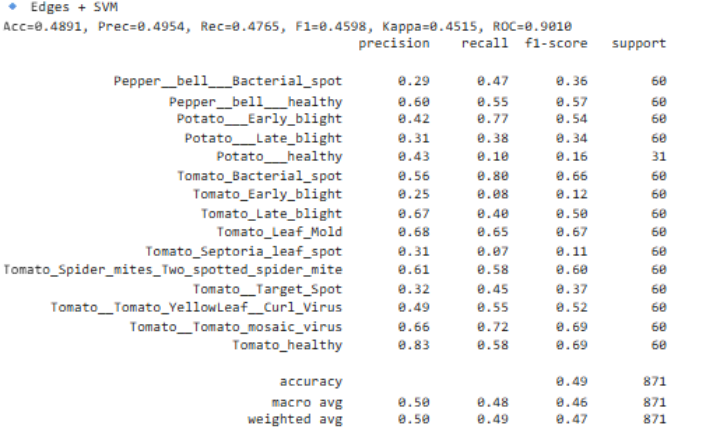


Fig 8: performance of EDGE detection +SVM in all classes

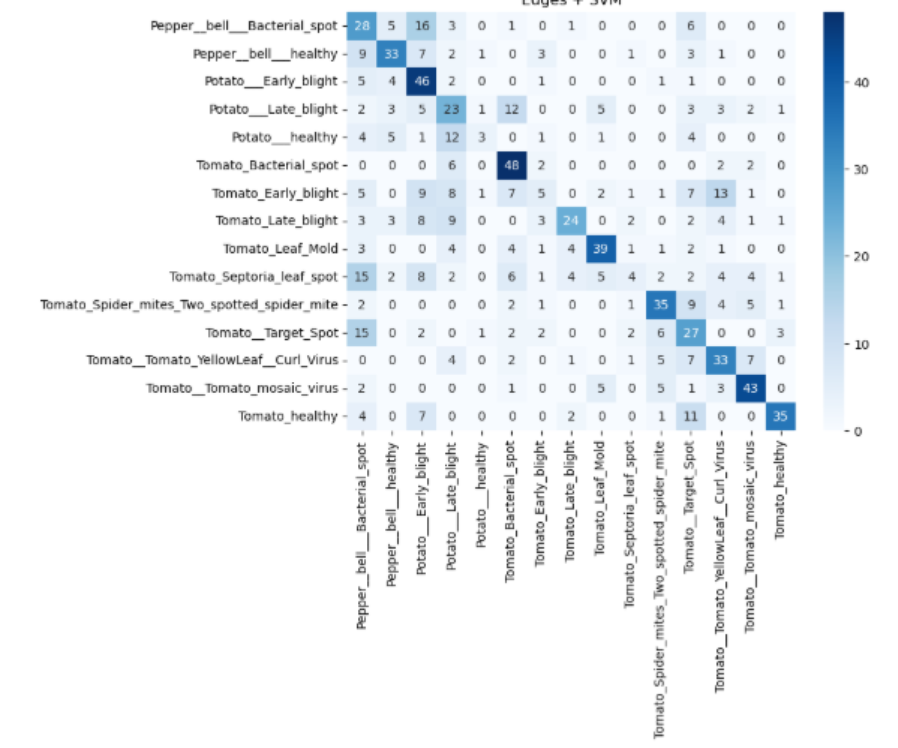


Fig 9: Confusion matrix of Edge detection+SVM

VGG16:

Table 4: classification report of VGG16 among all classifiers

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| classifier | accuracy | precision | recall | f1 | kappa | ROC-AUC |
| LogReg | 0.51 | 0.5205 | 0.5308 | 0.5077 | 0.4728 | 0.9321 |
| KNN | 0.4500 | 0.5890 | 0.4785 | 0.4502 | 0.4109 | 0.8167 |
| DecisionTree | 0.2300 | 0.2404 | 0.2131 | 0.2149 | 0.1735 | 0.5791 |
| RandomForest | 0.4700 | 0.5054 | 0.5265 | 0.4605 | 0.4323 | 0.8943 |
| SVM | 0.4500 | 0.6208 | 0.4775 | 0.4391 | 0.4097 | 0.9277 |

VGG16 learns features automatically and the classifiers added on the top will be work with more stronger input features. It is better than the previously done works (HOG/LBP/edges+ classifier) as VGG16 features are more powerful but it is slightly less powerful than CNN. VGG16 features were prertrained on ImageNet and it may not fully observe or capture the specific patterns in our datset. However, VGG16 still provide a better result of accuracy approximately 0.51% when it is used with a ML classifier.

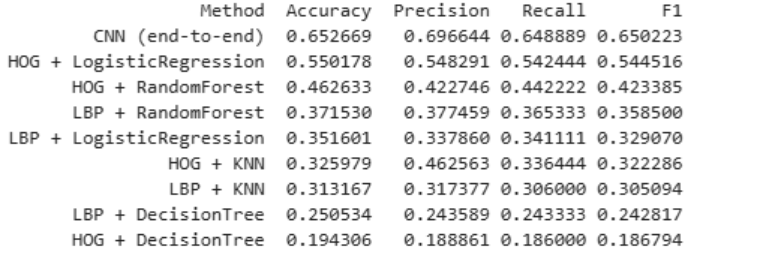


Fig 10 : performance of all ML models vs Deep learning model (CNN)

Out of all these, CNN performed well as compared with classical ML models as it automatically learns useful features at multiple levels

The traditional ML models demonstrated that feature extraction techniques like HOG and LBP are effective but limited by manual design. Their performance varied depending on which classifier was used, with Random Forest and SVM consistently being the strongest.

This shows that deep learning can capture and understand complex patterns in plant disease datasets better than the older methods that use hand made features.

In confusion matrices we can see that some disease classes were not easy to distinguish. Misclassifications were occurred between these type of similar diseases. CNN can handle such overlaps better due to its hierarchical feature learning

When tested with noisy or rotated images, the classical ml models showed a less accuaracy, while CNN was robust. This shows the adaptability of CNN and it it can handle real world applications better.

**ResNet50:**

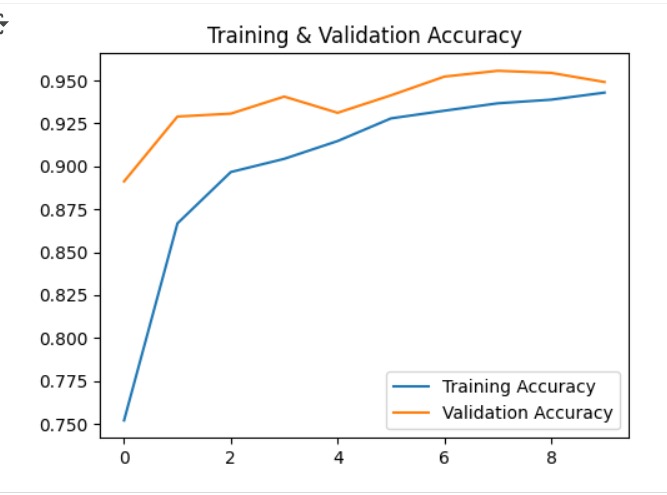


Fig 11: Traning accuary and Validation accuracy of ResNet50

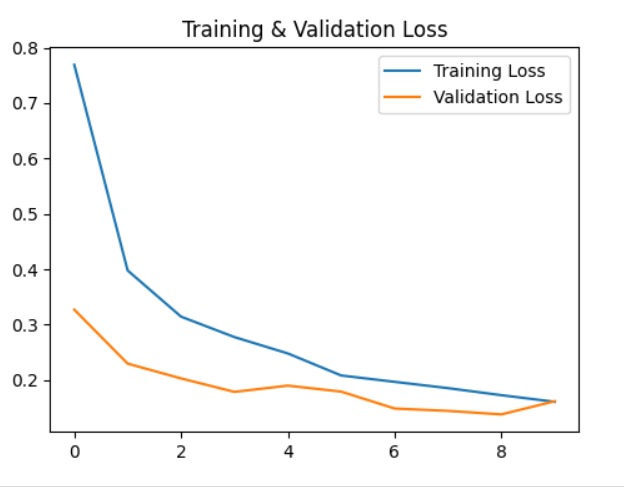


Fig 12: Training loss and Validation loss of ResNet50

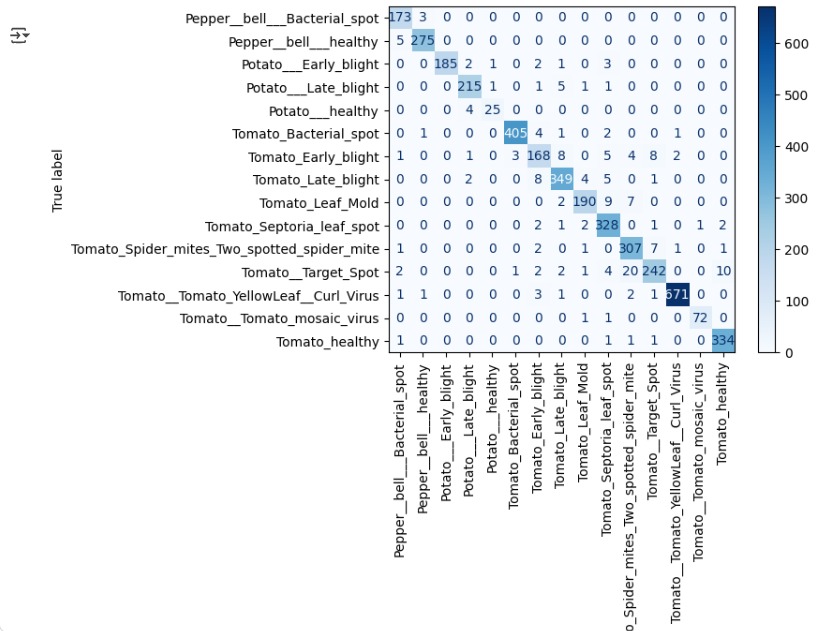


Fig 12: Confusion Matrix of ResNet16

In this work, ResNet50 was employed as a deep feature extractor by loading pre-trained ImageNet weights while excluding the final classification layer, allowing the model to focus on plant leaf disease classification. The base layers were frozen, and only a newly added classification head was trained, consisting of a GlobalAveragePooling2D layer to condense the 7×7×2048 feature maps into a compact 2048-dimensional vector, followed by a Dense layer with 256 units and ReLU activation with Dropout to reduce overfitting, and finally a softmax layer corresponding to the 15 disease categories. To optimize training, the input images were resized to 160×160 instead of the default 224×224, a batch size of 32 was used, and Early Stopping was applied on validation loss with a patience of three, using the Adam optimizer and sparse categorical crossentropy since labels were integer-encoded. Training was conducted on the PlantVillage dataset with an 80/20 split for training and validation, and despite being slower than traditional methods, the model converged within 5–8 epochs. The model summary indicated approximately 24 million total parameters, but only 528k were trainable due to freezing of the ResNet backbone. Experimentally, ResNet50 achieved significantly higher accuracy (≈90%+) compared to traditional handcrafted methods such as HOG, LBP, and Sobel (≈65–75%), and demonstrated greater robustness under noisy and rotated inputs. Confusion matrix analysis further showed that ResNet handled inter-class similarities between visually similar diseases more effectively than traditional approaches. Overall, the ResNet50 pipeline highlighted the strength of deep learning-based feature extraction, offering superior accuracy, generalization, and robustness, thereby reinforcing the objective of this study in evaluating the impact of feature extraction techniques on classification performance.

**CHAPTER 6**

**CONCLUSION**

In this work, we aimed in classifying plant leaf diseases using traditional machine learning methods with handcrafted feature extraction and deep learning with a Convolutional Neural Network (CNN). We used techniques like HOG, LBP, and edge detection to extract important features from the images, tested them with classifiers such as Logistic Regression, SVM, Decision Trees, Random Forests, and k-NN. These methods helped us to understand how different features impact classification accuracy. However, we found some limitations since handcrafted features can't always capture the complex patterns of plant diseases.

In contrast, the CNN model learns features directly from raw images and performed better by automatically detecting textures, shapes, and patterns. This showed the advantages of deep learning in image-based tasks compared to traditional methods.

Overall, our findings found out that traditional feature extraction techniques are easy to implement and not that much accurate, whereas Resnet50 offer a stronger and more scalable solution for detecting plant diseases. This study highlights the value of combining both approaches for better understanding, but it also shows that deep learning is becoming the preferred method for tackling real-world agricultural challenges.

**CHAPTER 7**

**FUTURE SCOPE**

Although this work compared classical machine learning algorithms with CNNs for classifying plant leaf diseases, we need to make more improvements and run additional experiments. Future works could include datasets with more images that has different lighting conditions, backgrounds, and plant varieties to improve generalization. Data augmentation methods like rotation, zooming, and flipping can be used to increase its robustness.

In deep learning, we can try other architectures like ResNet, VGG, or EfficientNet to see if they provide better accuracy than a CNN. Using pre-trained models for transfer learning can help us achieve better performance with less training data.

Integrating the trained model into a mobile or web application would let farmers and farm workers to use it for real-time disease identification. Including explainable AI techniques will help to provide insights into the predictions by enabling users to understand why a specific disease was identified.

In summary, this project lays as a foundation. Future enchantments can improve system's accuracy, robustness, and practicality for real-world agricultural applications.

**REFERENCES**

[1] [Ahmed, M., Khan, A. M., & Raza, K. (2021). Lightweight convolutional neural network for tomato leaf disease classification. Computers and Electronics in Agriculture, 178, 105735.](https://doi.org/10.1016/j.compag.2020.105735](https://doi.org/10.1016/j.compag.2020.105735)

[2] [Barbedo, J. G. A. (2013). Digital image processing techniques for detecting, quantifying and classifying plant diseases. SpringerPlus, 2(660), 1–12.](https://doi.org/10.1186/2193-1801-2-660](https://doi.org/10.1186/2193-1801-2-660)

[3][Brahimi, M., Boukhalfa, K., & Moussaoui, A. (2019). Deep learning for tomato diseases: Classification and symptoms visualization. Applied Artificial Intelligence, 31(4), 299–315.](https://doi.org/10.1080/08839514.2017.1315516](https://doi.org/10.1080/08839514.2017.1315516)

[4][Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for human detection. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 1, 886–893.](https://doi.org/10.1109/CVPR.2005.177)

[5][He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778.](https://doi.org/10.1109/CVPR.2016.90)

[6][Hughes, D. P., & Salathé, M. (2015). An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv preprint arXiv:1511.08060.](https://arxiv.org/abs/1511.08060](https://arxiv.org/abs/1511.08060)

[7][Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems (NIPS), 25, 1097–1105.](https://doi.org/10.1145/3065386](https://doi.org/10.1145/3065386)

[8][LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.](https://doi.org/10.1109/5.726791](https://doi.org/10.1109/5.726791)

[9][Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. Frontiers in Plant Science, 7(1419), 1–10.](https://doi.org/10.3389/fpls.2016.01419)

[10][Ojala, T., Pietikäinen, M., & Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7), 971–987.](https://doi.org/10.1109/TPAMI.2002.1017623)

[11][Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. Computers and Electronics in Agriculture, 161, 272–279.](https://doi.org/10.1016/j.compag.2018.03.032)

[12][Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems (NeurIPS), 30.](https://arxiv.org/abs/1706.03762](https://arxiv.org/abs/1706.03762)

**APPENDIX**

**Dataset link:** <https://www.kaggle.com/datasets/emmarex/plantdisease/data>

**Code link:** <https://github.com/rishitha013/dl_assignment>