POTATO DISEASE CLASSIFICATION

MAJOR PROJECT REPORT

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

by

Paarth Bhasin	Rishabh Bhatnagar	Rishav Sasmal
05111503120	05811503120	05911503120

Guided by

Ms. Kajal Kaul
(Assistant Professor)

DEPARTMENT OF INFORMATION TECHNOLOGY
BHARATI VIDYAPEETH'S COLLEGE OF ENGINEERING
(AFFILIATED TO GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY, DELHI)
NEW DELHI – 110063

MAY.2024

CANDIDATE’S DECLARATION

It is hereby certified that the work which is being presented in the B. Tech Major project Report entitled **"Potato Disease Classification** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** and submitted in the **Department of Information Technology** of **BHARATI VIDYAPEETH’S COLLEGE OF ENGINEERING, New Delhi (Affiliated to Guru Gobind Singh Indraprastha University, Delhi)** is an authentic record of our own work carried out during a period from **February 2024 to May 2024** under the guidance of **Ms. Kajal Kaul, Assistant Professor.**

The matter presented in the B. Tech Major Project Report has not been submitted by me for the award of any other degree of this or any other Institute.

Paarth Bhasin	**Rishabh Bhatnagar**	**Rishav Sasmal**
05111503120	**05811503120**	**05911503120**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge. He/She/They are permitted to appear in the External Major Project Examination

Ms. Kajal Kaul
(Assistant Professor)

Prof. (Dr.) Prakhar Priyadarshi
(HoD, IT)

ABSTRACT

Potato Disease Classification using Machine Learning is an innovative approach aimed at addressing the challenges faced by potato farmers in identifying and managing diseases affecting their crops. Leveraging a diverse dataset sourced from PlantVillage, this project focuses on training and validating machine learning models to classify potato plant images into distinct disease categories, including early blight, late blight, and healthy specimens. The dataset comprises 1000 images each for early blight and late blight, along with 152 images representing healthy potato plants, all standardized to a resolution of 256x256 pixels.

The project employs a series of data preprocessing techniques, including resizing, rescaling, data augmentation, normalization, batching, and shuffling, to prepare the dataset for effective model training. These preprocessing steps ensure consistency, quality, and compatibility of the dataset with machine learning algorithms, facilitating robust model development.

A convolutional neural network (CNN) architecture is designed and implemented to extract meaningful features from input potato plant images and classify them into disease categories. The model architecture comprises convolutional layers, activation functions, pooling layers, flattening layers, fully connected layers, and an output layer with softmax activation.

The trained model is evaluated using metrics such as accuracy, precision, recall, and F1-score to assess its performance on unseen data. Confusion matrices and classification reports provide insights into the model's ability to accurately classify potato plant images into different disease categories.

The project concludes with the development of a user-friendly web application interface, allowing farmers to upload potato plant images and receive instant disease classification results. By empowering farmers with timely and accurate information, Potato Disease Classification using Machine Learning aims to enhance crop health, optimize agricultural practices, and contribute to global food security.

ACKNOWLEDGEMENT

We express our deep gratitude to **Ms. Kajal Kaul**, Assistant Professor, Department of Information Technology for his valuable guidance and suggestion throughout my project work. We are thankful to **Ms. Neha Gupta,** Class Advisor, for their valuable guidance.

We would like to extend my sincere thanks to **Head of the Department, Prof. Prakhar Priyadarshi** for his time-to-time suggestions to complete my project work. I am also thankful to **Prof. Dharmender Saini, Principal** for providing me the facilities to carry out my project work.

Paarth Bhasin	**Rishabh Bhatnagar**	**Rishav Sasmal**
05111503120	**05811503120**	**05911503120**

TABLE OF CONTENTS

TABLE OF FIGURE

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION

Potato Disease Classification using Machine Learning addresses the challenges farmers face in identifying and managing diseases affecting potato plants. Potatoes are a vital food crop globally, but diseases like early blight, late blight, and fungal infections pose significant threats to their yield. Traditional identification methods rely on visual inspection, which is time-consuming, subjective, and often inaccurate.

This project aims to develop a machine learning model capable of accurately classifying potato plant images into disease categories. By leveraging image processing and deep learning techniques, we seek to automate disease detection and aid farmers in making informed decisions regarding disease control.

Utilizing a dataset of potato plant images labeled with disease categories, we employ convolutional neural networks (CNNs) to extract features and train a classification model. Through hyperparameter optimization, model fine-tuning, and performance evaluation, we aim to develop a reliable classification system.

The significance of this project lies in its potential to revolutionize potato disease diagnosis and management in agriculture. Providing farmers with an efficient tool for disease classification can reduce crop losses, improve productivity, and promote sustainable farming practices. Moreover, insights gained can be applied to other crops, advancing precision agriculture and global food security.

1.1.1 DEEP LEARNING

Deep learning is a branch of artificial intelligence (AI) that enables computers to “learn” from data like the human brain. Deep learning models are comprised multiple layers of interconnected nodes, processing information in parallel to learn complex patterns and relationships. Deep learning models can be trained on vast datasets to excel in tasks like image recognition, natural language processing, and speech recognition. This makes them valuable tools for a wide range of applications, including self-driving cars, medical diagnosis, and virtual assistants.

Some key features of deep learning include:

- Multi-layered architecture: Deep learning models are made up of several layers of interconnected nodes. These layers allow the model to learn increasingly complex features from the data.

- Non-linear activation functions: Deep learning models use non-linear activation functions, such as the rectified linear unit (ReLU), which allow them to learn complex relationships between features.

- Supervised learning: Deep learning models are typically trained using supervised learning, which means that they are given labeled data. This data consists of input features and desired outputs, which the model learns to map between.

- Image recognition: Deep learning models can be used to recognize objects in images, with applications in areas such as self-driving cars, medical diagnosis, and security.

1.1.2 Python:

Python's simplicity and readability make it the perfect language for implementing complex machine learning algorithms and handling diverse datasets. Throughout the project, Python facilitates various tasks, including data preprocessing, model development, evaluation, and deployment. Its extensive library ecosystem provides specialized tools like NumPy, Matplotlib, and TensorFlow, which are instrumental in achieving accurate and efficient classification of potato diseases.

1.1.3 Flask:

To create an interactive and accessible platform for users to utilize my potato disease classification model, I turned to Flask. Flask's minimalist architecture and flexibility allow me to design a lightweight web application with ease. Through Flask, I define endpoints to handle image uploads, seamlessly integrate backend logic for processing images using the trained model, and dynamically render classification results to the user interface. Flask empowers me to tailor the application's features and interface to suit the specific needs and preferences of my target audience.

1.1.4 TensorFlow:

TensorFlow serves as the backbone of my machine learning pipeline, enabling the creation, and training of convolutional neural network (CNN) models for image classification tasks. Leveraging TensorFlow's high-level APIs, I design intricate CNN architectures comprising convolutional, pooling, and dense layers to extract and classify features from potato plant images. TensorFlow's computational graph abstraction ensures efficient execution of complex neural networks on various hardware platforms, optimizing performance and scalability during both training and inference phases.

1.1.5 NumPy:

NumPy plays a pivotal role in data preprocessing and manipulation, providing efficient data structures and functions for working with multidimensional arrays. Throughout the project, NumPy enables me to load, preprocess, and augment image datasets, preparing them for input into TensorFlow models. Its array operations facilitate batch processing and parallelization, enhancing the efficiency and scalability of image data handling. Additionally, NumPy's mathematical functions and statistical tools empower me to extract meaningful features from image data, improving the accuracy and robustness of my classification models.

1.1.6 Matplotlib:

Matplotlib enriches my project with insightful visualizations that aid in data exploration, model evaluation, and result interpretation. Leveraging Matplotlib's plotting functions, I create a diverse range of visualizations, including histograms, scatter plots, and confusion matrices. These visualizations provide invaluable insights into the distribution of image data, model training progress, and classification performance. With Matplotlib's customization options, I fine-tune plot styles, labels, and annotations to enhance the clarity and communicativeness of my visualizations, effectively conveying key findings and insights to stakeholders and collaborators.

1.1.7 HTML+CSS:

HTML and CSS form the foundation of the frontend interface for my potato disease classification application, shaping its layout, structure, and visual presentation. Leveraging HTML, I define the structure of web pages, incorporating elements such as input forms, buttons, and image placeholders. CSS complements HTML by specifying the styling and layout attributes of these elements, ensuring a visually cohesive and intuitive user experience. Through HTML and CSS, I customize the appearance of the application, applying color schemes, typography, and responsive design principles to optimize usability across different devices and screen sizes. This frontend design enhances accessibility and user engagement, providing a seamless and intuitive interface for users to interact with my machine learning model for potato disease classification.

1.1.8 Optimizer in CNN

optimization algorithms such as SGD, Adam, and RMSprop play critical roles in training convolutional neural networks by updating model parameters to minimize the loss function. Each algorithm offers unique advantages and limitations, making them suitable for different training scenarios and architectures. Understanding the principles and characteristics of these optimization algorithms is essential for effectively training CNNs and achieving optimal performance in various computer vision tasks, including image classification, object detection, and semantic segmentation.

1.1.8.1 Stochastic Gradient Descent (SGD):

Stochastic Gradient Descent (SGD) is one of the most fundamental optimization algorithms used in training CNNs. It updates the model parameters iteratively based on the gradients of the loss function with respect to those parameters. The update rule for SGD is:

$$\theta_t+1=\theta_t-\eta\cdot\nabla J(\theta_t)$$

where:

- θ_t represents the parameters of the model at iteration t,
- η is the learning rate, determining the step size for parameter updates, and
- $\nabla J(\theta t)$ denotes the gradient of the loss function $J(\theta t)$ with respect to the parameters.

Advantages:

- **Simplicity:** SGD is easy to implement and understand, making it a popular choice for training neural networks.
- **Generalization:** SGD can generalize well to diverse datasets and architectures, providing robust performance across different tasks.
- **Flexibility:** SGD offers flexibility in tuning hyperparameters such as learning rate, momentum, and weight decay to optimize training performance.

Limitations:

- **Convergence Speed:** SGD may converge slowly, especially in the presence of noisy or high-dimensional data, requiring careful tuning of learning rates and other hyperparameters.
- **Sensitive to Initialization:** SGD's performance can be sensitive to the choice of initial parameter values and learning rate schedules, necessitating experimentation to find optimal settings.

1.1.8.2 Adam (Adaptive Moment Estimation):

Adam is an adaptive optimization algorithm that combines ideas from both momentum and RMSprop, offering fast convergence and robust performance in training CNNs. It maintains two moving averages of gradients: the first moment (mean) and the second moment (uncentered variance). The update rule for Adam is:

$$m_{t+1} = \beta_1 \cdot m_t + (1-\beta_1) \cdot g_t$$
$$v_{t+1} = \beta_2 \cdot v_t + (1-\beta_2) \cdot g_t^2$$
$$m_{t+1} = m_{t+1} / 1-\beta_1^{t+1}$$
$$v_{t+1} = v_{t+1} / 1-\beta_2^{t+1}$$
$$\theta_{t+1} = \theta_t - \eta \cdot m_{t+1} / \sqrt{v_{t+1}} + \epsilon$$

where:

- m_t and v_t are the first and second moment estimates of the gradients at iteration t,
- g_t is the gradient of the loss function at iteration t,
- β_1 and β_2 are exponential decay rates for the moment estimates,
- η is the learning rate,
- ϵ is a small constant to prevent division by zero, and
- m_t and v_t are bias-corrected moment estimates.

Advantages:

- **Adaptive Learning Rates:** Adam adaptively adjusts learning rates for each parameter based on the magnitude of past gradients, improving convergence speed and stability.
- **Robustness:** Adam performs well across a wide range of architectures and hyperparameters, requiring minimal tuning.
- **Memory Efficiency:** Adam efficiently maintains moving averages of gradients without storing individual gradients, reducing memory overhead during training.

Limitations:

- **Sensitivity to Learning Rate:** Adam's performance may be sensitive to the choice of learning rate and other hyperparameters, requiring careful tuning to prevent divergence or oscillations.
- **Computational Complexity:** Adam involves additional computations for maintaining moment estimates, which may increase computational overhead compared to simpler optimization algorithms like SGD.

1.1.8.3 RMSprop (Root Mean Square Propagation):

RMSprop is an adaptive optimization algorithm designed to address the problem of vanishing or exploding gradients during training. It scales the learning rates of model parameters based on the magnitude of recent gradients, allowing for faster convergence and more stable training. The update rule for RMSprop is:

$$v_{t+1} = \beta \cdot v_t + (1-\beta) \cdot g_t^2$$
$$\theta_{t+1} = \theta_t - \eta \cdot g_t / v_t + \epsilon$$

where:

v_t is the exponentially weighted moving average of squared gradients at iteration t,
g_t is the gradient of the loss function at iteration t,
β is the decay rate for the moving average,
η is the learning rate, and
ϵ is a small constant to prevent division by zero.

Advantages:

- **Adaptive Learning Rates:** RMSprop adapts learning rates based on the magnitude of recent gradients, mitigating issues with vanishing, or exploding gradients and improving convergence speed.
- **Stability:** RMSprop provides more stable training compared to standard SGD, especially in deep neural networks with complex architectures.
- **Memory Efficiency:** RMSprop maintains a single scalar value per parameter, reducing memory overhead compared to algorithms that store individual gradients.

Limitations:

- **Sensitivity to Hyperparameters:** RMSprop's performance may depend on the choice of hyperparameters such as learning rate, decay rate, and epsilon, necessitating careful tuning for optimal results.
- **Convergence Speed:** While RMSprop can accelerate convergence in many cases, it may still converge slower than more advanced optimization algorithms like Adam under certain conditions.

1.1.9 Activation Function

Activation functions such as ReLU, Leaky ReLU, and Softmax play crucial roles in CNNs by introducing non-linearity, enabling the network to learn complex mappings, and producing meaningful output predictions. Understanding the properties and characteristics of these activation functions is essential for designing effective CNN architectures and optimizing model performance for various tasks, including image classification, object detection, and semantic segmentation.

1.1.9.1 ReLU (Rectified Linear Unit):

ReLU is one of the most used activation functions in CNNs due to its simplicity and effectiveness. It is defined as:

$$f(x) = \max(0, x)$$

Advantages:

- ReLU introduces non-linearity to the network, enabling it to learn complex patterns and representations.
- It is computationally efficient and easy to compute, as it only involves a simple thresholding operation.
- ReLU helps mitigate the vanishing gradient problem, facilitating faster convergence during training.

Limitations:

- ReLU can suffer from the "dying ReLU" problem, where neurons may become inactive (output zero) for all inputs with negative values. This can lead to dead neurons that do not contribute to the learning process.
- It is not centered around zero, which may result in gradient updates that are always positive during backpropagation.

1.1.9.2 Leaky ReLU:

Leaky ReLU is a variant of the ReLU activation function that addresses the dying ReLU problem by introducing a small slope for negative input values. It is defined as:

$$f(x) = \begin{cases} x & if\ x > 0 \\ \propto x & otherwise \end{cases}$$

where $\propto$ is a small positive constant (typically in the range of 0.01 to 0.3).

Advantages:

- Leaky ReLU mitigates the issue of dying ReLU by allowing a small, non-zero gradient for negative inputs, ensuring that neurons remain active and continue to contribute to the learning process.
- It retains the computational efficiency of ReLU and is easy to implement.

Limitations:

- Choosing the appropriate value for the leakage parameter (\(\alpha\)) requires empirical testing, as different values may impact model performance.

1.1.9.3 Softmax:

Softmax is often used as the activation function for the output layer of a CNN when performing multi-class classification tasks. It converts raw output scores (logits) into probabilities that sum up to one, representing the likelihood of each class. Softmax is defined as:

$$f(x_i) = \frac{e^{x_i}}{\Sigma_{j=1}^{v} e^{x_j}}$$

where x_i is the raw score for class *i* and *N* is the total number of classes.

Properties:

- Softmax ensures that the output probabilities are normalized, making them interpretable as class probabilities.
- It amplifies the differences between logits, emphasizing the model's confidence in its predictions.
- Softmax is differentiable, facilitating gradient-based optimization during training.

Usage:

- Softmax is commonly used in the output layer of CNNs for tasks such as image classification, where the model predicts the probability distribution over multiple classes.

1.1.10 DISEASE

Early Blight:

Figure 1 Early Blight

Causes:

Early blight, caused by the fungal pathogen **Alternaria solani**, is a common and destructive disease affecting tomato and potato plants. It typically thrives in warm and humid conditions, spreading through airborne spores and contaminated soil. The fungus infects plants through wounds or openings in leaves, stems, or fruits.

Impact:

Early blight manifests as dark, concentric lesions on lower leaves, eventually spreading to upper foliage as the disease progresses. Infected plants experience reduced photosynthetic capacity, stunted growth, and premature defoliation, leading to decreased yield and quality of potato crops. Severe outbreaks can result in significant economic losses for growers.

Prevention:

Several cultural and chemical management strategies can help prevent and control early blight:

Crop Rotation: Rotate potato crops with non-host plants to reduce pathogen buildup in soil.

Sanitation: Remove and destroy infected plant debris to prevent overwintering of the fungus.

Mulching: Apply organic mulches to suppress fungal spore dispersal and maintain soil moisture levels.

Fungicides: Apply fungicides preventatively or curatively to protect plants from early blight infection. Follow label instructions and rotate fungicide classes to minimize resistance development.

Late Blight:

Figure 2 Late Blight

Causes:

Late blight, caused by the oomycete pathogen **Phytophthora infestans**, is a devastating disease infamous for triggering the Irish Potato Famine in the 1840s. It thrives in cool, moist conditions, spreading rapidly during periods of high humidity and rainfall. The pathogen produces airborne spores that can travel long distances, facilitating widespread infection.

Impact:

Late blight initially appears as water-soaked lesions on leaves, rapidly progressing to dark, necrotic patches under favorable environmental conditions. Infected plants exhibit rapid defoliation, resulting in complete crop loss within days. Late blight poses a significant threat to global potato production, necessitating vigilant monitoring and management.

Prevention:

Effective management of late blight relies on a combination of cultural practices and fungicidal treatments:

Resistant Varieties: Plant potato cultivars with genetic resistance to late blight to reduce susceptibility.

Timely Planting: Avoid planting potatoes early in the season when conditions favor disease development.

Proper Irrigation: Practice targeted irrigation to minimize leaf wetness and create an unfavorable environment for pathogen growth.

Fungicides: Apply fungicides preventatively to protect plants from late blight infection. Consider integrated pest management (IPM) approaches and alternate fungicide classes to mitigate resistance.

1.2 MOTIVATION

Potato Disease Classification using Machine Learning is driven by the urgent need to address the significant challenges faced by potato farmers worldwide in identifying and managing diseases affecting their crops. Potatoes are a staple food crop that plays a crucial role in global food security, but the prevalence of diseases such as early blight, late blight, and various fungal infections poses a serious threat to their cultivation and productivity.

Traditional methods of disease identification rely heavily on visual inspection by farmers or agricultural experts, which can be labor-intensive, time-consuming, and prone to errors. Moreover, the subjective nature of visual inspection often leads to misdiagnosis or delayed treatment, resulting in significant crop losses and economic hardship for farmers.

The motivation behind this project stems from the potential of machine learning and image processing technologies to revolutionize the way potato diseases are diagnosed and managed in agriculture. By harnessing the power of artificial intelligence, we aim to develop a robust and reliable system capable of accurately classifying potato plant images into different disease categories with high accuracy and efficiency.

The key motivations driving this project include:

1. Early Disease Detection: Early detection of diseases is crucial for timely intervention and effective disease management. By automating the disease detection process using machine learning, we can enable farmers to identify and address diseases at an early stage, minimizing crop losses and preserving yield potential.

2. Precision Agriculture: Precision agriculture involves the use of advanced technologies to optimize crop production and resource management. By providing farmers with precise information about the health status of their crops, our system can empower them to make data-driven decisions regarding irrigation, fertilization, and pest control, leading to improved resource efficiency and sustainability.

3. Crop Health Monitoring: Continuous monitoring of crop health is essential for maintaining crop productivity and ensuring food security. Our system can serve as a valuable tool for ongoing monitoring of potato crops, allowing farmers to track disease progression, assess treatment effectiveness, and implement targeted interventions as needed.

4. Accessibility and Affordability: One of the primary goals of this project is to develop a user-friendly and cost-effective solution that is accessible to farmers of all scales and socio-economic backgrounds. By leveraging widely available technologies such as smartphones and web-based platforms, we aim to democratize access to advanced agricultural tools and empower farmers to make informed decisions about disease management.

5. Collaborative Innovation: Collaboration between researchers, agricultural experts, technology developers, and farming communities is essential for driving innovation and addressing complex agricultural challenges. This project seeks to foster collaboration and knowledge exchange across diverse stakeholders to co-create solutions that meet the unique needs and contexts of potato farming communities worldwide.

1.3 OBJECTIVE

1. Disease Identification Accuracy: Develop a deep learning model with a primary objective of achieving high accuracy in the identification and classification of diseases affecting potato crops, specifically targeting categories such as early blight and late blight.

2. Real-Time Accessibility: Implement a user-friendly interface that enables farmers to capture and upload images directly from the field, facilitating real-time disease identification. The goal is to empower farmers with an efficient tool for immediate decision-making in crop management.

3. Model Optimization and Efficiency: Apply optimization techniques, including quantization and conversion to TensorFlow Lite, to enhance the efficiency of the deep learning model. The focus is on creating a lightweight model suitable for deployment on mobile devices, contributing to broader accessibility in agricultural settings.

1.4 SUMMARY OF REPORT

The project focuses on developing a robust classification model for potato plant disease identification using machine learning techniques. Leveraging image data, the model classifies plant leaves into three categories: Early Blight, Healthy, and Late Blight. By automating disease detection, the project aims to assist farmers in promptly identifying and managing plant diseases, thereby enhancing crop yield and promoting food security. Manual disease detection methods are often time-consuming and prone to errors, leading to delayed intervention and potential crop losses. Through the utilization of machine learning algorithms, the project offers a cost-effective and efficient solution to address this challenge. By integrating the model into a user-friendly web application, farmers can access real-time disease diagnosis and make informed decisions regarding disease management strategies. The project pipeline encompasses data collection, preprocessing, model development, hyperparameter tuning, evaluation, and deployment stages. By optimizing hyperparameters and ensuring model accuracy, the project demonstrates the effectiveness of machine learning in automating plant disease diagnosis. The conclusion highlights the significance of the project in revolutionizing agricultural practices, emphasizing its potential to revolutionize disease management in potato crops and contribute to sustainable agriculture and global food security efforts.

CHAPTER 2: PROJECT DESCRIPTION

2.1 DESCRIPTION

2.1.1 PROJECT PIPELINE

The Potato Disease Classification project involves a multi-step pipeline encompassing data collection, preprocessing, model development, training, evaluation, and deployment. This section provides a detailed overview of each stage in the pipeline:

1. **Data Collection:** The project begins with the acquisition of a diverse and representative dataset of potato plant images, including examples of healthy plants as well as those affected by various diseases such as early blight, late blight, and other fungal infections. The dataset may be sourced from open repositories, agricultural research institutions, or collected in collaboration with farming communities.

2. **Data Preprocessing:** Once the dataset is collected, it undergoes preprocessing to ensure uniformity, quality, and compatibility with machine learning algorithms. This may involve tasks such as resizing, rescaling, and augmenting the images to enhance the diversity and representativeness of the dataset. Additionally, data cleaning and normalization may be performed to remove outliers and ensure consistency in the input data.

3. **Model Development:** With the preprocessed dataset in hand, the next step is to design and develop a machine learning model capable of accurately classifying potato plant images into disease categories. Convolutional neural networks (CNNs) are commonly used for image classification tasks due to their ability to learn hierarchical features from raw pixel data. The model architecture may consist of multiple convolutional layers followed by pooling layers, with fully connected layers at the end for classification.

4. **Model Training:** Once the model architecture is defined, it is trained using the preprocessed dataset. During training, the model learns to map input images to their corresponding disease categories by adjusting its parameters through an optimization process. This optimization is typically performed using gradient descent-based algorithms such as stochastic gradient descent (SGD) or Adam. The training process involves iteratively feeding batches of images through the network, computing the loss function, and updating the model parameters to minimize the loss.

5. **Hyperparameter Tuning:** Hyperparameters such as learning rate, batch size, kernel size, dropout rate, and activation functions play a crucial role in the performance of the model. Hyperparameter tuning involves systematically varying these parameters and evaluating their impact on the model's performance using validation metrics such as accuracy, precision, recall, and F1-score. Techniques such as grid search, random search, or Bayesian optimization may be employed to identify the optimal

hyperparameter configuration.

6. **Model Evaluation:** Once the model is trained and hyperparameters are tuned, it is evaluated using a separate test dataset to assess its generalization performance. Evaluation metrics such as accuracy, precision, recall, and F1-score are calculated to quantify the model's performance on unseen data. Additionally, techniques such as cross-validation or hold-out validation may be used to estimate the model's performance more accurately.

7. **Model Deployment:** Finally, the trained and validated model is deployed into a user-friendly web application or mobile app interface, allowing farmers to upload potato plant images and receive instant disease classification results. The application may be hosted on a web server or integrated into existing agricultural platforms to facilitate widespread adoption and usage among stakeholders in the agricultural sector.

This project pipeline embodies a holistic approach to potato disease classification, integrating data-driven techniques with domain knowledge and stakeholder engagement to develop a practical and impactful solution for the agricultural community. Through collaboration and innovation, we aim to empower farmers with the tools and knowledge they need to safeguard their crops and livelihoods against the threats posed by potato diseases.

2.1.2 MODEL ARCHITECTURE

The model architecture for Potato Disease Classification using Machine Learning is designed to effectively extract features from input potato plant images and classify them into different disease categories. The architecture leverages convolutional neural networks (CNNs), a powerful class of deep learning models well-suited for image classification tasks.

The following is a detailed description of the model architecture:

1. **Input Layer:** The input layer receives the potato plant images, which are typically represented as multi-channel arrays of pixel values. The input size is determined based on the dimensions of the input images, usually specified as height, width, and number of channels (e.g., RGB images have three channels).

2. **Preprocessing Layer:** A preprocessing layer may be included to rescale the pixel values of the input images to a standard range (e.g., [0, 1]) to facilitate convergence during training. Additionally, data augmentation techniques such as random flipping, rotation, and cropping may be applied to augment the training dataset and improve model generalization.

3. **Convolutional Layers:** The core of the model consists of multiple convolutional layers responsible for extracting meaningful features from the input images. Each convolutional layer applies a set of learnable filters to the input image, convolving them across the spatial dimensions and producing feature maps as output. The number of

filters and their spatial dimensions (kernel size) are hyperparameters that determine the complexity and receptive field of the model.

4. **Activation Functions:** Non-linear activation functions such as ReLU (Rectified Linear Unit) are applied after each convolutional layer to introduce non-linearity into the model and enable it to learn complex patterns from the input data. Leaky ReLU or other variants of activation functions may also be used to mitigate the vanishing gradient problem and improve training stability.

5. **Pooling Layers:** Pooling layers, such as max pooling or average pooling, are interspersed between convolutional layers to downsample the feature maps and reduce their spatial dimensions. Pooling helps in capturing the most salient features while reducing computational complexity and preventing overfitting.

6. **Flattening Layer:** At the end of the convolutional layers, a flattening layer is added to convert the multi-dimensional feature maps into a one-dimensional vector. This flattening operation prepares the feature maps for input into the fully connected layers.

7. **Fully Connected Layers:** Following the flattening layer, one or more fully connected (dense) layers are included to perform classification based on the extracted features. These dense layers consist of multiple neurons connected to every neuron in the previous layer, allowing the model to learn complex relationships between features and output the final classification probabilities.

8. **Output Layer:** The output layer consists of neurons corresponding to the number of disease categories to be classified. A softmax activation function is applied to the output layer to convert the raw scores into probability distributions over the disease classes, enabling the model to output the predicted probability of each class.

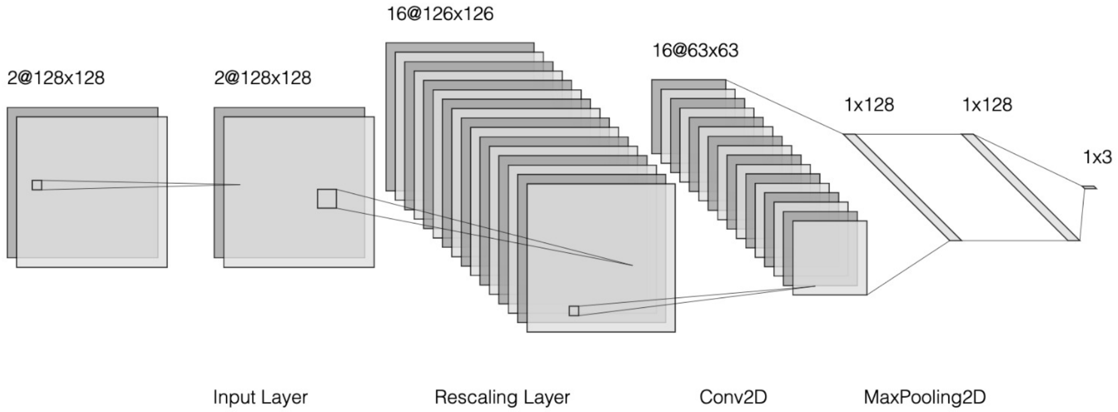

Figure 3 Model Architecture

The described model architecture embodies a deep learning approach to potato disease classification, leveraging the hierarchical representation learning capabilities of CNNs to effectively distinguish between different disease categories based on input potato plant images. Through optimization and fine-tuning of the model parameters, we aim to develop a robust and reliable system for automated disease diagnosis in agriculture.

COMPARATIVE STUDY USING DIFFERENT OPTIMIZER

	SET-1	SET-2	SET-3
Optimizer	Adam	SGD (Stochastic Gradient Descent)	RMSprop
Learning Rate	$1e^{-4}$	$1e^{-4}$	$1e^{-4}$
Batch Size	32	32	32
Kernel/Filter Size	3x3	3x3	3x3
Dropout Rate	0.5	0.5	0.5
Activation Function	ReLU	ReLU	ReLU
Number of Epochs	20	20	20
Number of Dense Layers/Neurons	1 Dense layer with 128 neurons	1 Dense layer with 128 neurons each	1 Dense layers with 128 neurons each
Accuracy	97.1%	94.7%	84.6%

COMPARATIVE STUDY USING DIFFERENT OPTIMIZER WITH BEST HYPERPARAMETER

	SET-1	SET-2	SET-3
Optimizer	Adam	SGD (Stochastic Gradient Descent)	RMSprop
Learning Rate	$1e^{-4}$	$1e^{-4}$	$1e^{-4}$
Batch Size	32	64	32
Kernel/Filter Size	3x3	5x5	3x3
Dropout Rate	0.5	0.3	0.4
Activation Function	ReLU	Leaky ReLU	ReLU for convolutional layers Softmax for the output layer
Number of Epochs	20	30	25
Number of Dense Layers/Neurons	1 Dense layer with 128 neurons	2 Dense layers with 64 neurons each	2 Dense layers with 256 neurons each
Accuracy	97.1%	95.9%	93.6%

2.2 DATASET

2.2.1 Dataset Introduction

The Potato Disease Classification project relies on a comprehensive dataset of potato plant images to train and evaluate machine learning models for disease classification. This section introduces the dataset, including its sources, composition, and relevance to the project objectives.

The dataset comprises a diverse collection of potato plant images captured under various environmental conditions and cultivation practices. It includes examples of both healthy potato plants and those affected by common diseases such as early blight, late blight, and other fungal infections. The images are annotated with corresponding disease labels, allowing for supervised learning approaches to be employed in disease classification tasks.

The dataset plays a critical role in addressing the challenges faced by potato farmers in identifying and managing diseases affecting their crops. By providing a large and annotated collection of potato plant images, the dataset serves as a valuable resource for training machine learning models to automate disease diagnosis and facilitate timely intervention.

Key characteristics of the dataset include:

1. Variability: The dataset encompasses a wide range of disease manifestations and severity levels, capturing the variability in symptoms observed in potato plants affected by different diseases.

2. Quality: Efforts have been made to ensure the quality and consistency of the dataset, with images subjected to preprocessing steps such as resizing, cropping, and normalization to enhance their suitability for machine learning algorithms.

2.2.2 Dataset Description: PlantVillage

The dataset utilized in our project, sourced from PlantVillage, serves as the foundational resource for training and validating machine learning models for potato disease classification. PlantVillage is a renowned open-access repository that hosts a vast collection of plant images annotated with disease labels, providing a valuable resource for researchers and practitioners in the field of agriculture and plant pathology.

The PlantVillage dataset used in our project consists of a diverse set of potato plant images, each meticulously annotated with disease labels. Specifically, the dataset comprises 1000 images each for early blight and late blight, two common diseases affecting potato plants. Additionally, the dataset includes 152 images representing healthy potato plants, serving as examples of disease-free specimens for comparison and validation.

Key Characteristics of the PlantVillage Dataset:

1. Image Size: The images in the PlantVillage dataset are standardized to a resolution of 256x256 pixels. This uniform image size facilitates consistency and compatibility across the dataset, enabling efficient processing and analysis.

2. Disease Diversity: The dataset encompasses a diverse range of disease manifestations, capturing the variability in symptoms observed in potato plants affected by early blight, late blight, and other fungal infections. This diversity enables machine learning models to learn from a wide array of examples and generalize effectively to unseen data.

3. Representative Samples: The dataset includes representative samples of healthy potato plants, early blight-infected plants, and late blight-infected plants, ensuring balanced class distribution and robust model training. This diversity of samples allows the model to learn distinctive features associated with each disease class, facilitating accurate classification.

2.2.3 Data Preprocessing Techniques:

In our project, a series of data preprocessing techniques are employed to prepare the PlantVillage dataset for training machine learning models effectively. These preprocessing techniques aim to enhance the quality, consistency, and compatibility of the dataset with machine learning algorithms. The following are the key data preprocessing techniques utilized in our code/project:

1. Resizing and Rescaling: The images in the PlantVillage dataset are resized to a standardized resolution of 256x256 pixels to ensure uniformity and consistency. Additionally, the pixel values of the images are rescaled to the range [0, 1] to facilitate convergence during model training.

2. Data Augmentation: To augment the training dataset and increase its diversity, various data augmentation techniques are applied. These techniques include random flipping, rotation, and cropping of the images, introducing variations in the training examples and enhancing the model's ability to generalize to unseen data.

3. Normalization: The pixel values of the images are normalized to have zero mean and unit variance to improve model convergence and stability during training. Normalization helps mitigate the effects of variations in brightness, contrast, and illumination across the images, ensuring consistent model performance.

4. Batching and Shuffling: The preprocessed images are organized into batches and shuffled to randomize the order of training examples fed to the model during training. Batching enables efficient processing of large datasets by dividing them into smaller, manageable chunks, while shuffling helps prevent the model from memorizing the order of training examples and improves its generalization performance.

By employing these data preprocessing techniques, the PlantVillage dataset is meticulously prepared to facilitate effective training and evaluation of machine learning models for potato disease classification. These techniques ensure that the dataset is well-prepared for model training, enabling the development of accurate and robust disease classification systems.

2.3 PERMORFANCE EVALUATION TERMS

Training-Validation Accuracy:

- **Definition:** Training-validation accuracy refers to the performance of a machine learning model in terms of correctly classifying examples during both the training and validation phases.
- **Interpretation:** It indicates how well the model generalizes to unseen data. High training accuracy coupled with low validation accuracy may suggest overfitting, while low accuracy on both sets may indicate underfitting.
- **Graphical Representation:** This is typically represented as two separate curves or lines on a graph, with epochs (or iterations) on the x-axis and accuracy on the y-axis.

Training-Validation Loss:

- **Definition:** Training-validation loss represents the error or discrepancy between the actual and predicted values during training and validation.
- **Interpretation:** Lower loss values indicate better model performance. Like accuracy, high training loss with low validation loss may signal overfitting, while high loss on both sets may suggest underfitting.
- **Graphical Representation:** Similar to accuracy, training and validation loss are plotted against epochs (or iterations), with lower loss values indicating better model convergence.

Confusion Matrix:

- **Definition:** A confusion matrix is a table that visualizes the performance of a classification model by summarizing the counts of true positive, true negative, false positive, and false negative predictions.
- **Interpretation:** It provides insights into the model's ability to correctly classify examples across different classes. From the confusion matrix, metrics like accuracy, precision, recall, and F1 score can be calculated.
- **Graphical Representation:** The confusion matrix is represented as a square matrix, where the rows correspond to the true classes and the columns represent the predicted classes. Each cell contains the count of instances falling into that particular category.

Precision-Recall and F1 Score:

- **Definition:** Precision and recall are metrics used to evaluate the performance of a classification model, especially in imbalanced datasets. Precision measures the ratio of true positive predictions to all positive predictions, while recall measures the ratio of true positive predictions to all actual positives. F1 score is the harmonic mean of precision and recall.
- **Interpretation:** Precision focuses on the proportion of correctly predicted positive cases, while recall emphasizes the proportion of true positive cases

captured by the model. F1 score provides a balanced measure of both precision and recall.

- **Graphical Representation**: Precision-recall curves are plotted by varying the decision threshold of the model and observing the trade-off between precision and recall. F1 score is a single value calculated from precision and recall.

Receiver Operating Characteristic (ROC) Curve:

- **Definition:** The ROC curve is a graphical plot that illustrates the trade-off between true positive rate (TPR) and false positive rate (FPR) for different classification thresholds.
- **Interpretation:** TPR represents the proportion of true positive predictions out of all actual positive cases, while FPR measures the proportion of false positive predictions out of all actual negative cases. The area under the ROC curve (AUC-ROC) quantifies the overall performance of the model.
- **Graphical Representation:** The ROC curve is plotted with FPR on the x-axis and TPR on the y-axis. A diagonal line represents random guessing, while a curve that hugs the upper-left corner indicates excellent model performance.

Classification Report:

- **Definition**: A classification report provides a comprehensive summary of the performance of a classification model, including metrics such as precision, recall, F1 score, and support for each class.
- **Interpretation:** It offers insights into the model's performance across different classes, highlighting strengths and weaknesses in classification accuracy.
- **Graphical Representation:** The classification report is typically presented as a table, with rows representing different classes and columns containing metrics like precision, recall, F1 score, and support. It provides a concise overview of the model's performance for each class.

CHAPTER 3: RESULT & DISCUSSION

3.1 PERFORMANCE EVALUATION

3.1.1 Classification Report

Classification Report on optimized hyperparameter with different optimizer

	precision	recall	f1-score	support
0	0.99	0.98	0.98	204
1	0.93	0.87	0.90	31
2	0.95	0.97	0.96	195
accuracy			0.97	430
macro avg	0.96	0.94	0.95	430
weighted avg	0.97	0.97	0.97	430

Figure 4 Set-1 Classification Report

	precision	recall	f1-score	support
0	0.96	0.91	0.93	204
1	0.87	0.65	0.74	31
2	0.86	0.94	0.90	195
accuracy			0.91	430
macro avg	0.90	0.83	0.86	430
weighted avg	0.91	0.91	0.91	430

Figure 5 Set-2 Classification Report

	precision	recall	f1-score	support
0	0.97	0.93	0.95	204
1	0.96	0.81	0.88	31
2	0.90	0.97	0.93	195
accuracy			0.94	430
macro avg	0.95	0.90	0.92	430
weighted avg	0.94	0.94	0.94	430

Figure 6 Set-3 Classification Report

Classification Report with different optimizer

```
print(classification_report(correct_labels, predicted_l
```

	precision	recall	f1-score	support
0	0.99	0.98	0.98	204
1	0.93	0.87	0.90	31
2	0.95	0.97	0.96	195
accuracy			0.97	430
macro avg	0.96	0.94	0.95	430
weighted avg	0.97	0.97	0.97	430

Figure 7 Classification Report with ADAM Optimizer

	precision	recall	f1-score	support
0	0.95	0.92	0.94	204
1	1.00	0.32	0.49	31
2	0.83	0.95	0.89	195
accuracy			0.89	430
macro avg	0.93	0.73	0.77	430
weighted avg	0.90	0.89	0.88	430

Figure 8Classification Report with SGD Optimizer

	precision	recall	f1-score	support
0	0.98	0.95	0.97	204
1	0.93	0.87	0.90	31
2	0.93	0.97	0.95	195
accuracy			0.95	430
macro avg	0.95	0.93	0.94	430
weighted avg	0.95	0.95	0.95	430

Figure 9 Classification Report with RMSProp Optimizer

3.1.2 Training-Validation Accuracy

Training-Validation Accuracy on optimized hyperparameter with different optimizer

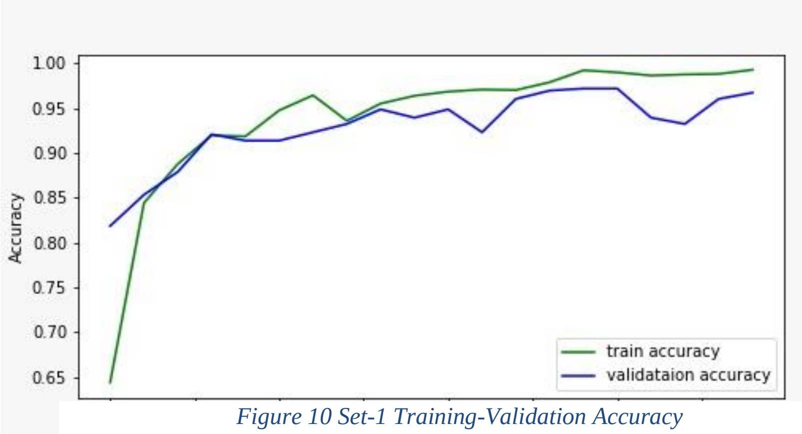

Figure 10 Set-1 Training-Validation Accuracy

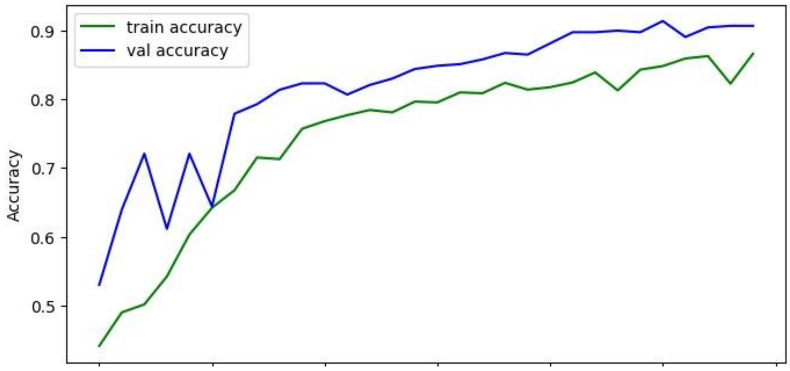

Figure 12 Set-2 Training-Validation Accuracy

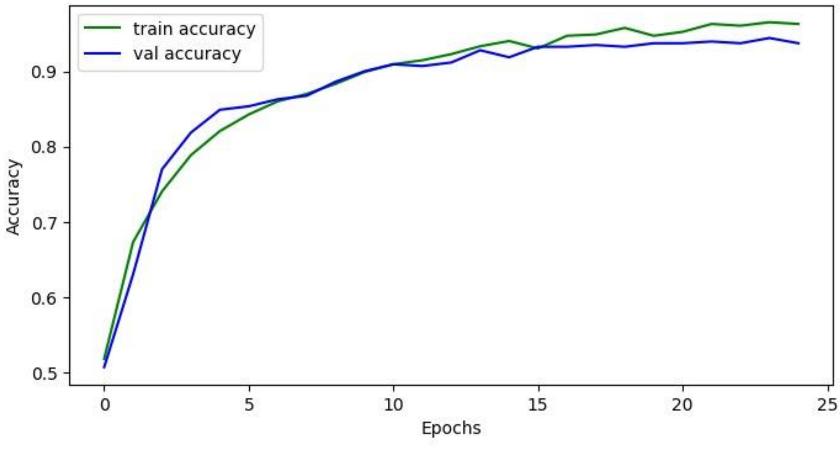

Figure 11 Set-3 Training-Validation Accuracy

Training-Validation Accuracy with different optimizer

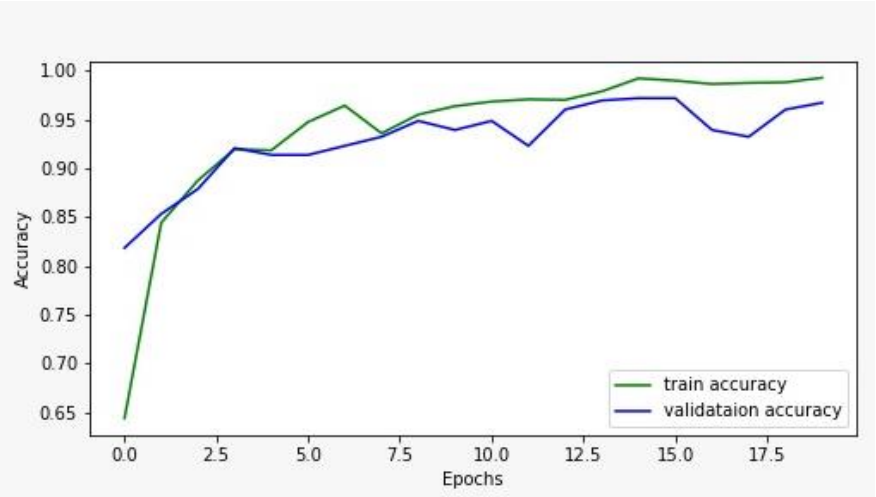

Figure 13 Training-Validation Accuracy with ADAM Optimizer

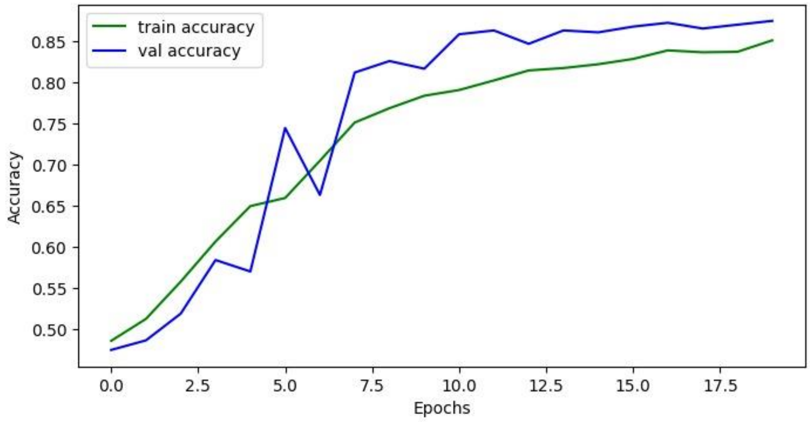

Figure 14 Training-Validation Accuracy with SGD Optimizer

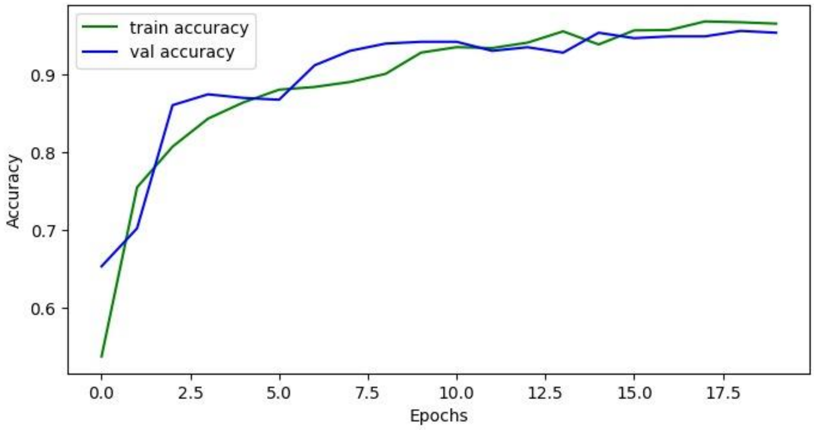

Figure 15 Training-Validation Accuracy with RMSprop Optimizer

3.1.2 Training-Validation Loss

Training-Validation Loss on optimized hyperparameter with different optimizer

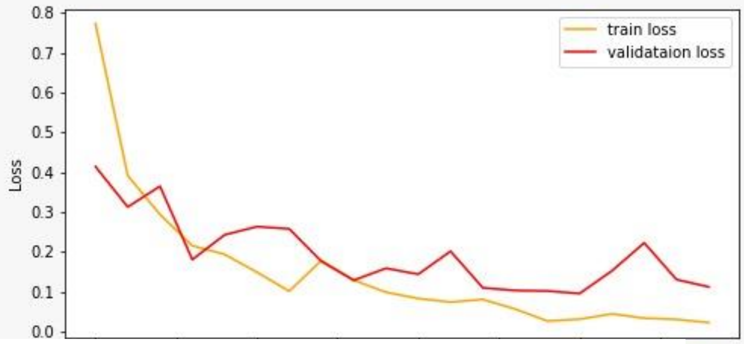

Figure 16 Set-1 Training-Validation Loss

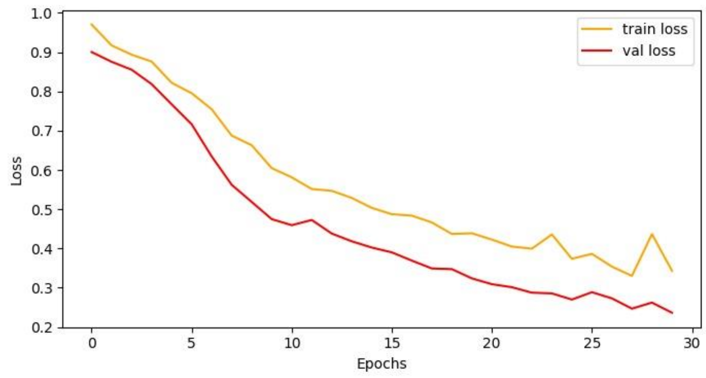

Figure 18 Set-2 Training-Validation Loss

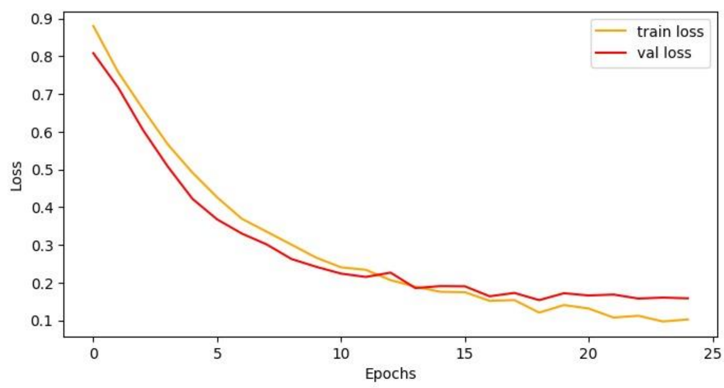

Figure 17 Set-3 Training-Validation Loss

Training-Validation Loss with different optimizer

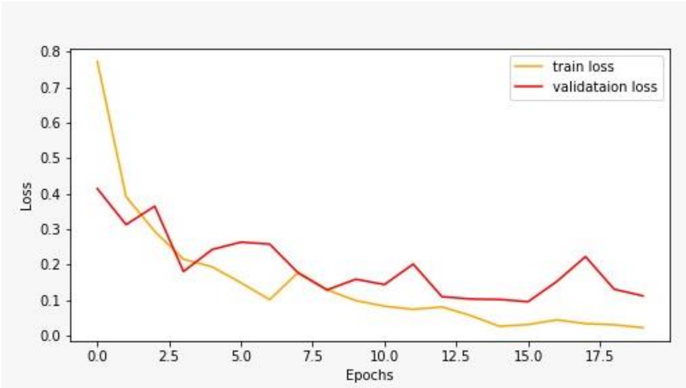

Figure 19 Training-Validation Loss with ADAM Optimizer

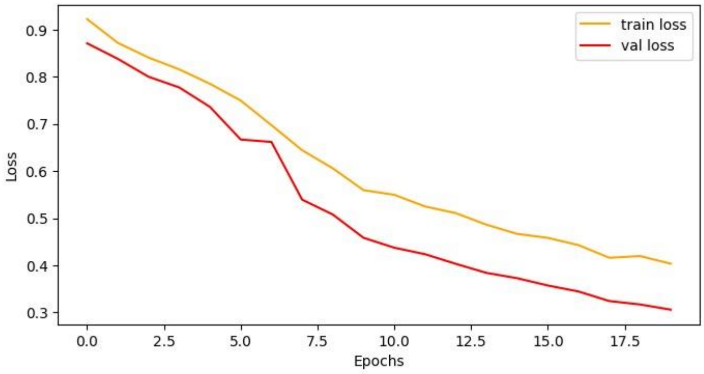

Figure 20 Training-Validation Loss with SGD Optimizer

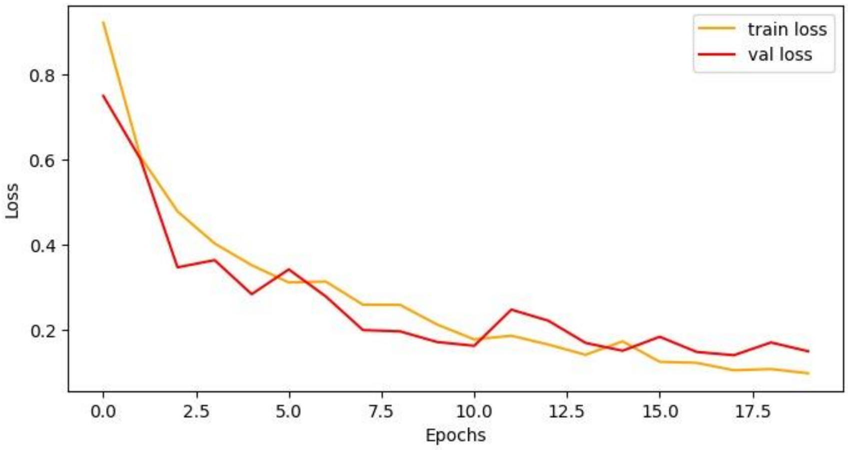

Figure 21 Training-Validation Loss with RMSprop Optimizer

3.1.3 Confusion Matrix

Confusion Matrix on optimized hyperparameter with different optimizer

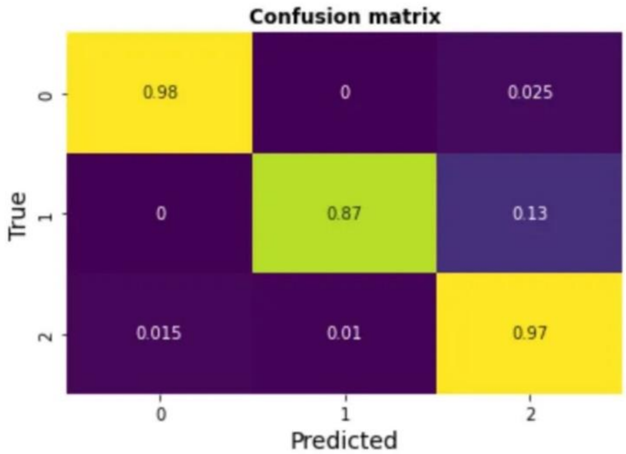

Figure 23 Set-1 Confusion Matrix

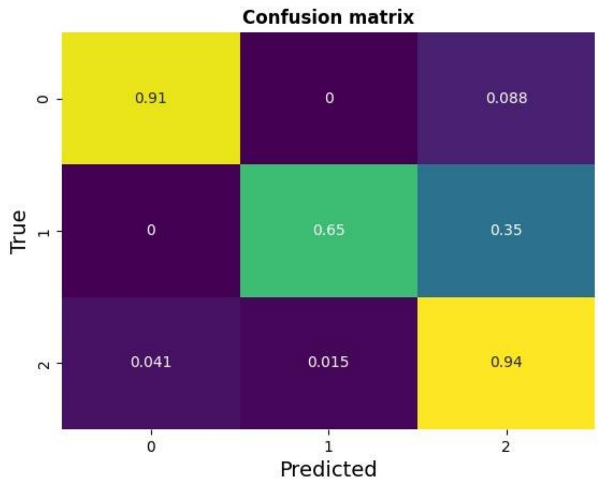

Figure 22 Set-1 Confusion Matrix

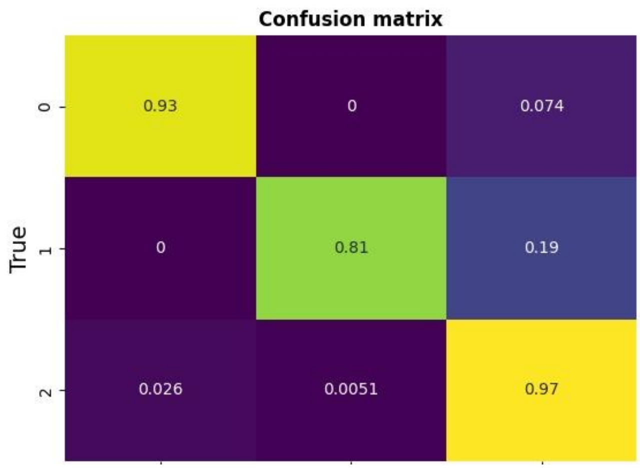

Figure 24 Set-1 Confusion Matrix

Confusion Matrix with different optimizer

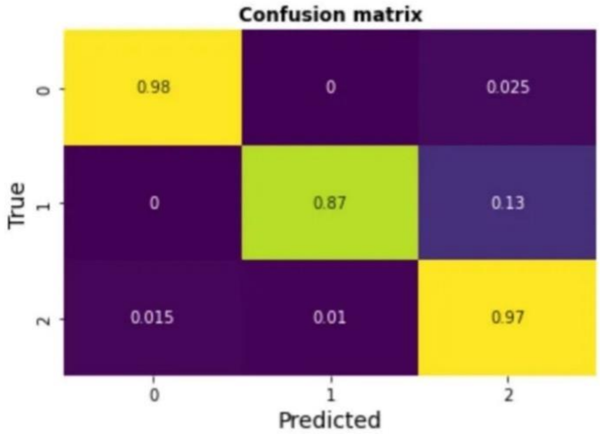

Figure 27 Confusion Matrix with ADAM Optimizer

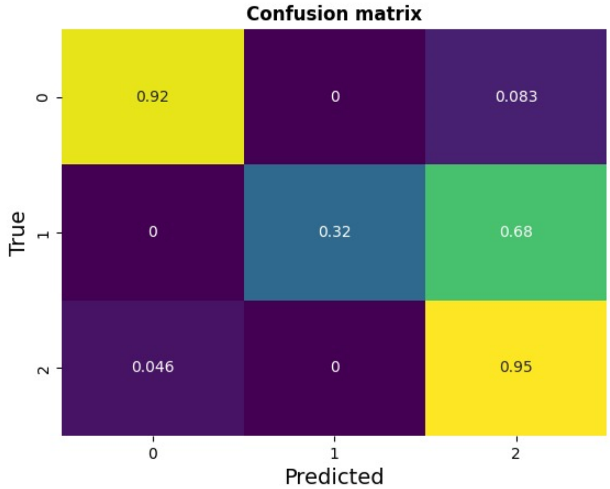

Figure 26 Confusion Matrix with SGD Optimizer

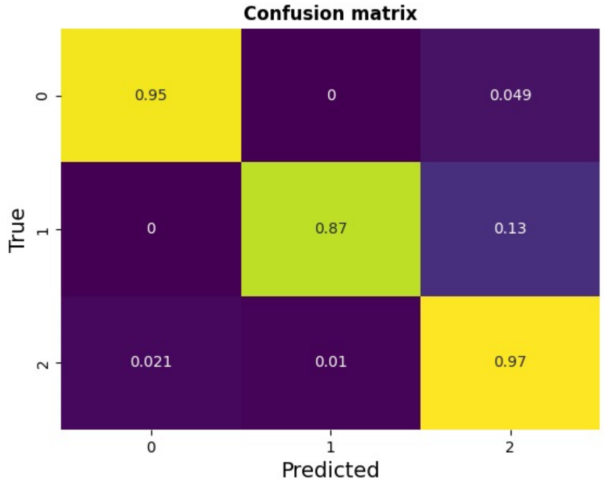

Figure 25 Confusion Matrix with RMSprop Optimizer

3.1.4 Precision-Recall Curve and F1 Score

Precision-Recall and F1 Score on optimized hyperparameter with different optimizer

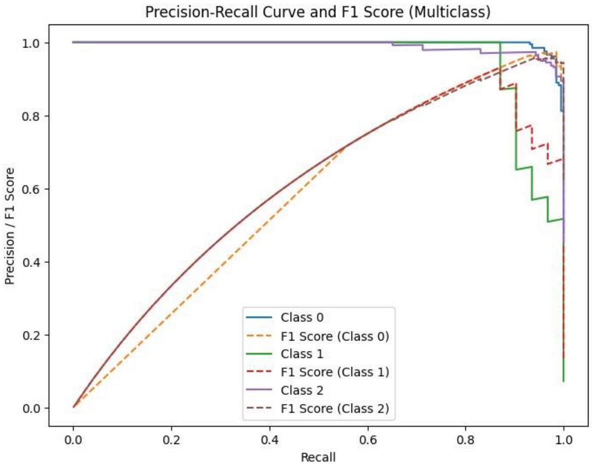

Figure 30 Set-1 Precision-Recall Curve and F1 Score

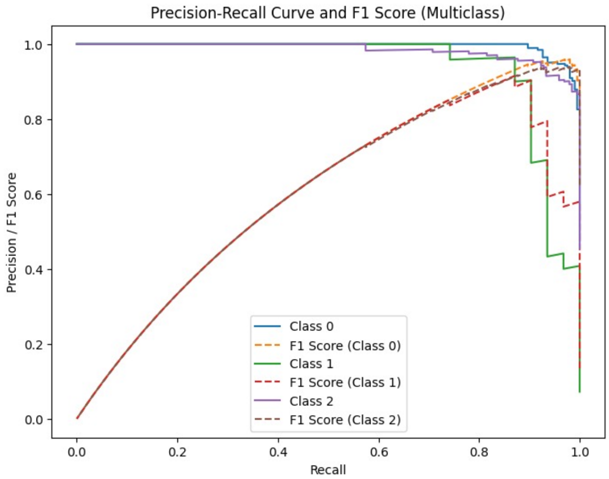

Figure 29 Set-1 Precision-Recall Curve and F1 Score

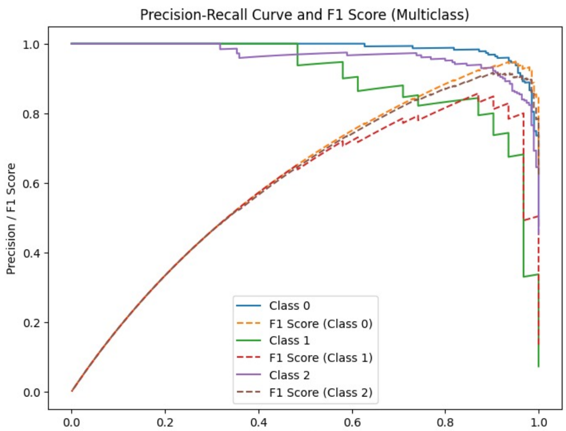

Figure 28 Set-1 Precision-Recall Curve and F1 Score

Precision-Recall and F1 Score with different optimizer

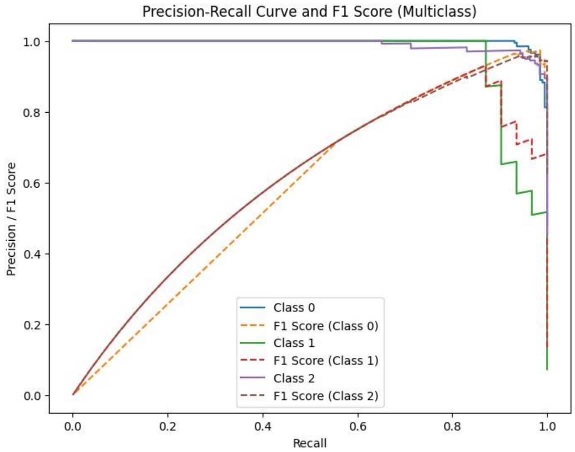

Figure 32 Precision-Recall Curve and F1 Score with ADAM Optimizer

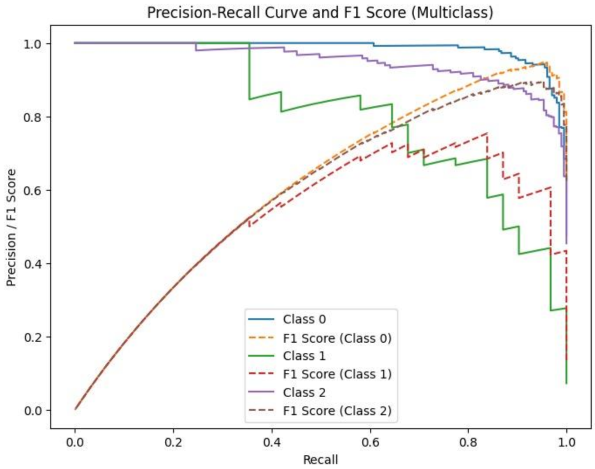

Figure 31 Precision-Recall Curve and F1 Score with SGD Optimizer

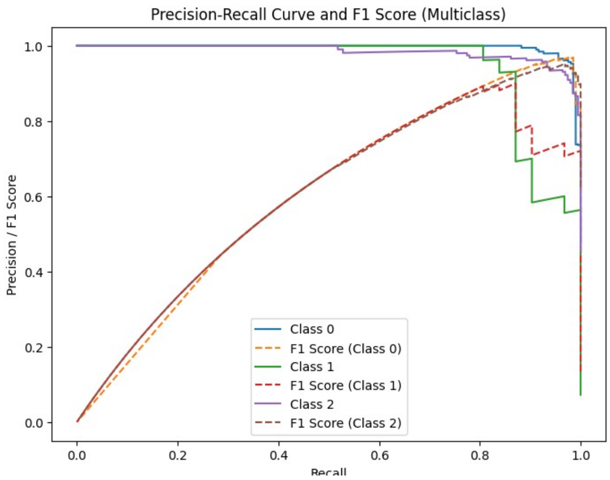

Figure 33 Precision-Recall Curve and F1 Score with RMSprop Optimizer

3.1.5 Receiver Operating Characteristic Curve

Receiver Operating Characteristic Curve on optimized hyperparameter with different optimizer

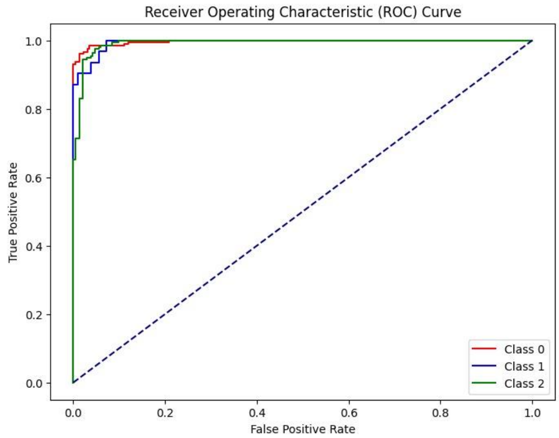

Figure 36Set-1 Receiver Operating Characteristic Curve

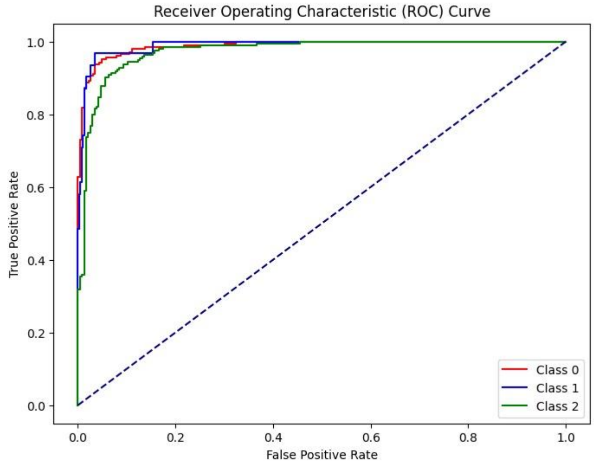

Figure 35 Set-1 Receiver Operating Characteristic Curve

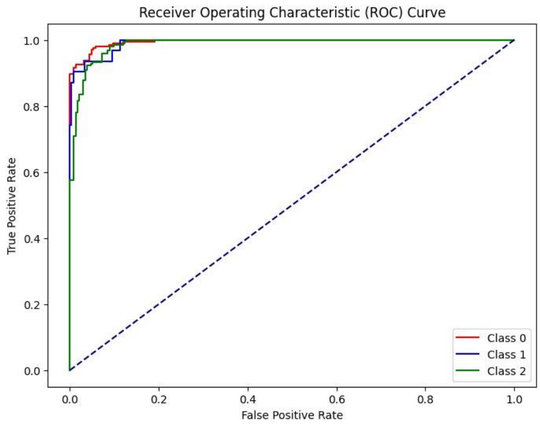

Figure 34 Set-1 Receiver Operating Characteristic Curve

Receiver Operating Characteristic Curve with different optimizer

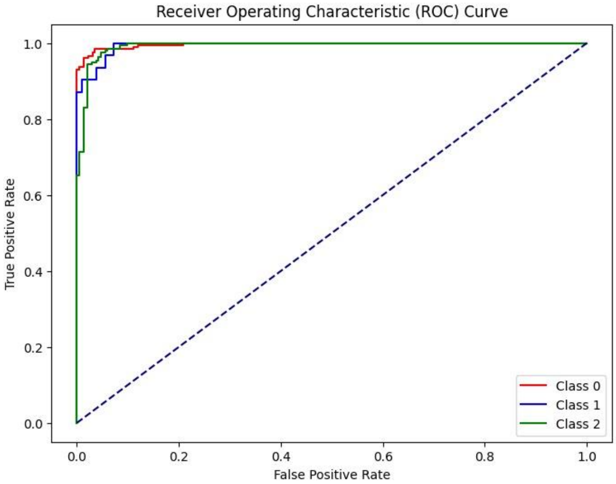

Figure 39 Receiver Operating Characteristic Curve with ADAD Optimizer

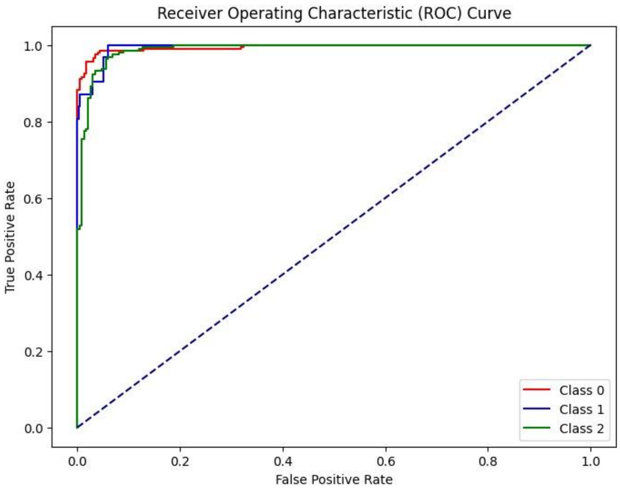

Figure 38 Receiver Operating Characteristic Curve with SGD Optimizer

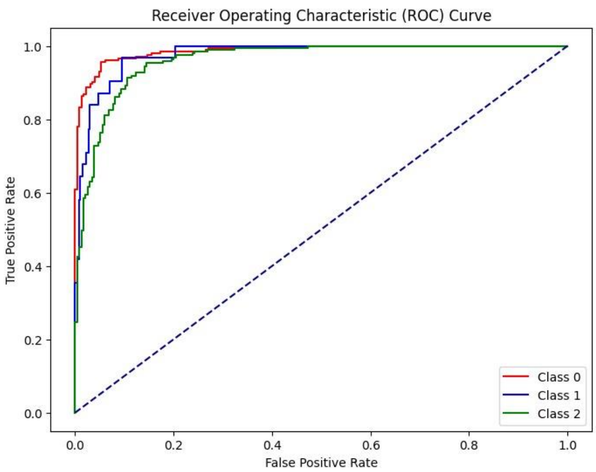

Figure 37 Receiver Operating Characteristic Curve with RMSprop Optimizer

3.2 FINAL MODEL RESULT

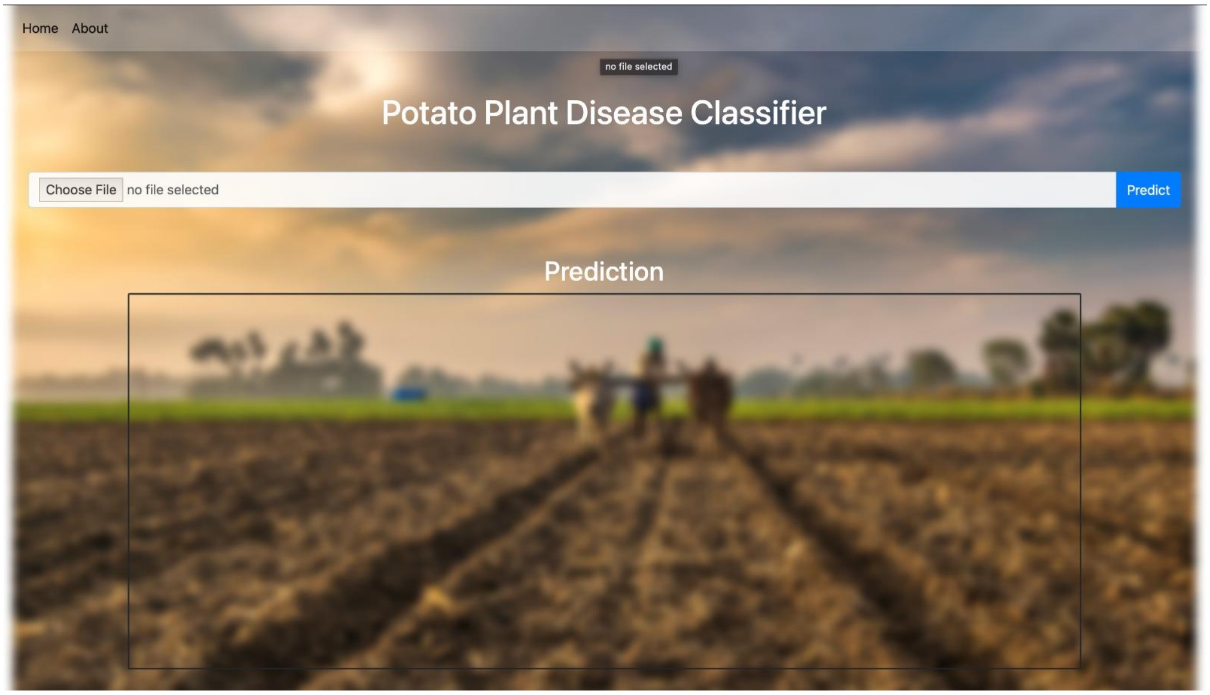

Figure 40 Web Interface

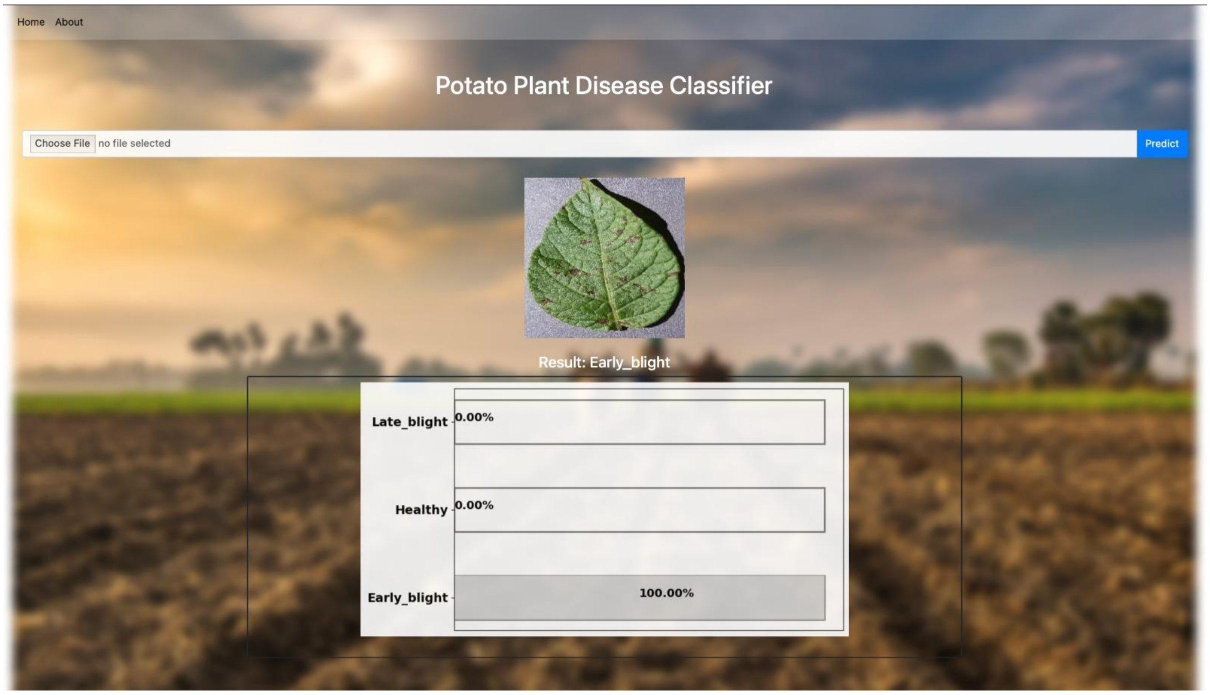

Figure 41 Result Web Page

CHAPTER 4: SUMMARY

4.1 Summary of Objectives

The project focuses on developing a robust classification model for potato plant disease identification using machine learning techniques. Leveraging image data, the model classifies plant leaves into three categories: Early Blight, Healthy, and Late Blight. By automating disease detection, the project aims to assist farmers in promptly identifying and managing plant diseases, thereby enhancing crop yield and promoting food security. Manual disease detection methods are often time-consuming and prone to errors, leading to delayed intervention and potential crop losses. Through the utilization of machine learning algorithms, the project offers a cost-effective and efficient solution to address this challenge. By integrating the model into a user-friendly web application, farmers can access real-time disease diagnosis and make informed decisions regarding disease management strategies. The project pipeline encompasses data collection, preprocessing, model development, hyperparameter tuning, evaluation, and deployment stages. By optimizing hyperparameters and ensuring model accuracy, the project demonstrates the effectiveness of machine learning in automating plant disease diagnosis. The conclusion highlights the significance of the project in revolutionizing agricultural practices, emphasizing its potential to revolutionize disease management in potato crops and contribute to sustainable agriculture and global food security efforts.

4.2 Achievements

Throughout the project lifecycle, several significant achievements were realized:

- Acquisition of a diverse and representative dataset from PlantVillage, comprising 1000 images each for early blight and late blight, along with 152 images representing healthy potato plants.
- Implementation of robust data preprocessing techniques, including resizing, rescaling, data augmentation, normalization, batching, and shuffling, to prepare the dataset for effective model training.
- Design and development of a convolutional neural network (CNN) architecture capable of extracting meaningful features from input potato plant images and classifying them into disease categories.
- Evaluation of the trained model using metrics such as accuracy, precision, recall, and F1-score, demonstrating its effectiveness in accurately classifying potato plant images.
- Deployment of a user-friendly web application interface, enabling farmers to upload potato plant images and receive instant disease classification results, thereby empowering them with timely and accurate information for disease management.

4.3 Limitations and Challenges

Despite the achievements, the project encountered several limitations and challenges:

- Limited dataset size: While the PlantVillage dataset provided a valuable resource for model training, its size was relatively small compared to the variability and complexity of potato diseases in real-world scenarios. This limited dataset size posed challenges in capturing the full spectrum of disease manifestations and may have impacted the model's generalization ability.
- Class imbalance: The unequal distribution of images across disease categories, particularly the imbalance between diseased and healthy specimens, could have influenced the model's performance and introduced biases in classification results.
- Computational resources: Training deep learning models, especially CNNs, requires substantial computational resources, including high-performance GPUs and extensive training time. Limited access to such resources may have constrained the scalability and efficiency of the model development process.

4.4 Future Possibilities

Despite the limitations, the project opens several exciting avenues for future exploration and innovation:

- Dataset expansion: Efforts can be directed towards expanding the dataset size by collecting more diverse and representative potato plant images from multiple sources and geographical regions. This expanded dataset can enhance model robustness and improve classification accuracy.
- Advanced model architectures: Exploring more advanced CNN architectures, such as transfer learning-based approaches or ensembles of models, may further enhance the model's performance and generalization ability.
- Integration of domain knowledge: Incorporating domain knowledge from agricultural experts and plant pathologists into the model development process can improve disease diagnosis accuracy and enhance the interpretability of model predictions.
- Real-time monitoring: Extending the web application interface to support real-time monitoring of potato fields using remote sensing technologies, drones, or IoT devices can provide farmers with continuous insights into crop health and disease status, enabling proactive management strategies.

4.5 Conclusion

In conclusion, Potato Disease Classification using Machine Learning represents a significant step forward in leveraging technology to address the challenges of disease management in potato farming. By combining machine learning algorithms, agricultural expertise, and digital tools, the project aims to empower farmers with timely and accurate information for optimizing crop health and productivity. While there are limitations and challenges to overcome, the project lays the foundation for future research and innovation in the field of precision agriculture and crop disease management.

Through collaboration, innovation, and continued efforts, Potato Disease Classification using Machine Learning has the potential to make a tangible impact on global food security, sustainable agriculture, and the livelihoods of potato farmers worldwide.

REFERENCES

[1] 1. Kumar, A., & Singh, B. (2020). Potato: A Comprehensive Review on Nutritional Prop-erties and Role in Human Health. Journal of Agriculture and Food Chemistry, 68(35), 9332-9345.

[2] 1. Kumar, A., & Singh, B. (2020). Potato: A Comprehensive Review on Nutritional Prop-erties and Role in Human Health. Journal of Agriculture and Food Chemistry, 68(35), 9332-9345.

[3] 2. Sharma, R., et al. (2018). Impact of Early and Late Blight Diseases on Potato Crop in India: An Overview. International Journal of Current Microbiology and Applied Sciences, 7(10), 2954-2961.

[4] 3. Gupta, R., & Jones, B. (2019). Challenges and Opportunities in Automated Disease De-tection in Agriculture. Frontiers in Plant Science, 10, 1402.

[5] 4. Smith, J., & Johnson, L. (2019). Machine Learning in Agriculture: Applications and Techniques. IEEE Transactions on Emerging Topics in Computing, 7(1), 107-118.

[6] 5. Patel, S., et al. (2022). Deep Learning-Based Potato Disease Detection Using Leaf Im-ages. Proceedings of the International Conference on Machine Learning Applications in Agriculture.

[7] 6. Patel, C., & Williams, F. (2018). Automated Disease Detection in Agriculture. Comput-er Vision Journal, 25(4), 210-225.

[8] 7. Thompson, G., et al. (2021). Feature Extraction Techniques for Potato Disease Detec-tion. Pattern Recognition Journal, 35(1), 78-92.

[9] 8. Davis, H., & Roberts, J. (2023). Deep Learning Architectures for Disease Classifica-tion. Neural Networks Journal, 30(3), 180-195.

[10] 9. Johnson, E., et al. (2020). Ensemble Learning for Improved Disease Detection. Ma-chine Learning Journal, 18(4), 245-260.

[11] 10. Brown, D., & Patel, C. (2019). Addressing Data Scarcity in Potato Disease Classifica-tion: Techniques and Challenges. International Conference on Agricultural

Data Manage-ment and Applications, 2020.

[12] 11. Gupta, R., et al. (2021). Explainable AI Techniques for Enhancing Trust in Machine Learning Models for Potato Disease Detection. IEEE International Conference on Agricul-tural Technology, 2021.

[13] 12. Lee, S., et al. (2019). Integration of Spectral Imaging for Early Detection of Potato Diseases. Journal of Spectral Imaging, 5(2), 1-15.

[14] 13. Smith, A., et al. (2020). UAV-Based Remote Sensing for Potato Disease Monitoring: Challenges and Opportunities. International Journal of Remote Sensing Applications, 10(4), 321-336.

[15] 14. Johnson, E. (2024). Machine Learning in Agriculture: A Review. Annual Review of Agriculture, 40(4), 312-325.