Journal of Innovative Image Processing (ISSN: 2582-4252)
www.irojournals.com/iroiip/

Agricultural Application of Convolutional Neural Networks: A Case Study on Potato Plant Disease Detection Using Keras Image Generator and Data Augmentation Techniques

Swathi Ganesan[1], Lakmali Karunarathne[2], Sangita Pokhrel[3], Nalinda Somasiri[4], Ganapathy Kumar[5]

[1-4]Department of Computer Science and Data Science, York St John University, United Kingdom

[5]Senior Consultant, IT sector, London, United Kingdom

Email: [1]s.ganesan@yorksj.ac.uk, [2]l.karunarathne@yorksj.ac.uk, [3]s.pohrel@yorksj.ac.uk, [4]n.somasiri@yorksj.ac.uk, [5]ganapathy.kumar@gmail.com

Abstract

Crop yields are severely impacted by plant diseases, leading to significant economic consequences. This study presents a plant disease prediction model that utilizes Convolutional Neural Networks (CNNs) and the Keras image augmentation technique. The CNN architecture includes multiple convolutional and pooling layers, as well as fully connected layers. Model training employs the Adam optimiser and categorical cross-entropy loss function, using a dataset of plant leaf images labelled with corresponding diseases for validation. After training the model with 10 epochs and a batch size of 32, an accuracy of 97% was achieved with a loss of 0.11. Validation accuracy and loss were 91% and 0.20, respectively. The Keras image augmentation technique was also evaluated for its effectiveness in generating new images from existing ones, which were used to test the model's ability to generalise when exposed to unseen data. The accuracy and loss on the test images were 95% and 0.25 and for augmented images were 94% and 0.22, respectively, demonstrating the model's potential for use in plant disease management as a diagnostic tool for farmers. This study is unique in combining CNN and Keras Image Generator for the detection of leaf diseases, and the results suggest that the proposed model could be useful for improving crop yields and farmers' income.

Journal of Innovative Image Processing, December 2024, Volume 6, Issue 4, Pages 433-455

DOI: https://doi.org/10.36548/jiip.2024.4.007
Received: 16.11.2024, received in revised form: 27.12.2024, accepted: 11.01.2025, published: 23.01.2025

Keywords: Machine Learning, Deep Learning, bacterial plant disease, Keras Image Generator, Convolutional Neural Network, Data Augmentation.

1. Introduction

Plant diseases and their effects on crop yields and quality are a major worry for farmers and agricultural specialists across the world. Numerous things, including bacteria, fungi, viruses, and environmental stressors, can cause these disorders. Plant diseases such as powdery mildew, leaf spot, blight, rust, and wilt can cause significant damage to crop. These disease's effects on crop productivity might result in large financial losses for farmers and agricultural enterprises [1]. Furthermore, plant disease might seriously harm the ecosystem since treating them could involve using dangerous pesticides and other chemicals. Furthermore, plant diseases may have a negative effect on the environment since treating them may necessitate the use of dangerous pesticides and other chemicals.

To address the issue of identifying and treating plant disease, researchers have been using cutting-edge technologies such as Artificial Intelligence (AI), Machine Learning, and CNNs—a type of AI technology that has shown promise in plant disease prediction. Through the use of CNNs to analyze plant images and patterns, scientists can identify early indicators of disease and take appropriate measures to prevent it from spreading. One way to use CNNs in conjunction with other methods to forecast plant diseases is data augmentation. It creates fresh training data by applying various effects to old images, such as rotation or flipping [2]. Due to the wider variety of training data this method offers, CNN performance can be enhanced. It is challenging for potato growers to prevent diseases from spreading throughout their harvest, which might lead to significant financial losses. The study focuses on predicting disease of potato leaves. Based on Integrated pest management or IPM about potatoes, some of the common diseases that affect the potato plants are shown in Figure 1 these include blackleg and soft rot, pink rot, Blackheart disease, Septoria leaf spot, Late blight, Early blight, frequent scab, Black scurf/canker, and viral infections.

Figure 1. Common Diseases-Potato and Potato Leaf

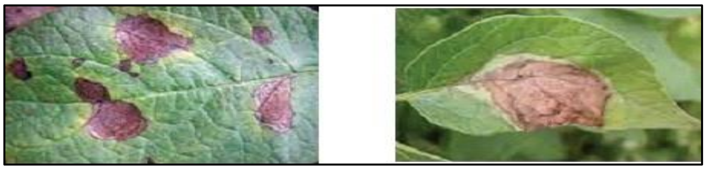


Figure 2. Early Blight (Left) and Late Blight (Right)

The Early Blight and Late Blight shown in Figure 2 are the two most prevalent infections. Different bacteria cause both early and late blight, but farmers may prevent a great deal of waste and financial loss if they can detect the disease early and treat it effectively. Since there are certain changes in the treatments for early and late blight, it is imperative to accurately identify the type of disease that is afflicting that potato plant.

The main goal of this research is to assess the effectiveness and precision of integrating Convolution Neural Networks (CNNs) with the Keras Image Generator method for plant disease detection. The study aims to advance the field of plant disease prediction by evaluating the possible advantages of combining CNNs with Keras Image Generator for precise and effective disease management and diagnosis.

2. Literature Review

Global agricultural production and food security are seriously threatened by plant diseases. Identifying plant diseases quickly and accurately is essential to putting management plans into action. Researchers have created a number of techniques and tools for the investigation of plant diseases throughout time. The purpose of this review of the literature is to give a broad overview of the body of knowledge on plant disease analysis, covering developing technologies, diagnostic methods, and approaches involving remote sensing.

Islam et al. [3] offer a technique combining machine learning and image analysis to precisely diagnose potato leaf diseases. In order to increase food security and sustainable agriculture, this technology aims to simultaneously identify plant and their physical characteristics. Traditional methods of identifying plant diseases, such manual interpretation, require a great deal of time, effort, and skill. According to this study, "Plant Village" is an automated technique that uses support vector machines to identify healthy and diseased potato plant leaves with a 95% accuracy rate based on images of the leaves. Because the approach is easy to use and successful at recognizing critical potato diseases including Early and Late Blight, farmers will be able to identify infections more promptly and effectively.

Techniques for evaluating leaf texture patterns and detecting plant diseases are covered by Waghmare et al [4]. Plant diseases have a negative effect on agricultural output and result in large economic losses [4]. As a result, the necessity for systems that can identify diseases in plant leaves has grown, especially when it comes to overseeing vast agricultural areas. According to the study, most plant diseases may be identified using the traits of the leaf. In order to identify the afflicted part of the leaf, image segmentation and a high-pass filter are used in the experiment described in the study to detect grape leaf diseases. A multiclass support vector machine (SVM) is then used to assess the leaf's textural pattern in order to classify it as healthy or diseased. The research focuses on black rot and downy mildew, two prevalent grape diseases. With a 96.6% forecast accuracy rate, the approach may give farmers rapid access to professional guidance. The authors of this study [5] employed methods for identifying plant diseases in 2018 based on images of leaves and saw encouraging outcomes. The work creates databases of healthy and disease leaves using the random forest approach. The dataset is identified, features are extracted, classifiers are developed, and healthy and disease leaves are classified as part of the study process. The model uses a gradient pointed towards the histogram to extract image attributes. Using 160 images of papaya leaves, the model in this study was trained to an accuracy of around 70%. Adding more local features with global characteristics, such as Speeded Up Robust Features and Bag of Visual Word paired with Scale Invariant Feature Transform Bag of Visual Word, might increase the accuracy.

The objective of Singh et al. [6] was the prompt and precise identification and classification of plant diseases. It took images using a digital camera and processed the images to extract important characteristics. They employed a vision-based detection technique that segmented data using the K-means clustering algorithm. The program finds the greatest number of green-colored pixels, masks the green pixels using Otsu's technique, and extracts features using the Color Co-occurrence Method (CCM). Neural networks were employed in a different study [7] to automatically identify leaf diseases, and it attained an overall accuracy of 94% as opposed to 89.5% in the prior study. Beans, Phaseolus vulgaris, and tea, Camellia asemia, are the study's main topics. A color conversion technique is utilized to turn the plant images into a monochrome grayscale image. K-means clustering is then employed for segmentation, and feature extraction is carried out using a color co-occurrence approach. For classification, a backpropagation neural network is employed. This research study report highlights the difficulties with using SVM if the training data is not linearly separable and examines various approaches to classifying plant leaf diseases. It concludes that the k-nearest-neighbor method is the most suitable and straightforward for class prediction based on test data [8][24]. The color transformation of the input RGB image, masking and deleting green pixels, segmenting the image, and finishing the segmentation are the four primary stages specified in their processing technique. Similarly, the study [12] focused on developing CNN algorithm for the detection of diseases and fertiliser deficiencies in paddy crops image detection. The result produced an average accuracy of 99% accuracy training. This led to successful predictions through the trained model.

A method for recognizing and classifying plant diseases utilizing machine learning techniques with image processing and colour concurrence using neural networks and other ML algorithms was reported in the articles [14],[15]. The paper focuses on using convolutional neural networks (CNNs)-based deep learning techniques as an alternative approach to classify plant diseases. In order to classify the plant leaf diseases, the article [16] reviews the most recent CNN networks. Additionally, [17],[18] give a summary of the DL ideas that are employed in plant disease classification. It also covered the difficulties and solutions associated with diagnosing plant diseases using CNNs and SVM, as well as the direction this area may go in the future. CNN is utilized not just for image classification but also for test analysis [19], disease diagnosis through image classification [13], sentiment analysis of people's actual sentiments and emotions [26], and product evaluations in e-commerce, software quality detection in industry sectors.

3. Experimental Study

For traditional machine learning and statistical methods, it is necessary to select the features manually which can lead to underperforming results. To overcome this, deep learning methods were created to diagnose different plant diseases using large sets of data [20]. We have proposed that CNN to classify the leaves of the potato plant into three classes healthy, early blight, and late blight diseases.

a) Plant Disease Classification System for Potato Plant Leaf

To identify and classify different potato diseases, the CNN algorithm was run through the potato disease classification system. A healthy and non-healthy potato leaf is sent into the CNN layers of the system as input for feature learning. Convolution RELU activation is the first operation in the CNN layers, which are then followed by Max Pooling. To identify and draw attention to the salient characteristics of the diseased leaf, these steps are performed several times. The output is then sent through a fully connected layer for ultimate classification following the feature learning procedure. The system can precisely identify and classify different potato diseases using CNN and completely linked layers, giving growers and producers of potatoes a useful tool.

b) Disease Focused

There are several diseases to be aware of that could affect potatoes. Some of them are Alternaria, black dot, blackleg, and bacterial soft rot, late blight, early blight, ring rot [21], and so on.

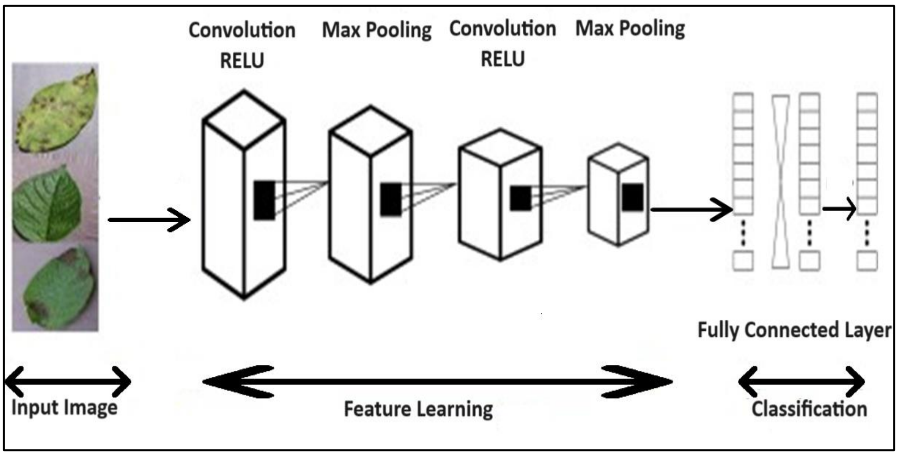


Figure 3. Block Diagram of Potato Plant Disease Detection and Classification System

The Oomycete Phytophthora is the source of late blight, and it grows best in damp, colder climates. Alternaria, the fungus that causes early blight, is more common in warmer climates. Whereas irregularly shaped lesions, plant rot, and a white cottony growth on the plant are signs of late blight, early blight is characterised by circular dark brown rings or lesions on older leaves. The plants are not greatly affected by early blight, but they are severely damaged by late blight [9]. While late blight spreads under chilly, damp circumstances, early blight is caused by persistent wetness.

The Potato Disease Detection and Classification system's fundamental architecture is depicted in the block diagram shown in Figure 3. The graphic shows how information moves through the system in a sequential fashion, beginning with input images of potato leaves. The feature learning step, which uses several layers to extract significant characteristics, is subsequently applied to these input images. Convolutional layers, activation functions (ReLU), and max pooling layers are all used in this feature learning process to jointly identify and extract significant patterns and features from the input images. Following feature extraction, the features move on to the classification stage, where a fully connected layer applies the learned features to the appropriate disease class labels. Based on the input images, the system can efficiently identify and classify potato plant diseases by adhering to this systematic approach. Figure 3 depicts the block diagram of potato plant disease detection and classification system.

c) Convolution Neural Network

Specifically created for image and voice recognition applications, Convolutional Neural Networks, often known as CNNs, are one kind of neural network. CNNs are ideal for these applications because of their convolutional layer, which helps to simplify images while maintaining their essential information. An illustration of the convolutional layer formula in a CNN is as follows [22]:

$$(X * W) + b = Z(i, j)$$

The formula denotes the following: $\mathrm{Z}(\mathrm{i}, \mathrm{j})$ is the output feature map at position (i, j); X is the input image or feature map from the previous layer; W is the filter weights; and b is the bias factor. This formula plays a crucial role during the training phase of CNNs. In training, the convolutional layers' weights (W) and biases (b) are learned and adjusted through the process of backpropagation. The forward pass of the network involves convolving the input image with the learned filters (W), applying the bias term (b), and then applying a non-linear activation function such as ReLU (Rectified Linear Unit) to introduce non-linearity into the model.

This convolutional operation is performed repeatedly across multiple layers in the CNN, enabling the network to learn increasingly abstract and complex features of the input data. The learned filters capture various levels of detail, from simple edges in the initial layers to complex textures and shapes in deeper layers. The combination of convolutional operations, bias adjustments, and non-linear activations allows CNNs to excel in tasks requiring image and voice recognition.

This formula is used implicitly throughout the training phase of this study. During training, the convolutional layers' weights (W) and biases (b) are discovered and modified using backpropagation. In order to add non-linearity to the model, the network's forward pass convolves the input image using the learnt filters (W), applies the bias term (b), and applies a non-linear activation function like ReLU.

The model architecture may be explained as shown in Figure 4. We feed the network with every pixel in an image to analyse it. For a 256x256x3 image (256 pixels by 256 pixels with three color channels: red, green, and blue), we need to input 196608 neurons (255 * 256 * 3). For red, blue, and green, there are three matrices. Because each neuron in the first hidden layer would receive 120,000 weights from the input layer, there is a problem here. When we increase the number of neurons in the hidden layer, the number of parameters will rise quite quickly. Since we have chosen a batch size of 32, the error gradient will be calculated using 32 samples from the training set before the model's parameters are changed. Convolution and pooling layers are used to execute the idea of dividing an image into smaller sub-images for study [10]. In order to reduce the dimensionality of the image, the convolution layer employs a filter to specify the size of the sub-images and a step length to decide how near the sub-images are to one another. Next, the pooling layer preserves significant tiny characteristics by selecting either the maximum or average value from the findings. Finally, a fully connected layer, like a regular neural network, is used to link the sub-images and carry out classification.

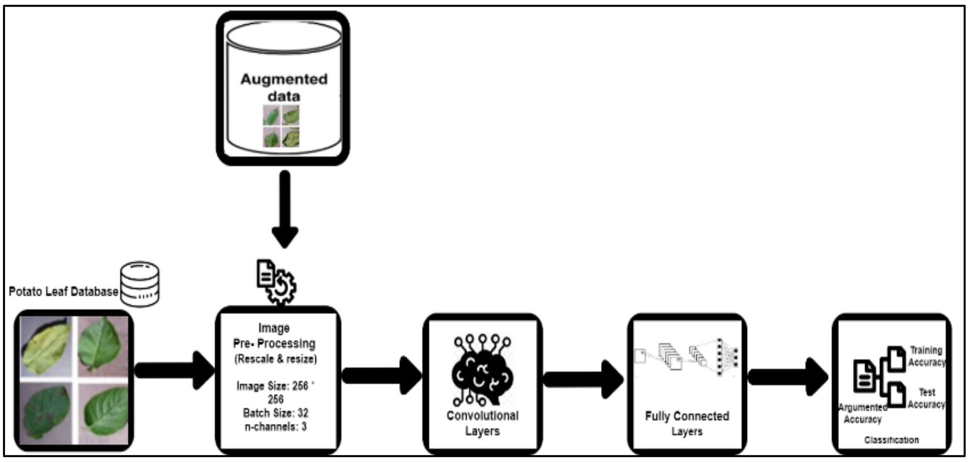


Figure 4. Overall Model Architecture

d) Data Augmentation Techniques

Data augmentation is a vital technique in deep learning, particularly for image classification tasks, where it enhances the robustness and generalisation of models by artificially expanding the dataset. In this study, data augmentation was employed to improve the performance of the Convolutional Neural Network (CNN) model developed for potato disease classification. The augmentation techniques included flipping, rotation, zooming, and shifting of images, which were applied to the training dataset as elaborated in Table 1. These techniques generate diverse training samples, reducing the risk of overfitting and enabling the model to perform better on unseen data.

Table 1. Data Augmentation Techniques

Parameter	Description
Rotation range	range (in degree) to randomly rotate the image
Width / Height shift range	A fraction of total width/height by which to randomly translate the image
Rescale	A factor to multiply the image after loading to rescale the pixel values
Shear range	The shear angle (in degrees) in a counterclockwise direction
Zoom range	The range for random zoom
Horizontal flip	Whether to randomly flip the image horizontally
Fill mode	The strategy used for filling in newly created pixels, which can appear after a rotation or a width/height shift. The possible values are ‘constant’, ‘nearest’, ‘reflect’, or ‘wrap’

The data augmentation process was implemented using TensorFlow's ‘tf.keras.preprocessing.image_dataset_from_directory’ function. Augmented images were generated dynamically during training, ensuring that the model was exposed to different variations of the input data in each epoch. The augmented dataset was then used to train the CNN model across multiple epochs, thereby enhancing its ability to generalise to new data. The Figure 5 shows the first image from the augmented dataset, including its predicted and actual labels as processed by the model.

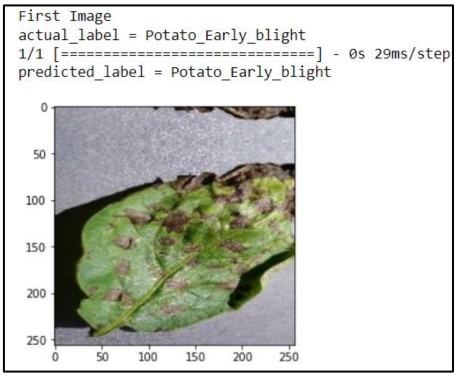


Figure 5. First image from Augmented Dataset

The model trained on the augmented dataset was evaluated, resulting in an accuracy of 85.56% in the final evaluation. This performance metric underscores the effectiveness of data augmentation in improving model accuracy and highlights its crucial role in the training process [19].

The Keras deep learning package, which includes Keras Image Generator, offers a user-friendly interface for creating and refining deep neural networks. Without requiring manual preprocessing, we may effectively preprocess our image while training our model with the help of the image Generator. By altering some aspects of the images, including as rotation, scaling, flipping, shearing, and more, we were able to get a variety of testing data using the data augmentation approach. This prevented overfitting and helped to broaden the range of our training data. To increase our model's generalisation, we performed data augmentation on training images using an image generator. The images are being preprocessed in real time by converting to grayscale, resizing, and normalizing the pixel values. It enables us to instantly create batches of images with an image size of 256 * 256 pixels and a batch size of 32 with three n-channels, which helped us optimise our deep learning model's use of memory.

4. Implementation

a) Dataset Description and Distribution

From an open repository for researchers, we were able to collect a batch of data that included 1500 photographs [11]. The 500 images in each of the three classes—healthy, early blight, and late blight—are equally distributed throughout the dataset. In order to evaluate the model's performance, we generated a total of 180 simulated images using the data augmentation approach. To assess the trained model's accuracy and performance, these generated images are used to expose it to an unknown collection of data. 300 images are set aside for testing, 10% for validation, and 80% for training deep learning models.

b) Preprocessing

Using the tf.data API, train, test, and validation datasets are linked to various data preparation steps. In order to ensure that the model does not get overfit by memorising the dataset's elemental order, we shuffled the elements' order during training. The buzzer size, or the number of items from the data set that will be utilised to randomly sample the next element, is represented by the argument 1000 provided to the shuffle () method. By over-lapping computation and data loading, the auto-tune for buffer size can help keep the GPU busy and boost performance while the current batch of data is being handled. The buffer size = tf.data.TensorFlow is instructed to automatically select an appropriate prefetch buffer size based on the hardware that is available by using the AUTOTONE parameter. All of the images should be resized and rescaled in order to standardise the input data and improve its suitability for model training.

c) Model Training

The model is built using the Keras library consisting of the following neural network layers:

- Conv2D layer with 32 filters of size (3,3) and ReLU activation
- MaxPooling2D layer to down-sample the feature maps
- Conv2D layer with 64 filters of size (3,3) and ReLU activation
- MaxPooling2D layer to down-sample the feature maps
- Conv2D layer with 64 filters of size (3,3) and ReLU activation
- MaxPooling2D layer to down-sample the feature maps
- Conv2D layer with 64 filters of size (3,3) and ReLU activation
- MaxPooling2D layer to down-sample the feature maps
- Flatten layer to convert the 4D feature maps into a 1D vector
- Dense layer with 64 units and ReLU activation

- Dense layer with 3 units and SoftMax activation to get the class probabilities.

The regularisation strategy is used to prevent the model from overfitting, which occurs when the model memorises the training data and does not generalise well to the validation data. During training, it incorporates a penalty term into the loss function to deter complicated or excessive parameter values. This enhances the model's capacity to generalize to new data and lessens its sensitivity to the training set. To the loss function, the L2 regularization is applied. It is computed by multiplying the sum of squares of all model weights by a regularization parameter, usually represented by λ. In the context of machine learning, the loss function (often mean squared error for regression tasks) is modified to include this regularization term. This can be expressed as [23]

$$\text{Regularised Loss} = \text{Loss} + \lambda * \text{sum}(\text{wi}^2)$$

By incorporating this penalty term, L2 regularization discourages the model from fitting the training data too closely, thereby improving its generalization ability on unseen data. The effect of L2 regularization is to shrink the weights of less important features toward zero, but not exactly to zero, thereby retaining all features in the model but reducing their impact. Adam optimizer and Sparse Categorical Cross entropy loss function are set for model compilation. The Adam optimizer combines the concepts of adaptive learning rates and momentum to efficiently update the weights during training. It maintains a running average of both the gradients and the second moments of the gradients. The uncertainty or chaos in a collection of data is measured by entropy. For multi-class classification tasks in this study, the Sparse Categorical Cross-Entropy loss function is employed. The difference between the actual distribution of the classes and the expected probability distribution is computed using the entropy loss function. When the projected probability differs significantly from the actual probabilities, the model is penalized more.

The formula for Sparse Categorical Cross-Entropy loss function is as follows [25]:

$$Loss = -sum(y\_true * log(y\_pred))$$

The one-hot encoded vector, or true distribution, is denoted by y_true, whereas the projected probability distribution generated by the model is denoted by y_pred. The model in this study is trained to attain high accuracy and low loss on training and validation datasets using regularizations, the Adam optimizer, and the Sparse Categorical Cross-Entropy loss function, confirming its usefulness in potato disease prediction.

d) Classification of Test Data

The model is tested with testing data containing 300 test images of 3 classes. The testing accuracy obtained by the model is 95% with a loss of 0.25. Figure 6 displays sample test images with actual and predicted labels.

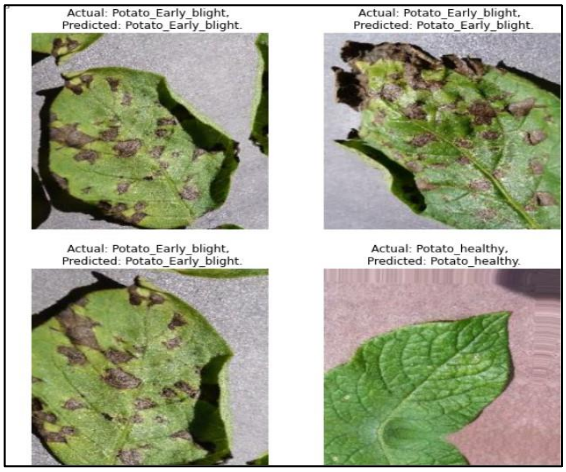


Figure 6. Classification result of the Testing Image

e) Classification of Augmented Data

The model evaluated is tested with images simulated from the existing images using a data augmentation technique. A total of 180 images are augmented with equal distribution from 3 classes. The model is tested with unseen data and the results obtained are 94% accurate with a loss of 0.38. Figure 7 depicts the accuracy of the model prediction for augmented images.

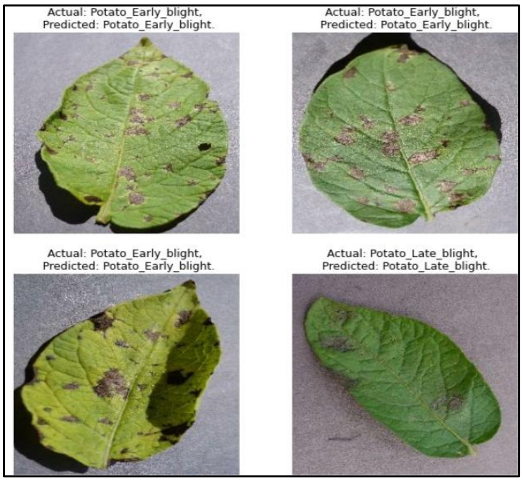


Figure 7. Classification Result of Augmented Image

5. Result And Discussion

a) Training And Validation

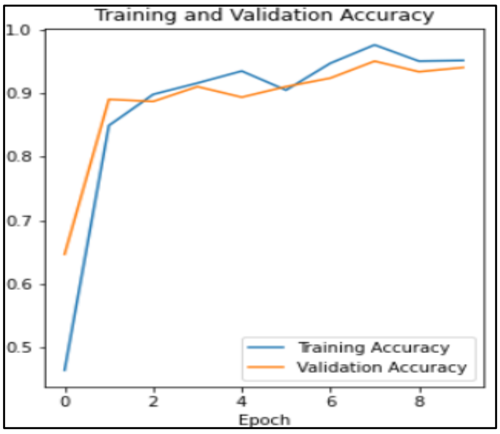
In this study, the preexisting data is used for classifying the plant disease on potato plant leaf images. The model is trained using CNN, a deep-learning approach for image classification.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 sequential (Sequential)     (32, 256, 256, 3)         0

 conv2d (Conv2D)             (32, 254, 254, 32)        896

 max_pooling2d (MaxPooling2D  (32, 127, 127, 32)       0
 )

 conv2d_1 (Conv2D)           (32, 125, 125, 64)        18496

 max_pooling2d_1 (MaxPooling  (32, 62, 62, 64)         0
 2D)

 conv2d_2 (Conv2D)           (32, 60, 60, 64)          36928

 max_pooling2d_2 (MaxPooling  (32, 30, 30, 64)         0
 2D)

 conv2d_3 (Conv2D)           (32, 28, 28, 64)          36928

 max_pooling2d_3 (MaxPooling  (32, 14, 14, 64)         0
 2D)

 flatten (Flatten)           (32, 12544)               0

 dense (Dense)               (32, 64)                  802880

 dense_1 (Dense)             (32, 3)                   195

=================================================================
Total params: 896,323
Trainable params: 896,323
Non-trainable params: 0
_________________________________________________________________
```

Figure 8. Model Summary

The model overview detailing the architecture of the model is given in Figure 8. The model has a linear layer stack and is sequential in nature. The input form of (3, 256,256,3) is taken by the 2D convolution layer with 32 filters and a 3X3 kernel size, and it produces (3, 254, 254, 3) with 896 parameters. The unique resolution of feature maps and other output forms is decreased by the 2D pooling layer (32,127,127, 32). Comparably, three convolution and max pooling layers are employed, succeeded by an output-flattened layer (32, 12544). The 3D tensor is flattened into a 1D tensor via the flattening layer. The 3-unit dense layer, having outputs of (32, 64) and (32, 3), comes after the 64 unit fully linked dense layer. With just 896,323 parameters, the model is well-trained and has no untrainable parameters. Because fewer weights and biases are updated during each training iteration, this increases training speed. Additionally, the model minimizes the chance of overfitting and is simpler to optimize.

After ten epochs, the model's accuracy, as shown in Figure 8, reached approximately 97% on the training data. A higher accuracy indicates that the model is producing correct predictions more frequently. The training process achieved a minimum loss of approximately 0.11, which suggests that the model's predictions closely matched the true labels. The model's validation accuracy, also depicted in Figure 9, is slightly lower than the training accuracy but shows strong generalization capability. The minimum validation loss attained was slightly above 0.20, indicating that the model's predictions on the validation set were reasonably close to the true labels. This suggests a good balance between the model's ability to fit the training data and generalize to unseen data.



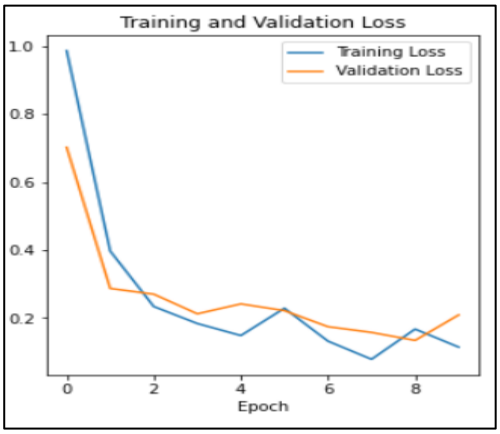


Figure 9. Training Validation Accuracy and Loss of the Model

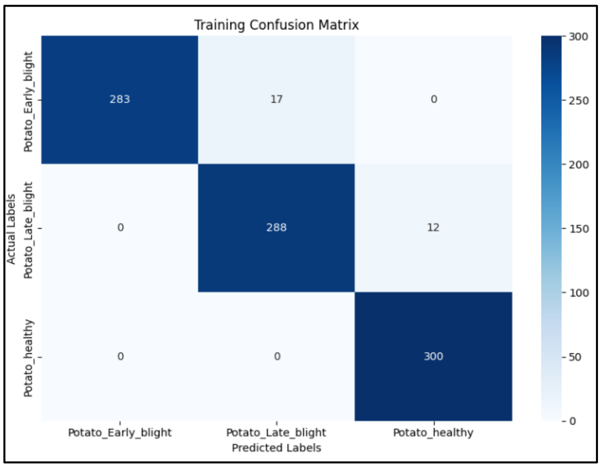


Figure 10. Confusion Matrix on Model Training

Figure 10 displays a confusion matrix that illustrates the classes correct and false predictions. For potato early blight, 283 classes were found to be true positives and 17 to be false negatives. Twelve classes were incorrectly classified, whereas 288 cases were correctly identified as potato late blight. For the potato healthy class, 300 classes were determined to be genuine positives and 0 to be false negatives.

b) Test Data Performance

Test data is used to assess the trained model. The confusion matrix and classification report are used to assess the model's accuracy. The model accurately identified 98 of the occurrences in the first row of Figure 11 as potato early blight. 88 of the occurrences in the second row were accurately identified as potato late blight. 99 of the occurrences in the third row were accurately predicted to be healthy potatoes. Due to the unseen data, this allows for a greater understanding of the model's strengths and flaws.

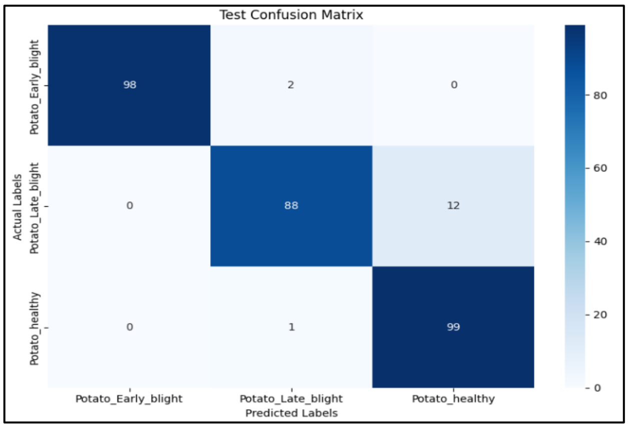


Figure 11. Confusion Matrix for Test Data

Figure 12 shows the Precision-Recall Curve (PRC), which is a graphical depiction of a model's performance. For early blight, the model predicts 100% with accuracy. With a balanced measure of 99% as the f1 score for early blight, recall 98% shows that the model correctly detects all the cases of early blight. In a similar vein, the model performs well for the remaining classes within the 95% overall accuracy range for the test data.

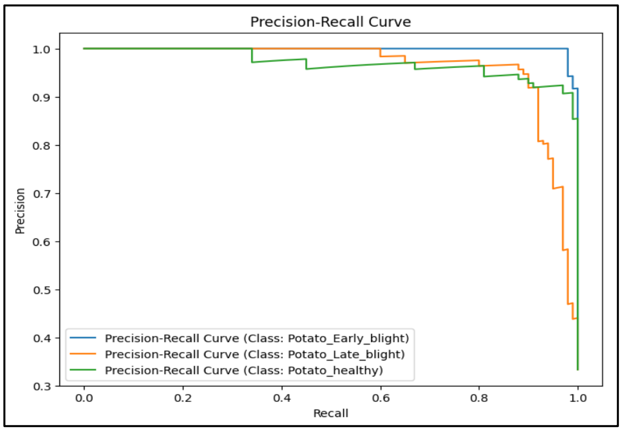


Figure 12. Precision Recall Curve for Test Data

The trade-off between recall and accuracy may be found by examining the PRC in order to determine the right threshold. The overall model performance of a classification model across several thresholds is quantified by the area under the receiver operating characteristic curve (AUC-ROC) measure. As can be shown in Figure 13, early blight has an AUC of 1.00, showing an exceedingly high degree of prediction accuracy. With an AUC of 0.97, the late blight class appears to be quite accurate in differentiating between cases of late blight. With an AUC of 0.99, the healthy class is quite accurate in predicting samples that would be healthy.

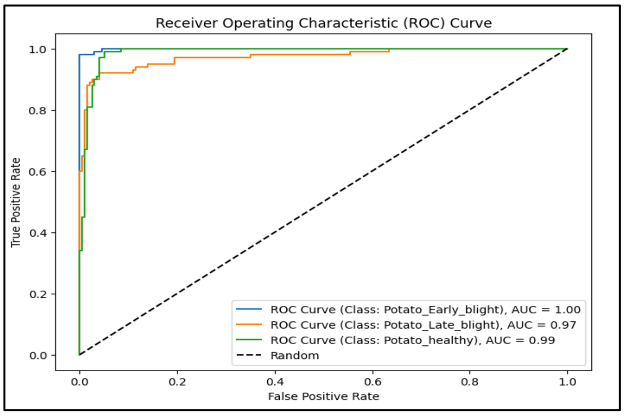


Figure 13. Receiver Operating Characteristic Curve for Test Data

c) Augmented Data Performance

Simulated new images using data augmentation are used for evaluating the model and testing its performance. The model's performance was evaluated using augmented data consisting of 180 images. The confusion matrix, as shown in Figure 14, provides valuable insights into the model's predictions. In the first row, 59 instances were correctly classified by the model as potato early blight. Similarly, in the second row, 55 instances were accurately identified as potato late blight. In the third row, the model correctly predicted 56 instances as potato healthy. This analysis allows for a comprehensive understanding of the model's ability to handle unseen data, highlighting its strengths and weaknesses

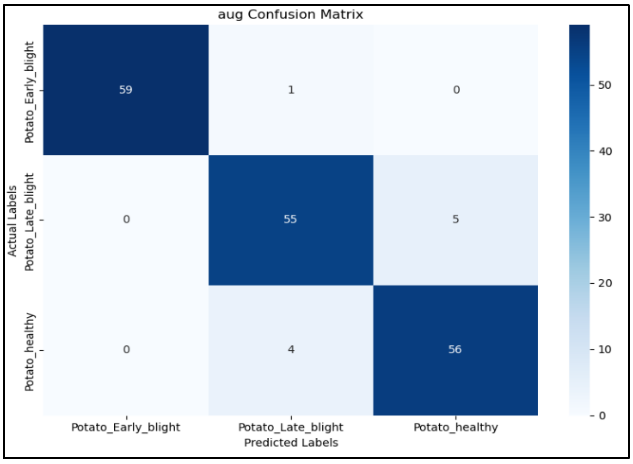


Figure 14. Confusion Matrix for Augmented Data

In Figure 15, the Precision-Recall Curve (PRC) visually represents the model's performance. The model exhibits 100% accuracy in predicting early blight, with a recall of 98%, indicating that it correctly identifies all instances of early blight. The f1 score, which balances precision and recall, stands at 99% for early blight. Likewise, the model performs well for other classes, resulting in an overall accuracy of 94% for the augmented data.

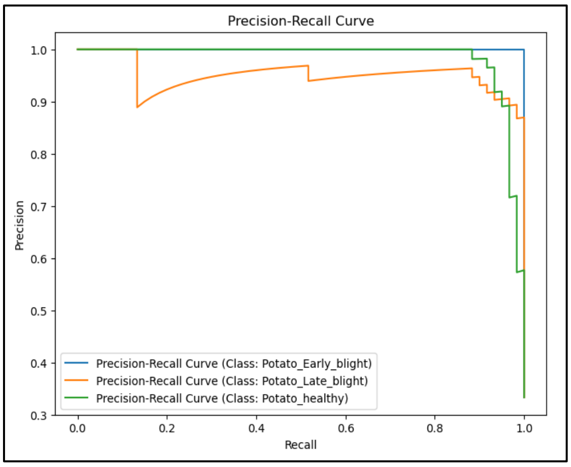


Figure 15. Precision-Recall Curve for Augmented Data

Figure 16 shows the model's performance evaluated using Receiver Operating Characteristic (ROC) curves. The model's capacity to discriminate between several classes is gauged by the area under the curve (AUC). The AUC for the early blight class is 1.00, demonstrating a high degree of precision in early blight prediction. With an AUC of 0.81, the late blight class appears to have a higher degree of accuracy in identifying late blight.

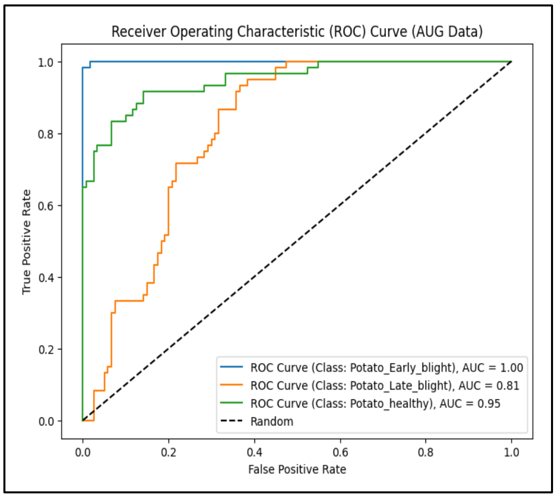


Figure 16. Receiver Operating Characteristic Curve for Augmented Data

The healthy class's AUC of 0.95 shows that it can predict healthy samples with a reasonable degree of accuracy. With these analysis, the ROC graph offers perceptions of how well the model functions for various classes according to their own AUC values.

6. Conclusion

Precision agriculture has been transformed by deep learning models, which make it possible to accurately identify and classify crop diseases. In this work, we created a CNN-based model for predicting potato diseases, and utilizing leaf images, we were able to identify potato diseases with an astounding 97% accuracy rate. For potato producers, this model is an invaluable resource as it facilitates early disease identification and timely disease management methods. Additionally, we showed how the Keras image augmentation method improved the model's ability to generalize, yielding an accuracy of 94% for supplemented data and 95% for test data. The study demonstrates the possibility of using CNNs in conjunction with image augmentation to automatically detect plant diseases, which might ultimately result in higher agricultural yields and more profitable farming. By consistently developing and improving these models, we can provide farmers with strong and dependable instruments to successfully fight crop diseases, guaranteeing the agricultural practices' long-term viability.

References

[1] Li, Xudong, Yuhong Zhou, Jingyan Liu, Linbai Wang, Jun Zhang, and Xiaofei Fan. "The detection method of potato foliage diseases in complex background based on instance segmentation and semantic segmentation." Frontiers in Plant Science 13 (2022): 899754.

[2] T.-Y. Lee, J.-Y. Yu, Y.-C. Chang, and J.-M. Yang, "Health detection for potato leaf with convolutional neural network," in 2020 Indo–Taiwan 2nd International Conference on Computing, Analytics and Net- works (Indo-Taiwan ICAN),Rajpura, India. IEEE, 2020. 289–293

[3] M. Islam, A. Dinh, K. Wahid, and P. Bhowmik, "De- tection of potato diseases using image segmentation and multiclass support vector machine," in 2017 IEEE 30th canadian conference on electrical and computer engineering (CCECE), Canada. IEEE, 2017. 1–4

[4] H. Waghmare, R. Kokare, and Y. Dandawate, "Detec- tion and classification of diseases of grape plant us- ing opposite colour local binary pattern feature and machine learning for automated decision support sys- tem," in 2016 3rd international conference on signal processing and integrated networks (SPIN), Delhi. IEEE, 2016. 513– 518

[5] S. Ramesh, R. Hebbar, M. Niveditha, R. Pooja, N. Shashank, P. Vinod, et al., "Plant disease detection using machine learning," in 2018 International conference on design innovations for 3Cs compute com- municate control (ICDI3C), Bangalore, Indiapp. IEEE, 2018. 41–45

[6] M. K. Singh, S. Chetia, and M. Singh, "Detection and classification of plant leaf diseases in image process- ing using matlab," International journal of life sci ences Research, vol. 5, no. 4, 2017. 120–124

[7] S. N. Ghaiwat and P. A. Detection, "Classification of plant leaf diseases using image processing tech- niques: A review international journal of recent ad-vances in engineering and technology (ijraet) issn (on- line): 2347-2812," Volume-2, Issue-3, 2014.

[8] Badnakhe, Mrunalini R., and Prashant R. Deshmukh. "An application of K-means clustering and artificial intelligence in pattern recognition for crop diseases." In International conference on advancements in information technology, vol. 20, 2011. 134-138.

[9] S. Stearns, "Early Blight and Late Blight of Potato — Integrated Pest Management— ipm.cahnr.uconn.edu." https://ipm.cahnr.uconn.edu/ early-blight-and-late-blight-of-potato/. (Accessed 22-Apr-2023).

[10] Vishnoi, Vibhor Kumar, Krishan Kumar, Brajesh Kumar, Shashank Mohan, and Arfat Ahmad Khan. "Detection of apple plant diseases using leaf images through convolutional neural network." IEEE Access 11 (2022): 6594-6609.

[11] "Potato Disease Leaf Dataset (PLD) — kaggle.com." https://www.kaggle. com/datasets/rizwan123456789/ potato-disease-leaf-datasetpld. (Accessed 22-Apr-2024).

[12] Ratnayake, Upadya, Nalinda Somasiri, Swathi Ganesan, and Sangita Pokhrel. "Enhancing CNN Models with Data Augmentation for Accurate Fertilizer Deficiencies and Diseases Identification in Paddy Crops." In Annual International Conference On Business Innovation (ICOBI 2023), srilanka vol. 1,ICOBI, 2023. 575-582

[13] D. K. Gummalla, S. Ganesan, S. Pokhrel, and N. Somasiri, "Enhanced Early Detection of Thyroid Abnormalities Using a Hybrid Deep Learning Model: A Sequential CNN and K-Means Clustering Approach," Journal of Innovative Image Processing, vol. 6, no. 3, 244–261, 2024

[14] U. Shruthi, V. Nagaveni, and B. Raghavendra, "A review on machine learning classification techniques for plant disease detection," in 2019 5th International conference on advanced computing & communication systems (ICACCS), Coimbatore India. 281–284, IEEE, 2019.

[15] V. Pooja, R. Das, and V. Kanchana, "Identification of plant leaf diseases using image processing tech- niques," in 2017 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR), IEEE, 2017. 130–133

[16] Q. Yan, B. Yang, W. Wang, B. Wang, P. Chen, and J. Zhang, “Apple leaf diseases recognition based on an improved convolutional neural network,” Sensors, vol. 20, no. 12, 2020. 3535

[17] Lu, Jinzhu, Lijuan Tan, and Huanyu Jiang. "Review on convolutional neural network (CNN) applied to plant leaf disease classification." Agriculture 11, no. 8 (2021): 707.

[18] Fulari, Utkarsha N., Rajveer K. Shastri, and Anuj N. Fulari. "Leaf disease detection using machine learning." J. Seybold Rep 1533 (2020): 9211.

[19] Zulkarnain, Izuardo, Rin Rin Nurmalasari, and Fazat Nur Azizah. "Table information extraction using data augmentation on deep learning and image processing." In 2022 16th International Conference on Telecommunication Systems, Services, and Applications (TSSA), Indonesia, IEEE, 2022. 1-6.

[20] R. C. Joshi, M. Kaushik, M. K. Dutta, A. Srivastava, and N. Choudhary, “Virleafnet: Automatic analysis and viral disease diagnosis using deep-learning in vigna mungo plant,” Ecological Informatics, vol. 61, 2021. 101197.

[21] “Potato Disease Identification — AHDB — potatoes.ahdb.org.uk.” https://potatoes.ahdb.org.uk/knowledge-library/ potato-disease-identification. (Accessed 22-Apr-2023)

[22] Mo, Weilong, Xiaoshu Luo, Yexiu Zhong, and Wenjie Jiang. "Image recognition using convolutional neural network combined with ensemble learning algorithm." In Journal of Physics: Conference Series, vol. 1237, no. 2, IOP Publishing, 2019. 022026.

[23] Saud, Arjun Singh, and Subarna Shakya. "Analysis of l2 regularization hyper parameter for stock price prediction." Journal of Institute of Science and Technology 26, no. 1 (2021): 83-88.

[24] Ganesan, Swathi, Nalinda Somasiri, Rebecca Jeyavadhanam, and Gayathri Karthick. "Improved Computational Efficiency of Machine Learning Algorithms Based on Evaluation Metrics to Control the Spread of Coronavirus in the UK." International Journal of Computer and Systems Engineering 17, no. 10 (2023): 532-537.

[25] “tf.keras.losses.SparseCategoricalCrossentropy | TensorFlow Core r2.0,” TensorFlow, 2019.

https://www.tensorflow.org/api_docs/python/tf/keras/losses/SparseCategoricalCrossentropy

[26] Ganesan, Swathi, Nalinda Somasiri, and Chandima Colombage. "Deep learning approaches for accurate sentiment analysis of online consumer feedback." In 2023 International Conference on Computer Communication and Informatics (ICCCI),Coimbatore.. IEEE, 2023. 1-5