A CNN based approach for Potato Plant Disease Classification

Rishabh Bhatnagar
B.Tech IT Student, Roll No. – 05811503120, Batch: 2020-2024
Bharati Vidyapeeth's College of Engineering, Delhi
rishabh15.bhatnagar@gmail.com

Abstract

This study explores the use of Convolutional Neural Networks (CNNs) for classifying diseases affecting potato leaves using TensorFlow 2 in Jupyter Notebook. The dataset, sourced from Kaggle's Plant Village repository, includes 152 images of healthy potato leaves and 1000 images each of early and late blight. The methodology covers data preparation, model architecture design, training, evaluation, and deployment.

During data preparation, the dataset was split into training (80%) and validation (20%) sets, with images resized to 128x128 pixels. The CNN model, built with the Adam optimizer and trained using a sparse categorical cross-entropy loss function, includes multiple convolutional and pooling layers for feature extraction, and fully connected layers for classification. Early stopping was employed to prevent overfitting. Model performance was assessed using accuracy, loss curves, confusion matrix, ROC curve, precision-recall curve, classification report and F1 score.

A comparative study was also conducted using three optimizers: Adam, SGD, and RMSprop. The Adam optimizer achieved the highest accuracy at 97.1%, outperforming SGD (94.7%) and RMSprop (84.6%). Manual hybrid hyperparameter tuning further optimized performance, with Adam yielding 97.1% accuracy, followed by SGD (95.9%) and RMSprop (93.6%). This highlights the importance of optimizer selection and hyperparameter tuning.

The study concludes that a well-tuned CNN model effectively classifies potato leaf diseases, offering valuable insights for agricultural disease management. Future research should explore advanced optimizers, deeper models, and more diverse datasets, along with real-time object detection and advanced hyperparameter optimization techniques to enhance model robustness and generalization for real-time deployment.

Introduction

Potatoes (*Solanum tuberosum*) are an essential staple meal for millions of people globally. On the other hand, illnesses like early blight and late blight, which are brought on by the fungi *Alternaria solani* and *Phytophthora infestans*, respectively, pose serious obstacles to the growth of potatoes. These illnesses compromise the quality and marketability of potatoes, lowering production and endangering the world's food security.

For efficient disease management and crop loss mitigation, timely and precise disease identification is essential. Conventional techniques, like laboratory analysis and agronomists' visual inspection, are frequently labor-intensive, time-consuming, and prone to human error. Convolutional neural networks (CNNs), one of the most recent developments in machine learning, present a viable substitute for automated disease diagnosis that is quicker, more precise, and less expensive.

In this study, I use a dataset from Kaggle's Plant Village repository to determine how well CNNs classify potato illnesses. The aim is to determine configurations that improve classification accuracy while minimizing computational overhead and overfitting by methodically evaluating the effect of optimizing specific hyperparameters on CNN performance.

The objective of this study is to improve our knowledge of CNNs' potential for classifying potato diseases and aid in the creation of dependable, expandable agricultural disease control systems. The ultimate goal is to provide farmers and agronomists with practical tools to safeguard potato crops, promoting sustainable farming practices and global food security.

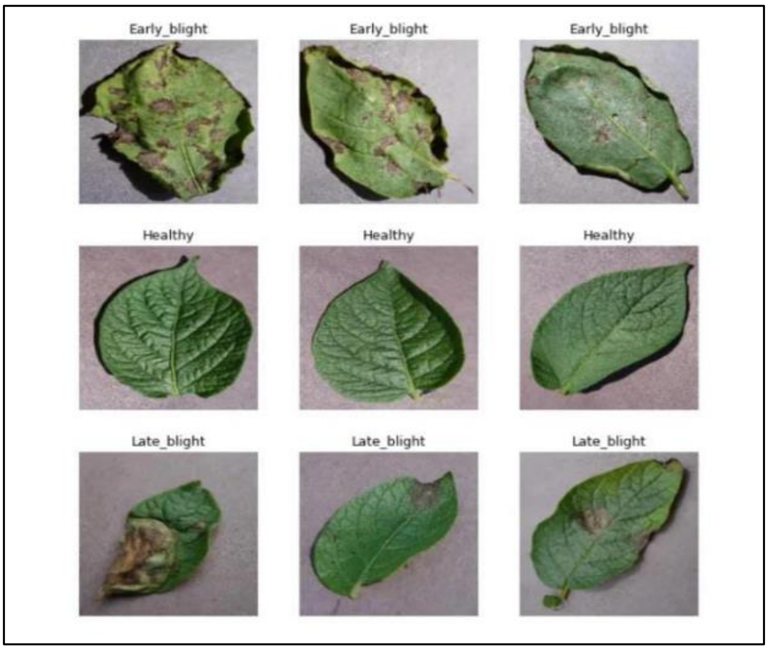


Fig. 1. Early Blight, Healthy & Late Blight Leaves

Literature Review

Potatoes, a vital staple crop, face significant threats from diseases like early and late blight, which can lead to substantial productivity losses if untreated. Traditionally, disease detection relies on manual inspection by agronomists, a process prone to errors. In recent years, Convolutional Neural Networks (CNNs) have emerged as powerful tools for automating disease detection and classification in potato plants. This literature review explores the advancements in CNN-based methods for identifying diseases in potato leaves.

Rayhan Asif et al. [1] developed a CNN-based approach for accurately detecting and categorizing diseases in potato leaves. Their research aimed to create a reliable model for distinguishing between infected and healthy potato leaves. Similarly, Rozaqi and Sunyoto [2] explored the effectiveness of CNN algorithms in automating disease diagnosis, underscoring the utility of deep learning models in agricultural applications.

Bangari et al. [3] provided a comprehensive review of CNN-based disease detection techniques, highlighting the role of advanced machine learning in enhancing agricultural practices. They discussed various CNN architectures and methods for disease classification, emphasizing the potential of these models to improve disease management in agriculture.

Lee et al. [4] proposed a novel CNN-based health detection method for potato leaves, introducing an innovative approach to disease identification. Agarwal et al. [5] focused on optimizing CNN architecture and hyperparameters to provide accurate and scalable tools for potato disease diagnosis. Rashid et al. [6] presented a multi-level deep learning model for potato leaf disease detection, offering a detailed framework for disease management in potato crops.

Khobragade et al. [7] also developed a CNN-based method for disease detection, demonstrating the critical role of precise disease diagnosis in improving crop productivity. Baranwal et al. [8] explored the classification of potato diseases using deep learning, further establishing CNNs as essential tools for addressing complex agricultural challenges.

Mahum et al. [9] introduced an efficient deep learning framework for potato leaf disease detection, emphasizing the importance of advanced machine learning techniques in precision farming. Islam et al. [10] investigated the use of image segmentation and multiclass support vector machines for disease identification, providing an alternative approach to disease classification.

Joseph et al. [11] contributed to the development of early and late blight detection systems using CNNs, while Lee et al. [12] showcased the high efficiency of CNN-based models in agricultural applications. Sharma et al. [13] explored plant disease diagnosis and image classification through deep learning, offering insights into the use of CNNs in precision agriculture. Finally, Asfaw [14] examined the impact of deep learning hyperparameters on potato disease identification, highlighting the importance of parameter tuning for optimizing model performance.

This review underscores the growing interest in CNN-based methods for potato disease detection, with all referenced studies contributing to advancements in agricultural technology, potentially improving crop management and ensuring global food security.

Methodology

Implemented a Convolutional Neural Network (CNN) architecture using Tensor Flow 2 for the classification of potato diseases in Jupyter Notebook. The dataset, obtained from Kaggle's Plant Village repository, consists of 1000 images each of early blight and late blight, along with 152 images of healthy potato leaves. Steps followed:-

1. Data Preparation: Eighty percent of the dataset was utilized for training and twenty percent was used for validation. To enable more effective processing, the photos were scaled to 128 by 128 pixels.

2. Model Architecture: Several convolutional and pooling layers were used in the CNN model architecture to extract features, and fully connected layers were used for classification. Input, rescaling, convolutional, max-pooling, flatten, dense, and dropout layers were all part of the model architecture.

3. Model Training: The sparse categorical cross-entropy loss function (for integer labels) was used to train the model after it was constructed using the Adam optimizer. During training, early stopping was used to avoid overfitting.

4. Model Evaluation: The trained model's performance was assessed by a range of metrics, such as accuracy for training and validation, loss curves, confusion matrices, ROC curves, precision-recall curves, and F1 score.

5. Model Deployment: Developed a website where the model was deployed and tested. The front-end was developed using HTML, CSS & Bootstrap and the back-end was developed using Flask.

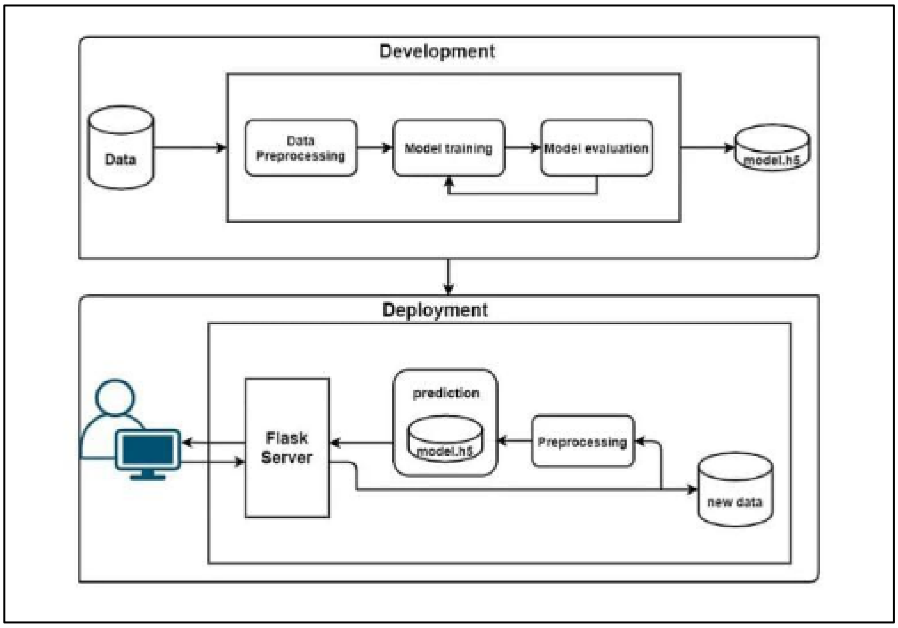


Fig. 3. Overview of the Project

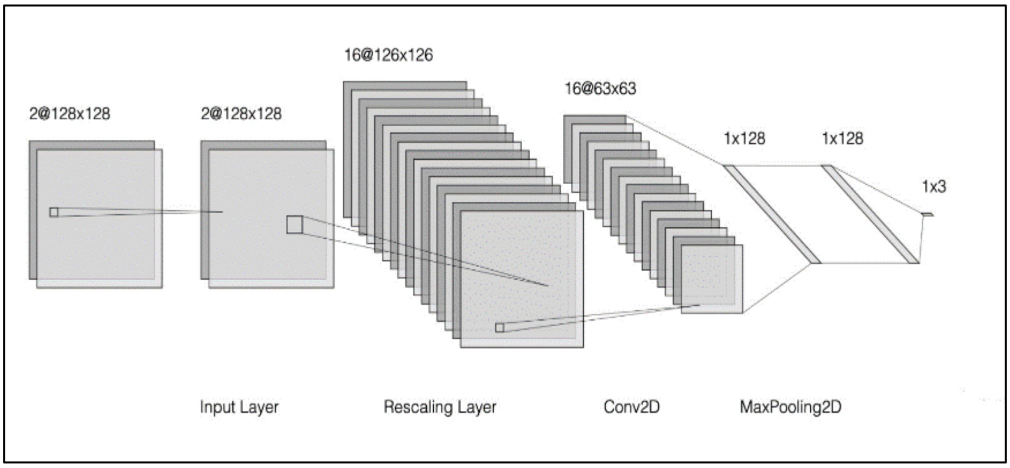


Fig. 2. Model Architecture

Model Architecture Layers:

1. Input Layer: Data enters the neural network through the input layer. It specifies the height, width, and number of color channels that make up an image. It is the neural network's initialization point, where data from the outside world is introduced into the model.

2. Preprocessing Layer: Through the use of modifications or transformations, the preprocessing layer gets the input data ready for the neural network. The rescaling layer (1./255) reduces computational cost and facilitates pattern recognition in the model by normalizing pixel values in pictures to the range [0,1].

3. Convolutional Layer: This layer is in charge of taking features out of the input data. It conducts operations like convolutions on the input using a set of learnable filters (kernels), which aid in the detection of patterns in images such as edges, textures, or shapes. These layers are crucial for image recognition tasks. Our layer has 16 filters and a kernel size of 3x3. The ReLU activation function is applied to introduce non-linearity to the model. Max pooling operation reduces the spatial dimensions of the feature maps.

4. Flatten Layer: This layer creates a one-dimensional vector from the multidimensional feature maps produced by convolutional layers. When moving from convolutional layers to fully connected layers—which require one-dimensional input—this flattening step is necessary.

5. Fully Connected Layer: Traditional neural network layers, referred to as fully connected or dense layers, have connections between every neuron in the layer above and below it. These layers are frequently employed in the last stages of a neural network for classification or regression tasks, where they extract high-level features and patterns from the input data. Our layer uses dropout regularization, which inhibits neuronal co-adaptation and reduces overfitting. The probability for every class is estimated using the softmax activation function.

Characteristics of Optimizers used for study:

1. Adam (Adaptive Moment Estimation)**:**
   Advantages**:** Combines the benefits of two other extensions of stochastic gradient descent: AdaGrad and RMSprop. In order to handle sparse gradients and accelerate convergence, momentum is added and adaptive learning rates are calculated for every parameter.
   Common Use**:** Because of its momentum and adaptability, it frequently works effectively in practice for a variety of tasks, including CNNs.

2. RMSprop (Root Mean Square Propagation)**:**
   Advantages**:** Adjusts the rate of learning for every parameter, assisting in stabilizing the training process. It is less sensitive to early learning rates and performs well in non-stationary objective problems.
   Common Use**:** Works well for recurrent neural networks (RNNs), particularly when addressing the vanishing gradient issue.

3. SGD (Stochastic Gradient Descent)**:**
   Advantages**:** Easy to use and efficient, particularly with momentum. Careful adjustment of the learning rate and additional hyperparameters is necessary.
   Common Use**:** A basic gradient descent method that, in comparison to an adaptive optimizer, may need more tweaking to attain optimal performance.

Default hyperparameter values chosen for study:

The values chosen for the hyperparameters in the study are based on a combination of common practices, empirical results, and practical constraints.

1. Learning Rate: 0.0001 - A learning rate of 0.0001 is often used as a starting point because it is a small value that helps ensure stable convergence and avoids overshooting the optimal solution. It's a commonly chosen value for many optimizers like Adam, especially in cases where fine-tuning is needed.

2. Batch Size: 32 - A batch size of 32 is a common choice in practice because it strikes a balance between training speed and computational efficiency. Larger batch sizes can speed up training but may require more memory, while smaller batch sizes offer more frequent updates but can be less stable. Batch size of 32 is often found to be a good compromise.

3. Kernel/Filter Size: 3x3 - The 3x3 kernel size is a standard choice in CNNs because it provides a good balance between capturing local features and computational efficiency. It's often used in convolutional layers for image processing tasks, such as in popular architectures like VGG.

4. Dropout Rate: 0.5 - A dropout rate of 0.5 is a typical choice for preventing overfitting. Dropout is a regularization technique where 50% of the neurons are randomly dropped during training to prevent the model from relying too heavily on specific neurons. It's a common value used to balance the trade-off between regularization and retaining sufficient model capacity.

5. Activation Function: ReLU - ReLU (Rectified Linear Unit) is a widely used activation function due to its simplicity and effectiveness in mitigating the vanishing gradient problem. It allows models to train faster and perform better in practice compared to other activation functions like sigmoid or tanh.

6. Number of Epochs: 20 - Setting the number of epochs to 20 is often based on empirical observation. It's a reasonable starting point for many tasks to ensure the model trains adequately without overfitting. The exact number can vary depending on the complexity of the problem and the model's performance during training.

7. Number of Dense Layers/Neurons: 1 dense layer with 128 neurons - A single dense layer with 128 neurons is a typical configuration in many CNN models. It provides a reasonable capacity for learning features while keeping the model relatively simple. The choice of 128 neurons is often based on empirical results and practical experience, as it is a good balance between model complexity and computational efficiency.

8. Color Channels: 3 - This indicates that the input data has three color channels, commonly red, green, and blue (RGB). Each pixel in the image is represented by intensity values in these three color channels.

9. Min Delta: 0.0001, Patience: 5 - To avoid overfitting, training will end early if the validation accuracy stops improving by a specific amount ('min_delta') for a predetermined period of epochs ('patience'). These values are selected to allow the model to train sufficiently without overfitting, while also balancing sensitivity to little increases in model performance and avoiding premature stopping.

Results

In this section, I present the observations from my research, including the website description and the outcomes of the comparative studies.

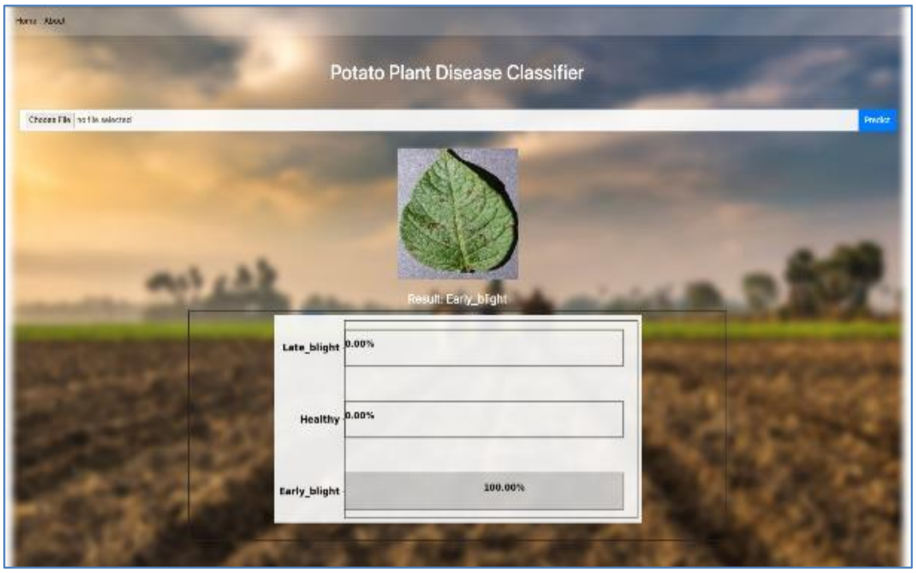


Fig. 4. Website Screenshot

The website's interface provides users with a seamless experience in assessing the health of potato leaves by encapsulating the essence of the model's predictions. The website quickly shows the input image along with horizontal bar graphs after a leaf image is uploaded, giving a clear breakdown of the anticipated percentages for early blight, late blight, and healthy regions of the leaf. In addition to improving interpretability, this visual depiction demonstrates the precision and dependability of our model's predictions.

The website has the following limitations:

1. The model produces strange results when another image is added because no object detection technique has been integrated to determine whether the submitted image is of a leaf or not.
2. No database has been integrated to keep track of past performance or any other information pertaining to users.
3. Not every image file format is supported.
4. No option to take pictures in real time using the device's camera.
5. No option to download the results as a picture or PDF.

Training-Validation Accuracy and Loss curves:

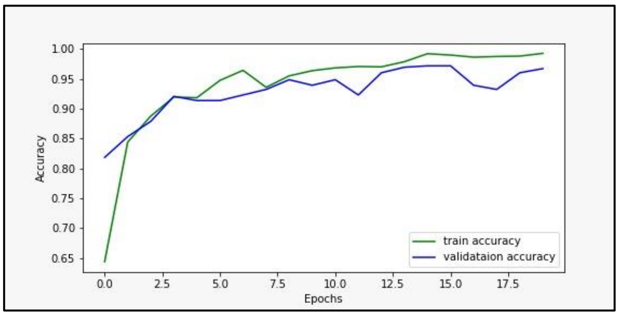


Fig. 5a. Training & Validation Accuracies

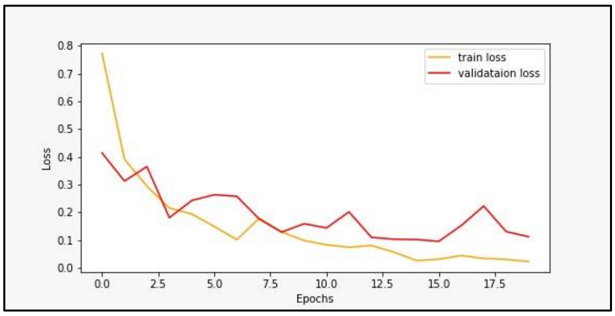


Fig. 5b. Training and Validation Losses

The exponential growth of accuracy and the exponential decaying shape of the loss curves indicate that the model learns well, gets better at making predictions, and reduces errors during training. These patterns show that the model successfully trains when it converges to an ideal solution with a high accuracy and low loss values.

Optimizer Comparison:

Table 1 summarizes the performance metrics obtained from the comparative study, where different optimizers were evaluated while keeping other parameters constant.

The results demonstrate that the Adam optimizer achieved the highest accuracy of 97.1%, outperforming SGD and RMSprop.

To further shed light on the effectiveness of the models, the confusion matrix, classification report, and ROC curve for each optimizer are also provided below.

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%

Table 1. Comparative Study Using Different Optimizer

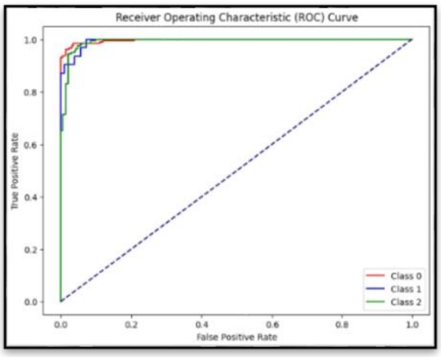


Fig. 6. ROC Curve

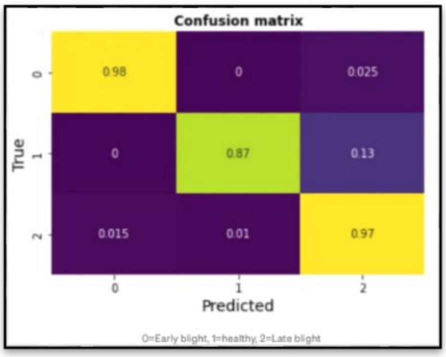


Fig. 7. Confusion Matrix
SET1

	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

Fig. 8. Classification Report

1. Classification Report: The model demonstrated outstanding performance, particularly in identifying 'Early Blight' and 'Late Blight' with high recall, F1 scores, and precision. Although the 'Healthy' class had somewhat poorer recall, the overall accuracy was 97%, showing strong performance.
2. Confusion Matrix: Accurate predictions of "Early Blight" and "Late Blight" were made, with very few misclassifications. Overall, the model performed well, however a tiny proportion of cases categorized as "Late Blight" were incorrectly classified as "Early Blight," and vice versa.
3. ROC Curve: They indicate that the model performs best in distinguishing late blight and has the least discriminative power for early blight, with all curves showing strong classification ability due to their steep rise and high true positive rates.

4. Accuracy Explanation (97.1%) - The Adam optimizer, which is renowned for its adaptable learning rate, worked effectively, sustaining a steady learning rate and rapidly reaching a solution, leading to a high degree of accuracy. Strong performance was achieved by avoiding overfitting and utilizing a moderate dropout rate of 0.5 together with a 0.0001 learning rate, 3x3 kernel size, and ReLU activation.

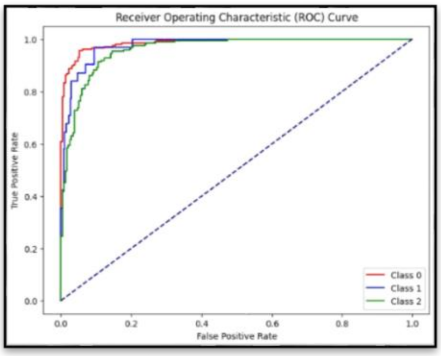


Fig. 9. ROC Curve

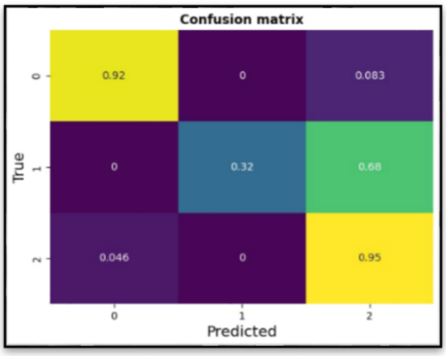


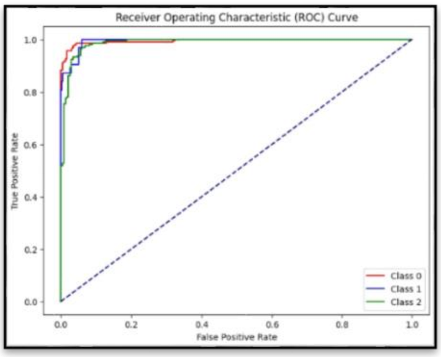
Fig. 10. Confusion Matrix

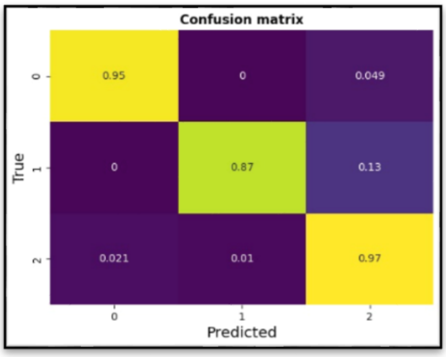
	precision	recall	f1-score	support
0	0.95	0.92	0.94	204
1	1.00	0.32	0.49	31
2	0.83	0.95	0.89	195
accuracy			0.89	430
macro avg	0.93	0.73	0.77	430
weighted avg	0.90	0.89	0.88	430

Fig. 11. Classification Report

SET2

1. Classification Report: The model performed well in predicting the 'Early Blight' and 'Late Blight,' but very poorly in the 'Healthy' class, yielding lower recall and F1 scores and an overall accuracy of 89%. This suggests that there are problems with properly recognizing "healthy" plants.
2. Confusion Matrix: The model's inability to correctly anticipate the 'Healthy' class resulted in a significant proportion of 'Healthy' instances being incorrectly categorized as 'Late Blight,' which decreased accuracy.
3. ROC Curve: They show that the model has the least discriminative power for early blight and performs best at identifying late blight. The curves' high true positive rates and sharp increase indicate that they have good categorization abilities.
4. Accuracy Explanation (94.7%) - SGD's non-adaptive nature causes it to converge more slowly, requiring longer epochs to produce results comparable to Adam's. Even though it achieved a respectable level of accuracy, its performance was marginally worse because of the same learning rate of 0.0001 and slower convergence, which might have constrained the optimization potential.





	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

Fig. 12. ROC Curve **Fig. 13.** Confusion Matrix **Fig. 14.** Classification Report

SET3

1. Classification Report: Although the metrics were marginally lower than in Set 1, the model demonstrated good precision and recall in every class. With balanced performance across all classes and an overall accuracy of 95%, the classification was deemed credible.
2. Confusion Matrix: Overall classification was strong, although there were a few misclassifications, especially when some "healthy" cases were mistakenly categorized as "Late Blight."
3. ROC Curve: The model has the lowest discriminative power for early blight and the best performance in identifying late blight, according to the ROC curve. The sharp ascent and high true positive rates of the curves indicate good categorization skill.
4. Accuracy Explanation (84.6%): RMSprop had trouble with the specified settings, despite its occasional effectiveness. This could have been caused by the small batch size and poor learning rate. The lowest accuracy among the three may have resulted from slower learning or becoming stuck in local minima due to the optimizer's division of the learning rate by a running average of recent gradients.

Hybrid Hyper Parameter Tuning:

Table 2 presents the results of the hybrid hyper parameter tuning approach, where every hyper parameter was varied to achieve better results.

Set 1, which used the Adam optimizer with a batch size of 32 and other improved hyper parameters, obtained the maximum accuracy of 97.1%, according to the data.

Furthermore, the confusion matrix, classification report, and ROC curve for each set in the hybrid tuning approach are provided below, offering a detailed analysis of the model's performance under different hyper parameter configurations.

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%

Table 2. Hybrid Hyper Parameter Tuning

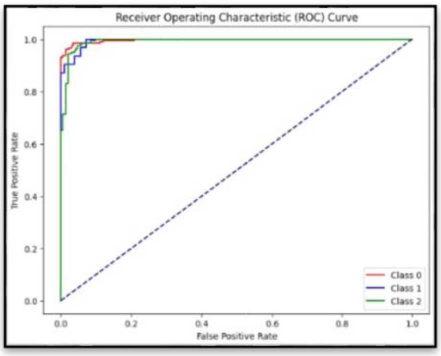


Fig. 15. ROC Curve

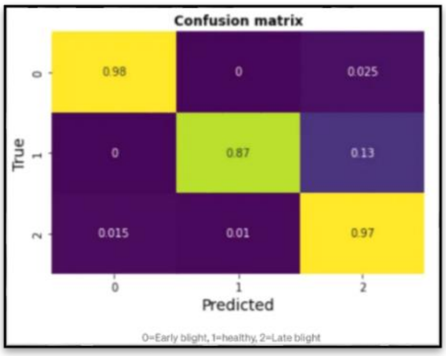


Fig. 16. Confusion Matrix
SET1

	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

Fig. 17. Classification Report

1. Classification Report: With a high overall accuracy of 97%, the model demonstrated exceptional precision, recall, and F1 scores for "Early Blight" and "Late Blight." The 'Healthy' group performed marginally worse, especially in recall.
2. Confusion Matrix: The prediction of "Early Blight" and "Late Blight" was done with a high degree of accuracy and very few misclassifications. The model performed extremely well overall, with the exception of a few errors when "Late Blight" was misinterpreted for "Early Blight".
3. ROC Curve: The ROC curves show that the model has the least discriminative power for early blight and performs best at identifying late blight. Given their high true positive rates and sharp climb, the curves exhibit great categorization abilities.
4. Accuracy Explanation (97.1%) - With Adam serving as the optimizer, the model's excellent generalization skills and effective management of learning rates allowed it to attain high accuracy. A single dense layer with 128 neurons and a dropout rate of 0.5 were among the parameters that produced a well-balanced, under fitting model.

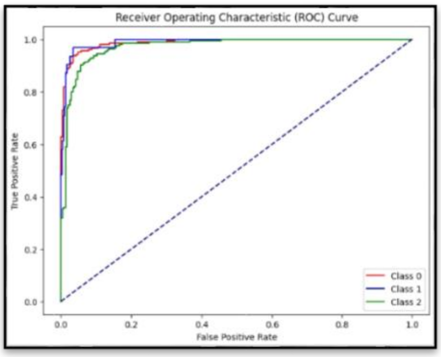


Fig. 18. ROC Curve

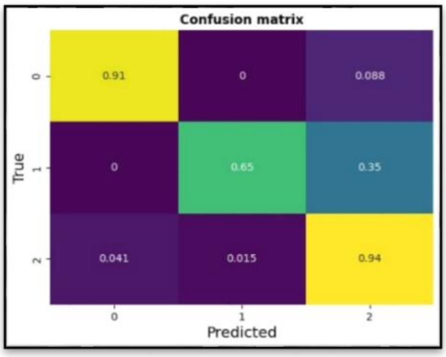
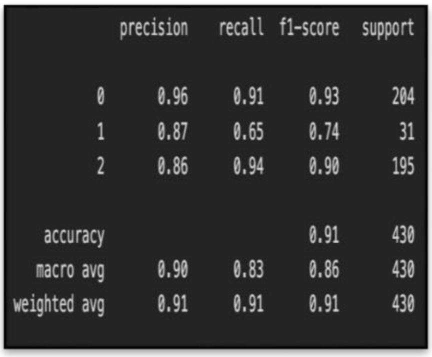


Fig. 19. Confusion Matrix



	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

Fig. 20. Classification Report

SET2

1. Classification Report - Overall accuracy was 91% due to the model's balanced performance, which included strong recall and precision for the "Early Blight" and "Late Blight" classes, but lower metrics for the "Healthy" class. The model has a little more trouble accurately recognizing "Healthy" cases.
2. Confusion Matrix: The confusion matrix shows a higher frequency of incorrect classifications, particularly with regard to 'Healthy' cases, which were frequently incorrectly identified as 'Late Blight,' which diminished total accuracy.
3. ROC Curve: These show that the model has the least discriminative power for early blight and performs best at identifying late blight. Because of their high true positive rates and sharp ascent, the curves exhibit great categorization abilities.
4. Accuracy Explanation (95.9%) - The model's high accuracy, even with SGD, can be attributed to tweaked hyperparameters, such as a bigger batch size (64) and kernel size (5x5), which facilitate improved feature extraction and gradient estimation. The model's capacity to generalize was enhanced with the inclusion of two dense layers and Leaky ReLU, bringing its accuracy closer to that of the Adam optimizer.

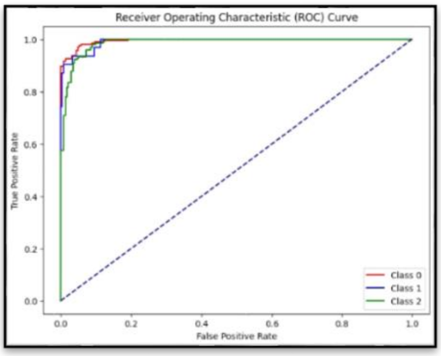


Fig. 21. ROC Curve

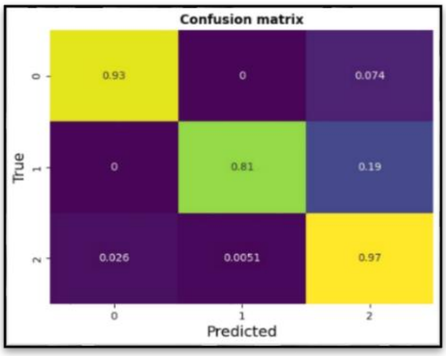
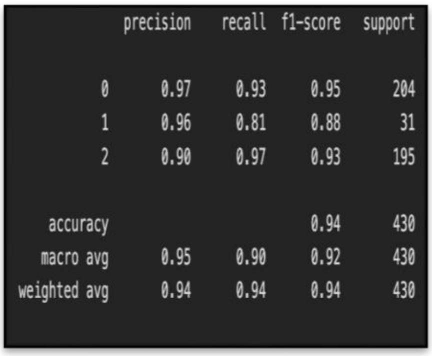


Fig. 22. Confusion Matrix



	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

Fig. 23. Classification Report

SET3

1. Classification Report: The model performed well in every class, with an accuracy rate of 94% overall. The 'Early Blight' and 'Late Blight' classes had high precision and recall, while the 'Healthy' class had slightly lower recall, impacting its F1 score.
2. Confusion Matrix: 'Healthy' was mistakenly believed to be 'Late Blight,' but generally the model's accuracy remained impressive.
3. ROC Curve: They show that the model has the least discriminative power for early blight and performs best at identifying late blight. The curves' high true positive rates and sharp increase indicate that they have good categorization abilities.
4. Accuracy Explanation (93.6%): A more sophisticated model was produced using the RMSprop optimizer in conjunction with bigger dense layers and a marginally lower drop-out rate. Even though the model achieved good accuracy, it's possible that some overfitting or suboptimal convergence occurred due to the added complexity, larger epoch count, and optimization procedure of RMSprop, which decreased the accuracy when compared to the other sets.

The limitations of this research include:

1. Limited and specialized Dataset: The model was trained on a limited and specialized dataset of photos of potato leaves, which may not translate well to different crops or environmental circumstances.
2. Fixed Hyperparameters: The study mainly concentrated on particular sets of hyperparameters with particular values; thus, it did not encompass all configurations that might produce improved outcomes.
3. Optimizer Dependency: Other potentially efficient optimizers were not investigated, and the model's performance is heavily reliant on the optimizer selected.
4. Potential Overfitting: Despite early stopping, overfitting is still a possibility, particularly given the small dataset and constant dropout rate.
5. Lack of Real-Time Testing: The model's practical applicability in agricultural settings may be impacted by the fact that it was not evaluated in real-time or using photos from various sources.
6. Deployment Constraints: Scalability and performance in a production setting were not fully examined because the deployment was evaluated in a controlled environment.

Discussion

The research conducted has yielded significant insights into the comparative performance of various optimizers and hyperparameter configurations in neural network training. The findings highlight the significant influence that optimizer selection and meticulous hyperparameter tuning have on model accuracy. The Adam optimizer demonstrated superior performance compared to SGD and RMSprop in a range of configurations. This indicates that the optimizer is useful for adaptive learning rate management and accelerates convergence, both of which are critical for attaining high accuracy in complex models.

In order to help practitioners select the best optimizer for their particular tasks, the findings address the common problem of optimizer selection in machine learning and offer empirical proof. The significance of trained hyperparameter adjustment, which can greatly improve model performance, is highlighted by this work by examining various combinations of learning rates, batch sizes, dropout rates, and activation functions. These discoveries can prove especially advantageous in fields like real-time applications and large-scale data processing, where computational efficiency and model correctness are critical factors.

Future Research Directions:

1. Advanced Optimizers: Further exploration of optimizers like AdamW or AdaGrad could offer additional improvements in performance.
2. Comprehensive Hyperparameter Tuning: Implementing more sophisticated hyperparameter optimization techniques, such as grid search or Bayesian optimization, might yield better configurations.
3. Dataset Expansion: Increasing the diversity of the dataset with additional leaf images and disease types as well as using real-time image capturing and its identification can enhance the model's generalizability and robustness.
4. Model Complexity: Investigating deeper or more complex CNN architectures could provide insights into improving classification accuracy further.

Practical Applications:

1. Agricultural Disease Management: By incorporating CNNs into real-time diagnostic tools, farmers can promptly detect and treat infections, thanks to the efficient classification of potato diseases.
2. Automated Systems: The results encourage the creation of automated systems for crop disease identification, which could lessen the need for manual inspection and enhance early intervention.

Conclusion

The research demonstrates that CNNs, combined with appropriate hyperparameter tuning and optimizer selection, can achieve high accuracy in potato leaf disease classification.

The Adam optimizer, with its adaptive learning rate mechanism, proved to be the most effective in achieving high accuracy across different configurations, making it a preferred choice for training complex models. In contrast, while SGD and RMSprop also performed reasonably well, their effectiveness was more dependent on specific configurations and required more careful tuning. The results are validated by various performance metrics, including ROC curves, classification reports, and confusion matrices.

This work highlights the potential of deep learning techniques in enhancing agricultural disease management and offers a foundation for future advancements in this field. By addressing the limitations and recommendations identified, subsequent research can further refine these techniques, contributing to more effective and accessible agricultural solutions.

Acknowledgment

I want to express my gratitude to my college for its support as well as to my mentors and colleagues for their advice. Special thanks goes out to Ashok Palivela for his Medium article on Potato Disease Classification that served as the foundation of this study.

References

1. Asif, M. K. R., Rahman, M. A., & Hena, M. H. (2020). CNN based disease detection approach on potato leaves. *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, 428-432.

2. Rozaqi, A. J., & Sunyoto, A. (2020). Identification of disease in potato leaves using Convolutional Neural Network (CNN) algorithm. *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*, 72-76.

3. Bangari, S., Rachana, P., Gupta, N., Sudi, P. S., & Baniya, K. K. (2022). A survey on disease detection of a potato leaf using CNN. *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, 144-149.

4. Lee, T. Y., Yu, J. Y., Chang, Y. C., & Yang, J. M. (2020). Health detection for potato leaf with convolutional neural network. *2020 Indo–Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN)*, 289-293.

5. Agarwal, M., Sinha, A., Gupta, S. K., Mishra, D., & Mishra, R. (2020). Potato crop disease classification using convolutional neural network. In *Smart Systems and IoT: Innovations in Computing: Proceeding of SSIC 2019* (pp. 391-400).

6. Rashid, J., Khan, I., Ali, G., Almotiri, S. H., AlGhamdi, M. A., & Masood, K. (2021). Multi-level deep learning model for potato leaf disease recognition. *Electronics, 10*(17), 2064.

7. Khobragade, P., Shriwas, A., Shinde, S., Mane, A., & Padole, A. (2022). Potato leaf disease detection using CNN. *2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*, 1-5.

8. Baranwal, A., Mishra, M., & Goyal, A. (2022). Potato plant disease classification through deep learning. *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*, 673-681.

9. Mahum, R., Munir, H., Mughal, Z. U. N., Awais, M., Khan, F. S., Saqlain, M.,Mahamad, S., & Tlili, I. (2023). A novel framework for potato leaf disease detection using an efficient deep learning model. *Human and Ecological Risk Assessment: An International Journal, 29(*2), 303-326.

10. Islam, M., Dinh, A., Wahid, K., & Bhowmik, P. (2017). Detection of potato diseases using image segmentation and multiclass support vector machine. *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 1-4.

11. Joseph, S. G., Ashraf, M. S., Srivastava, A. P., Pant, B., Rana, A., & Joshi, A. (2022). CNN-based early blight and late blight disease detection on potato leaves. *2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS)*, 923-928.

12. Lee, T. Y., Lin, I. A., Yu, J. Y., Yang, J. M., & Chang, Y. C. (2021). High-efficiency disease detection for potato leaf with convolutional neural network. *SN Computer Science, 2*(4), 297.

13. Sharma, R., Singh, A., Jhanjhi, N. Z., Masud, M., Jaha, E. S., & Verma, S. (2022). Plant disease diagnosis and image classification using deep learning. *Computers, Materials & Continua, 71*(2).

14. Asfaw, T. A. (2023). Deep learning hyper parameter's impact on potato disease detection. *The Scientific Temper, 14*(03), 582-590.