A Project Report

on

Potato Leaf Disease Recognition

Submitted in partial fulfillment of the
requirement for the award of the degree of

Bachelor of Computer Science and Engineering

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Dr. D Rajesh Kumar
Assistant Professor

Submitted By

Ashish Kumar Rajbanshi
19SCSE1010815
Anuj Kanu Baniya
19SCSE1010805

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
May, 2023

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled **"Potato Disease Leaf Recognition"** in partial fulfillment of the requirements for the award of the Bachelor of Computer Science and Engineering– submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of February, 2023 to May , 2023, under the supervision of Dr. D Rajesh Kumar, Assistant Professor, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by me/us for the award of any other degree of this or any other places.

Ashish Kumar Rajbanshi (19SCSE1010815)

Anuj Kanu Baniya (19SCSE1010805)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. D Rajesh Kumar

Assistant Professor

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of Ashish Kumar Rajbanshi (19SCSE1010815) and Anuj Kanu Baniya (19SCSE1010819) has been held on 02-05-2023 and his/her work is recommended for the award of Bachelor of Computer Science and Engineering.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: May, 2023

Place: Greater Noida

Abstract

Because crop species, crop disease symptoms, and environmental conditions vary, it might be difficult to detect potato leaf disease in its early stages. These elements make it challenging to spot potato leaf diseases in their early stages. To identify illnesses in potato leaves, numerous machine learning methods have been developed. The existing technology, however, is unable to identify crop species or crop diseases in general because these models are developed and evaluated using images of plant leaves from a particular region. A multi-level deep learning model for identifying potato leaf disease has been built in this study. Using the YOLOv5 image segmentation method, the first level of the algorithm extracts the potato leaves from the image of the potato plant.

In order to identify early blight and late blight potato illnesses from photographs of potato leaves, a novel deep learning technique has been created at the second level. A dataset for potato leaf disease was used to train and evaluate the suggested model for disease identification. The 4062 photos of the potato leaf disease were gathered from the Pakistani province of central Punjab. On a dataset involving potato leaf disease, the suggested deep learning technique had 90% accuracy. On the Plant Village dataset, the effectiveness of the suggested strategies was also assessed. The accuracy and computing cost of the suggested technique are greatly improved when compared to those of the most recent models.

Table of Contents

CHAPTER-1 Introduction

1.1 Introduction

As “Potatoes (Solanum tuberosum) are the world’s most important vegetable crop. Due to the vast diversity in types and high consumer consumption, potatoes are a good enterprise option for many growers. Around 130 and 95 developing nations grow potatoes, the world’s fourth most important staple food. The output of potatoes worldwide has been steadily increasing over the past year, including in emerging nations. However, it is also estimated that over 32% of potatoes are lost annually due to illnesses and pests” [1]. “Potato farmers in Bangladesh lose at least Tk 2,500 crore every year due to unsold surplus production and postharvest losses” [2]. In order to help our framer, we have proposed a model based on convolutional neural network to classify potato leaf disease. “Potato farming dominates as an occupation within the agriculture domain in additional than 125 countries. However, even these crops are subjected to infections and diseases, primarily categorized into two grades: (i) Early blight and (ii) Late blight. Moreover, these diseases cause damage the crop and reduce its production. In fact, potatoes have a more favorable overall nutrient-to-price ratio than many other fruits and vegetables and are an affordable source of nutrition worldwide” [3]. “Late blight damages leave, stems, and tubers. The leaves affected by this disease appear blistered and dry out. When drying out, leaves turn brown or black in color. The remedy to the matter is high humidity, cold, and leaf wetness. The primary blight could also be a specific disease occurring on the foliage at any stage of the expansion, causing characteristic leaf spots and blight. The primary blight is first observed on the plants as small black lesions totally on the older foliage. Lesions on the stems are quite like those on leaves, sometimes girdling the plant if they occur near the soil line. The remedy to the problem is warm, rainy, and wet weather" [4]. "Identifying diseases in potato plants quickly and accurately is important to chop back the impact of diseases on plants. Manual monitoring activities dispensed by farmers become difficult and impractical because it takes an extended time and in-depth knowledge. Identification of plants diseases types that are slow will trigger the spread of diseases in plants uncontrollably. Besides, farmers generally identify diseases in plants in some way that's approximately and assumptions that allow inaccurate identification results because the symptoms on the leaves appear to possess similarities that are difficult to elucidate at a glance". [5] "Farmers use the results of personal identification without expert advice within the sector of plant diseases as a reference for preventing plants infected with the disease. As a result, preventive measures taken by farmers could even be ineffective and will damage crops thanks to inadequate knowledge and misinterpretation of disease intensity, excessive dosage, or lack of dosage" [5]. This problem is that the inspiration of the proposed research is to facilitate farmers in identifying and classifying diseases in potato plants that are fast and accurate. This paper presents a Convolutional Neural Network based approach to identify and classify two common potato infections. Farmers can easily detect the disease in potato crops using the proposed method with little computational effort.

1.2 Formulation of Problem

Farmers Identification of leaf diseases it is the important and one of the major problem in early stages. Disease is caused by pathogen which is any agent causing disease. In most of the cases pests or diseases are seen on the leaves or stems of the plant. Therefore identification of plants, leaves, symptoms and finding out the pest or diseases, percentage of the pest or disease incidence, symptoms of the pest or disease attack, plays a key role in successful cultivation of crops. What is wrong with my plant; followed by, what can I do to get rid of the problem? It may be too late to help the specific plant when the question is asked, but proper diagnosis may be extremely important in preventing the problem on other plants or in preventing the problem in the future. Control measures depend on proper identification of diseases and of the causal agents. Therefore, diagnosis is one of the most important aspects of a plant pathologist's training. Without proper identification of the disease and the diseasecausing agent, disease control measures can be a waste of time and money and can lead to further plant losses. consider various environmental and cultural factors. Be able to identify a disease and disease-causing agent, Be able to narrow the problem down to several possibilities which will require further study in the laboratory before he can make a final diagnosis, or Identify characteristic symptoms. Describing the characteristic symptoms exhibited by a specimen can be very difficult to do accurately. Because of this, it is often difficult, if not impossible, to determine what is wrong with a plant when a person is describing symptoms over the phone. As a test of this you may want to take a plant exhibiting symptoms and have three different individuals describe the symptoms that they observe on a sheet of paper.

1.2.1 Tools and Technology Used

As Early disease diagnosis and identification are essential components in the world of agriculture because infections in plants are unavoidable. The following actions must be completed in order to use an automated image processing system to accomplish this. It has been offered to provide the work's methodological analysis. The leaf type detection and classification of more number of samples is attempted here colour, texture and area features and result of such system is applicable to automation of detection and classifying diseases on leaf in agriculture field. The work involves processing of images of different types of leaves (Healthy and Diseased) as shown below.

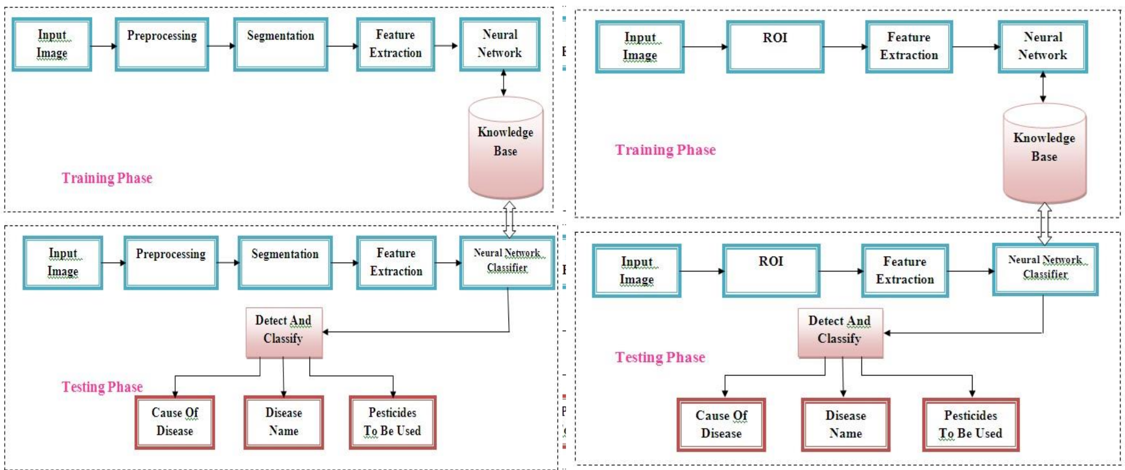


Figure 3.1 shows possible interactive and non-interactive systems.

The work accepts a colour leaf image as input and, after accepting the image, does pre-processing on the leaf image by shrinking the image's size and applying a Gaussian filter technique to the image. In the segmentation process, the K-mean clustering technique is used to extract the leaf section. For example, if the leaf is healthy, the entire image is extracted, and if the leaf is sick, the infected area of the image is extracted. Following the extraction of the leaf's portion from the image, the features (colour, texture, and area) will be extracted. For later retrieval, the retrieved features are set aside in a knowledge base. The multilayered feed forward network is the neural network architecture that is most frequently utilised with the back propagation technique. The training procedure is divided into four steps:

I. Put together the training data.
II. Establish the network.
III. Train the network.
IV. Evaluate and confirm network response to fresh inputs

Images of actual leaves are used for neural network training, testing, and validation. A combined features set with 24 input features is created by combining the feature sets for colour, texture, and area. A classifier was created using a three layer BPNN. By examining the feature set in the detect and classify step, the outcome will be displayed. The outcome is categorised as either a healthy leaf or an infected leaf (name, reason, and insecticides to be applied).

Image Gathering and Pre-Processing:

In the last chapter, we covered the difficulties farmers encounter when trying to identify and categorise various potato leaf diseases as well as the suggested work strategy. The tools and techniques used to identify and categorise the various potato leaf kinds are covered in this stage.

Imaging System ::

A digital camera (Sony Corporation, DSC-W690, built in China) was used to capture the photos. A personal computer with an Intel Dual-Core CPU housed the photographs. The camera was put on a straightforward platform that allowed for simple vertical adjustment to precisely adjust the camera's position with regard to leaf photos, which were then taken and saved in jpg format. Images of healthy and sick leaves were captured in direct sunshine while maintaining the same distance. High-quality zoom lens is a feature of the camera.

Leaf Sample Imaging ::

In this investigation, photographs of both healthy and diseased potato leaves were taken into consideration. The first and second sets of images each contained a single sort of leaf image. From the initial set of photos, a classifier was trained using a few features (RGB Colour Features, Texture Features, and Shape Features). The segmentation method and trained classifier are put to the test using the second set of photos.

Image Samples:

demonstrates a picture of the leaf samples. For the investigation, various leaf image kinds (both healthy and infected) are taken into consideration.

Figure 3.2: Pictures of several leaf samples

Pre-processing Methodology::

The image quality, which affects both the capacity to detect features, is crucial for the findings analysis. It can also be described as a method whereby an image's data is converted to digital form and then subjected to a number of mathematical procedures, typically using a digital computer. This is done to improve the image's usefulness or aesthetic appeal to a human spectator, or to carry out some of the interpretation and recognition tasks often done by humans. Techniques like image resizing, filtration, segmentation, etc. are used during pre-processing.

Resize::

In order to lessen the computational load on the processing stage, the acquired photos are initially downsized to a fixed resolution. Additionally, it increases the effectiveness of storage.

Filtering::

Image A software procedure called filtering modifies the appearance of an image or a portion of an image by changing the pixel's hues and lines in some way. Filters can be used to add a wide range of textures, tones, and unique effects to a picture in addition to boosting brightness and contrast. Dewdrops, insect waste, and dust are all potential sources of image distortion during image collecting. The segmentation and feature extraction of illness spots would be hampered by them, which are viewed as picture noises. So, they must be eliminated or diminished before performing any additional image analysis. To reduce noise, you can use filters like Gaussian, median, linear, low pass, high pass, Laplacian, etc.

CHAPTER-2

Literature Survey

2.1 Literature Survey

Recent years have seen the introduction of innovative machinery for rice cultivation that can measure the water temperature and level in the paddy field autonomously. A cultivation management system was also recommended to gather farmer experience. Other problems with biological information sensing, such as physiological and growth conditions, do exist.

"Hokkaido Agricultural Research Center develops unmanned system Agricultural machinery (rice planters, tractors, etc.) equipped with remote monitoring functions" [6]. Using object detection and straight-line tracking, this system will be connected to quasi-zenith satellites. In a supermarket context, Faria and other writers have presented a framework for classifier fusion that can assist automatic fruit and vegetable recognition. The authors show that the suggested framework performs better than a number of related works in the literature. [7]. "Geetharamani & the authors proposed a nine-layer CNN for leaf disease classification. They claimed that their model outperforms traditional approaches in terms of accuracy" [8]. Bouaziz et al. proposed "a deep learning-based approach that automates the process of classifying banana leaf diseases" [9]. After rice and wheat, potatoes are the most widely grown and sought-after crop. The potato is one of the most important food crops in the world and is a native of the Peruvian and Bolivian Andes. As a cooked vegetable, potatoes are commonly served whole or mashed. They are also processed into potato flour, which is used in baking and as a sauce thickening. The tubers offer ascorbic acid, protein, thiamin, and niacin and are extremely digestible.

Shen and the authors suggested developing a unique method based on computer image processing because the majority of plant disease grading now done is done by eye. The Otsu approach was used to segment the image's leaf area after a thorough examination of all pertinent variables. The H component was selected to divide the sickness spot in the HSI colour system in order to reduce the disturbance brought on by changes in lighting and veins. Then, illness spot regions were divided using the Sobel operator in order to examine disease spot edges. Last but not least, plant illnesses are graded using the ratio of disease spots to leaf surface area. Studies have shown that this system of classifying plant leaf spot infections is quick and accurate [10].

Using different plant varieties, Appasaheb and the authors give a general review of leaf parameter analysis, healthy, sick, or affected leaf region detection, and categorization of leaf diseases. It is essential and difficult for human eyes to see the particular type of leaf disease with the naked eye. A different set of disease symptoms can be seen on each plant leaf. The algorithm designed for one plant does not work well with the leaf of another plant. To identify leaf diseases in custard apple plants, specialised algorithms are required in addition to the leaf parameter analyzer [11].

The authors and Patnaree concentrate on creating a deep learning-based system to analyse and classify orchid illnesses. In this study, we created a deep learning-based model and compared it against ResNet-50, VGG-16, and VGG-19, three previously trained models. The model is trained, and the parameters are changed, using databases of photos of orchid diseases [12].

"Algorithm is a novel and first technique to address and report the successful implementation for the detection and classification of four diseases in potato leaves," suggested Rabbia and the authors. On the testing set, the algorithm's performance was assessed, and its accuracy was found to be 97.2%. [13]. "Methodology finally achieved 97.89% accuracy for classification between late and early blight syndromes as compared to healthy potato leaf," according to Chakraborty and the authors. This study displayed the fine-tuned VGG16 model's detailed architecture together with validation accuracy and losses. Additionally, the existing approaches have been contrasted with our proposed methodology" [14]. According to Chen and the authors' proposed "procedure," distinct potato disease kinds could be identified with an average accuracy of 97.73%, outperforming other approaches that were compared. Experimental results demonstrate a competitive performance and support the applicability of the suggested method" [15].

Chapter-3

Functionality / Working of the Project

3.1 Deep Learning:

In order to identify features and learn representations, the Deep Learning (DL) method uses a complicated hierarchy that connects several internal layers. The expression of the extraction of critical information from observation data via representation learning is applied in the actual world. Artificial operations extract features by trying and failing.

However, DL acquires the feature that is most appropriate for identifying the image by using the pixel level of the image as an input value [16,17]. The most basic kind of neural network (NN) is a single-layer perceptron network, which has a single layer of output and feeds its inputs directly to its outputs. As a result, it is the most fundamental kind of feed-forward network. Similar to a conventional multi-layer perceptron, Convolutional Neural Networks (CNN) employ the backpropagation paradigm throughout their learning phase. The coupling coefficient and weighting filter are updated using CNN using stochastic gradient descent. In order to determine the ideal feature, CNN employs convolutional and pooling techniques [18,19].

Deep learning allows computational models with several processing layers to learn representations of data at different levels of abstraction. These methods have considerably improved the state-of-the-art in many other domains, such as drug discovery, genomics, object identification, visual object recognition, and speech recognition. Deep learning can reveal intricate structure in huge data sets by using the backpropagation approach to recommend adjustments to a machine's internal parameters that are used to compute the representation in each layer from the representation in the previous layer. While deep convolutional nets have made progress in the processing of images, video, voice, and audio, recurrent nets have shed light on sequential data types including text and speech [20,21].

3.2 Methodology::

As depicted in Fig. The four primary steps of the methodology outlined in this study are data collection, data pre-processing, data augmentation, and image classification [22].

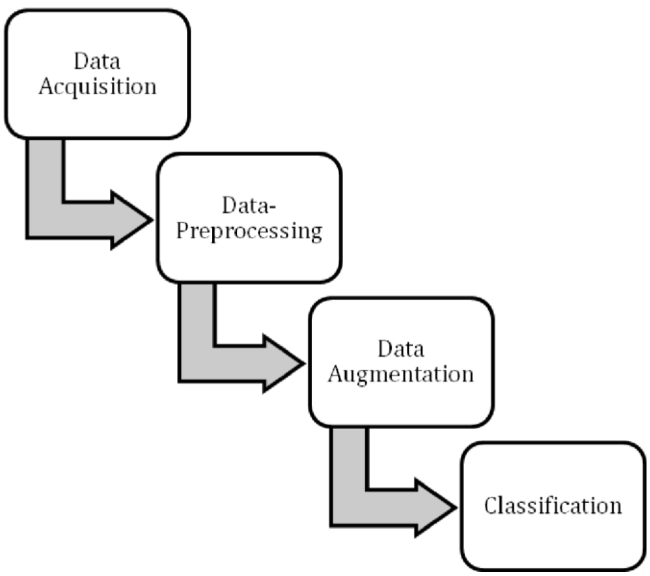


Fig. 1. The suggested approach

4.1 Data acquisition

Various sources, including those gathered by the authors from a potato plantation in Patuakhali, provided images with various resolutions and sizes. Additionally, we made use of Kaggle's free image database. 2152 photos from Kaggle and 7848 taken directly from the field were collected by the author. In the publication, the investigation was conducted using a total of 10,000 photographs. There are three classes for all of the photographs. They are Healthy, Late Blight, and Early Blight.

Early Blight: The bacteria Alternaria solani causes early blight, a type of plant epidemic. Tiny black specks develop into large, brown to black, round to ovoid lesions that can occasionally be restrained by leaf veins but may also be connected to lenticels. The leaves' undersides then sprout a black fungus. Early blight may be the cause of potato tuber wilt. This illness will begin to spread as temps rise above 26 C. It typically manifests when the activity of potatoes is reduced as a result of high-temperature drying or a deficiency in fertiliser.

Figure 2. Potato leaf early blight illness

Late Blight: The bacterium Phytophthora infestans causes the plant disease known as late blight. In years with low temperatures and a lot of rain, outbreaks can significantly reduce the number of potatoes produced.

Fig. 3: The potato leaf disease known as late blight

Healthy Leaf: A healthy leaf has a fresh appearance and is free of any disease.

4.2 Pre-processing of Data

Pre-processing Noise in the image can be decreased by deleting the area of the image that is not the region of interest. If there is too much noise in the image, it won't be used. After being obtained from multiple sources and of various sizes, the input photographs for the dataset must be scaled to 256x256 pixels.

4.3 Data augmentation

Data augmentation is a technique for enhancing data without changing the original intent of the data. The data for this study need to be supplemented. The augmentation parameters in this work are produced by the automatic application of simple geometric transformations such translations, rotations, scale changes, shearing, and vertical and horizontal flips.

Fig. 4. Healthy potato leaf

4.4 Image classification

Artificial intelligence (AI) includes machine learning (ML), also known as deep learning (DL), deep neural learning, or deep neural network. The word "deep" in deep learning denotes that it has more layers than machine learning. In many areas, including object identification, speech recognition, object categorization, and image classification, deep learning techniques have increased the bar [19]. One of the kinds of deep learning that is most popular is convolutional neural networks. Based on the condition of the leaves, convolutional neural networks have been utilised in various studies to detect plant diseases.

Convolutional neural networks, as a whole, consist of one or more convolutional layers that are grouped according to function. Frequently, one or more fully linked layers that are typical of a neural network come after the subsampling layer. Each feature layer receives an input feature set from the previous layer that is enclosed in a small area.

3.3 Related Work::

The classification of plant diseases has been the subject of research using a variety of techniques. However, it is still regarded as missing [5] and has evolved into a study topic that is currently being explored due to how widely this field's subject matter differs. Agriculture-related articles totaling 37 were released between 2015 and 2017. More specifically, 7 papers in 2016, 15 papers in 2017, and 15 papers in

2016 were published. This fact demonstrates how recent and cutting-edge this technology is in the field of agriculture [6].

An approach based on deep learning can be developed to learn a useful features representation. Deep learning [7] has demonstrated excellent performance in a variety of tasks involving visual perception, including text detection [8, [9], victim detection [10, [11], target tracking [12, [13], and object detection [14, [15], and [16]. Accuracy may increase with a deeper network. Intriguingly, it has been theoretically and empirically demonstrated in the work [10] that the final layers of the deep network can capture more semantic information or abstraction; as a result, it is more resistant to variations in pose, colour, scale, and deformable object, suggesting that it may be suitable for robustly classifying leaf diseases.

Prajwala and others [4] In this study, Septoria Leaf Spot and Yellow Leaf Curl, two tomato leaf diseases that are frequently observed in tomato crops, are detected and identified using the LeNet architectural model and Convolutional Neural Network model architecture. Researchers were able to access the dataset through PlantVillage, one of the Open Access picture databases. The methodology suggested for this paper yields an accuracy level of 94% to 95%.

Convolutional neural network (CNN) method-based experimental results on a model created by Srdjan Sladojevic et al. [17] with a novel strategy to identify diseases in five different plant species and 13 different types of diseases. An average accuracy of 96.3% was found from this investigation. Erika Fujita et al. [18] developed a disease diagnosis system for cucumber plants utilising the Convolutional Neural Network technique and the AlexNet architecture, and they achieved an average accuracy of 82.3%.

Raja P et al. [19] collected 2000 photos of maize leaves from the PlantVillage open-access image database. Using a variety of variables and the Multiclass SVM

(Support Vector Machine) approach, the dataset is utilised to classify the three different illnesses that might affect maize leaves. The accuracy of the method was also assessed in this instance in comparison to the histogram method and feature-based grey level co-occurrence. The obtained findings demonstrate the complete accuracy of the SVM (Support Vector Machine) Multiclass technique. Segmentation with K-mean clustering is used by Pranjali B. Padol and Prof. Anjali A. Yadav [20] to locate the illness zone, and then the colour and texture of the image are extracted. While they achieve an accuracy of 88.89% when using the SVM (Support Vector Machine) Classifier technique for classification.

For the feature extraction stage, Jobin Francis et al. [21] used GLCM (Grey Level Co-occurrence) and the Backpropagation Neural Network approach to classify different disease types in pepper plants. Berry Spot Disease, a type of fungal infection seen in pepper, and Rapid Disease, a type of disease brought on by mineral deficiencies such nitrogen, magnesium, and potassium, are examples of pepper plant diseases that can be identified. The classification of illnesses including Alternaria Alternata, Anthracnose, Bacterial Rot, Leaf Spot, and Plant Leaf Cancer was done using the same method by Eftekhar Hossain et al. [22]. The KNN disease identification system in this research performed with an accuracy of 96.76%.

Artificial neural networks and fuzzy logic techniques were used in a study by Aakanksha Rastogi et al. [23] to identify plant illnesses based on the health of the leaves. The goal of this study is to recognise and categorise leaf diseases in maple and hydrangea plants. There are two classifications for the disease-affected plants' leaves: scorched leaves and leaf spots. Leaf spots are areas of the leaf where the disease is concentrated, whereas burnt leaves are areas of the leaf where the disease has spread widely.

Chapter-4

Module Discussions

4.1 Implementation::

As In this study, we identify the important illnesses of potato leaves for categorization using the dataset from potato leaves. 10,000 images from three classes that look surprisingly similar but depict different diseases are included in the collection. The code was created using the Python programming language.

A list of CNN hyperparameters is shown in Table 1. To create a training model is the aim. As a result, we utilised a batch size of 32 and a maximum of 50 epochs for the training model. The image is only 256*256 pixels in size. There are four convolutional layers, four pooling layers, and four fully connected layers in the sequential model on which the network is constructed.

Rectified Linear Unit (ReLU) is used as the activation function to improve. The image of the model. Our CNN model accounts for the dropout layer to prevent overfitting. In the output layer, Softmax is used as the activation function to segment the final result into several diseases.

Table 1. List of hyperparameters

Function	**Values**
Epoch	30, 40 & 50
Batch size	32
Filter sizes for convolution layer	3×3
Activation function	ReLU
Loss function	sparse Categorical cross-entropy
Optimizer	adam

4.2 Convolutional Neural Network (CNN)

Here, we employed a convolutional neural network (CNN)-based method, a kind of deep learning (DL) methodology that takes an input image and prioritises a number of other things in the image while also differentiating between them.

A CNN requires substantially less pre-processing than other classification algorithms do. With enough practise, CNN can learn these filters and their properties, as opposed to basic approaches that demand hand-engineering of filters [23].

These layers make up the bulk of our architecture::

- Input
- Convolution
- Pooling
- Fully connected
- Output
- These are some layer of many layer in computer system.

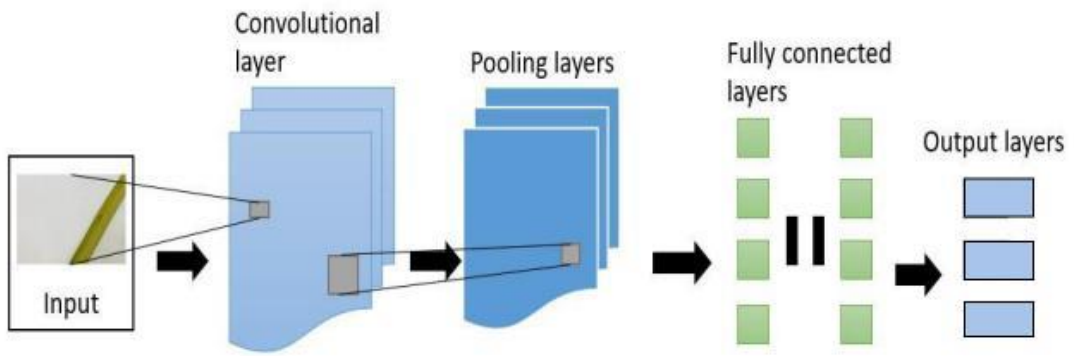


Fig. 5. Layers of CNN

The graphic above depicts CNN's organisational structure. The input is provided to CNN in the form of an image after preprocessing the data and extracting the relevant features. CNN then processes the input through three layers of CNN to accurately represent it. The ultimate outcome is then displayed [24].

- **Input Layer:** The dataset makes up the input layer of CNN. A 3X3 matrix will be used to represent the input data.

- **Convolution Layer:** A layer that extracts features from an image by using filters to learn from more manageable portions of the input data.

- **Pooling Layer:** By reducing the dimensionality of the image, this layer can reduce the processing power needed for later layers. There are two distinct pooling methods. As follows:
- **Max pooling:** While parsing the input, the pixel with the highest value is chosen and sent to the output. Compared to average pooling, it is the strategy that is employed the most.

- **Fully Connected Layer (Dense):** The Fully Connected Layer (Dense), one of CNN's last layers, can identify features that are strongly related to the output class. The outcome of flattening the pooling layer results is a one-dimensional vector.

- **Dropout Layer:** This layer is used to address the issue of model overfitting by eliminating a random assortment of neurons from it. It has a connection to FC layer.

- **Output Layer:** The output layer is where the final categorization outcome is stored.

Researchers must be cautious while choosing a machine setup. We used an Intel Core i5 processor, 8 GB of RAM, and an Nvidia Geforce GPU for this study. Windows 10 was the operating system in use. It is strongly advised to use the GPU when training the model to avoid lengthy training times. Analysis of the CNN's performance is possible thanks to the model's training utilising supervised learning on the dataset. In supervised learning, data annotations serve as references during the training phase.

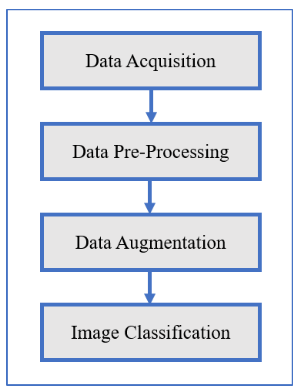
Windows 10 was the default operating system. It is strongly advised to train the model on the GPU rather than a CPU otherwise it will take a long time. The performance of the CNN can be evaluated once the model has been trained using supervised learning on the dataset. Throughout the training phase of supervised learning, data annotations are used as references.

Table 2. Fundamental hardware & software

Hardware/Software Characteristic	Hardware/Software Characteristic
Processor (CPU)	Intel core-i5 (8th Gen)
RAM	8GB DDR4
Operating System	Windows 10
Graphics (GPU)	NVIDIA GeForce MX230
Environment	Tensorflow
Programming Language	Python

4.3 Proposed Approach::

As depicted in Fig. 3. The four primary steps of the methodology outlined in this work are data acquisition, data pre-processing, data augmentation, and image categorization.



Block Proposed Methodology in Diagram 3, Figure 3..

A. Data Acquisition

A variety of image resolutions and sizes were gathered from several sources, including those gathered by the authors from a potato farm in Malang, Indonesia, an open-access image library called PlantVillage [24], and Google photos. obtained a dataset of approximately 5,100 photos, which was split into class five: diseases induced by Alternaria Solani, as seen in Figs. 4 and 5, healthy individuals, Phytophthora infestans, shown in Figs. 6 and 7, viruses, and insects, shown in Figs. 7 and 8.

Solani's Alternaria, Figure 4. Leaves with brown, non-glossy dead spots and concentric circles in a target-board pattern are signs of this disease..

Healthy, as shown in Figure 5.

Figure 6. A bug. The presence of obvious holes in the leaves indicates an insect attack.

Figure 7. A virus. There are numerous viral kinds that affect potato leaves, but only leaves with necrotic patches show symptoms.

Phytophthora Infestans, Figure 8. This disease can be identified by the irregular, dark brown to black dots that appear on the leaves.

B. Data Pre-Processing

In order to produce an accurate training model output, pre-processing data tries to enhance the quality of the data. The first stage is to reduce the amount of noise in the image; if there is too much noise, the image won't be used. To standardise the entry of photos into datasets, acquired images are reduced to 800x600 pixels from their original diversity of sizes.

C. Data Augmentation

In contrast to the shallow networks used in machine learning, deep learning (deep network) requires a lot of data. Common issues with machine learning include a shortage of training data and an imbalance in the volume of data for each class [25]. Data augmentation is the approach taken to solve this issue. Data augmentation is a method of modifying data without distorting its original meaning.

This study requires the use of data augmentation because the 5100 datasets currently available are insufficient to get the best performance. In total, we produce 9000 photos of augmented data. Data augmentation employs a variety of techniques, ranging from histogram-based techniques to straightforward methods for transforming photos, such as turning, rotating, expanding, and cropping. Simply rotating and cropping were the only straightforward transformation techniques used in this investigation.

D. Image Classification

Deep learning (DL), similarly known as deep neural learning or deep neural network, is part of machine learning (ML) in artificial intelligence (AI). The term "deep" means that Deep Learning has more layers than Machine Learning. Deep learning methods have improved the state-of-the-art in image classification, speech recognition, visual object recognition, object detection, and many other domains [7]. In Deep Learning, Convolutional Neural Network is one of the popular classes. Some studies use the convolutional neural network method to detect diseases in plants based on leaf conditions [4] [17] [18].

One or more convolutional layers that are organised into groups according to function make up convolutional neural networks in general. The subsampling layer is frequently followed by one or more fully linked layers that are typical of a neural network. A feature set contained in a limited area on the previous layer serves as input for each feature layer. The CNN architecture model started with LeNet and has since developed into more contemporary CNN architectural models as AlexNet, VGG network, GoogLeNet, residual networks (ResNet), and densely connected networks (DenseNet) [26].

4.4 Source Code::

Import all the Dependencies

```python
import tensorflow as tf
from tensorflow.keras import models, layers
import matplotlib.pyplot as plt
from IPython.display import HTML
```

Set all the Constant

```python
BATCH_SIZE = 32
IMAGE_SIZE = 256
CHANNELS=3
EPOCHS=50
```

Zip file Unziping

```
!unzip archive.zip
```

```
Archive:  archive.zip
  inflating: PlantVillage/Potato___Early_blight/001187a0-57ab-4329-baff-e7246a9edeb0___RS_Early.B 8178.JPG
  inflating: PlantVillage/Potato___Early_blight/002a55fb-7a3d-4a3a-aca8-ce2d5ebc6925___RS_Early.B 8170.JPG
  inflating: PlantVillage/Potato___Early_blight/009c8c31-f22d-4ffd-8f16-189c6f06c577___RS_Early.B 7885.JPG
  inflating: PlantVillage/Potato___Early_blight/00d8f10f-5038-4e0f-bb58-0b885ddc0cc5___RS_Early.B 8722.JPG
  inflating: PlantVillage/Potato___Early_blight/0182e991-97f0-4805-a1f7-6e1b4306d518___RS_Early.B 7015.JPG
  inflating: PlantVillage/Potato___Early_blight/02578b86-b234-4ac0-9bc3-691b5610e2bf___RS_Early.B 7562.JPG
  inflating: PlantVillage/Potato___Early_blight/0267d4ca-522e-4ca0-b1a2-ce925e5b54a2___RS_Early.B 7020.JPG
  inflating: PlantVillage/Potato___Early_blight/028f9b73-142f-499a-9c7b-d7c1ed5e5506___RS_Early.B 8546.JPG
  inflating: PlantVillage/Potato___Early_blight/034959c1-f1e8-4a79-a6d5-3c1d14efa2f3___RS_Early.B 7136.JPG
  inflating: PlantVillage/Potato___Early_blight/03b0d3c1-b5b0-48f4-98aa-f8904670290f___RS_Early.B 7051.JPG
  inflating: PlantVillage/Potato___Early_blight/042135e2-e126-4900-9212-d42d900b8125___RS_Early.B 8791.JPG
  inflating: PlantVillage/Potato___Early_blight/044c3abc-0bc9-45fb-8fd5-094aeb605f90___RS_Early.B 8044.JPG
  inflating: PlantVillage/Potato___Early_blight/048d18ae-98b1-484d-97da-5a0e69b9ebc1___RS_Early.B 6845.JPG
  inflating: PlantVillage/Potato___Early_blight/04c8e6b9-7710-4cdd-b259-2d78b15d1036___RS_Early.B 7066.JPG
  inflating: PlantVillage/Potato___Early_blight/04ee51b6-07e2-4182-84f8-46b22c8938a2___RS_Early.B 8091.JPG
  inflating: PlantVillage/Potato___Early_blight/04fd2a46-ddd4-4b0b-8f19-5ecca482a7d5___RS_Early.B 7273.JPG
  inflating: PlantVillage/Potato___Early_blight/05c35093-11b8-4cd0-b67a-148859754440___RS_Early.B 8939.JPG
  inflating: PlantVillage/Potato___Early_blight/0604174e-3018-4faa-9975-0be32d2c0789___RS_Early.B 7123.JPG
  inflating: PlantVillage/Potato___Early_blight/060fd5a7-1606-4a59-895b-604c90d6b414___RS_Early.B 7205.JPG
  inflating: PlantVillage/Potato___Early_blight/065fc68f-88c9-4fc3-b0a6-a6f5e1072eaa___RS_Early.B 7174.JPG
```

```
...
  inflating: PlantVillage/Potato___healthy/f5bd2732-197b-4d8e-bb50-a4966dc416f5___RS_HL 1840.JPG
  inflating: PlantVillage/Potato___healthy/f66af6c0-6fc6-4020-b1de-a73685d497c2___RS_HL 4181.JPG
  inflating: PlantVillage/Potato___healthy/fd92bd9c-02cb-4599-8673-9316159504a1___RS_HL 1927.JPG
  inflating: PlantVillage/Potato___healthy/ff700844-68ad-4e99-8427-58a39c07f817___RS_HL 1860.JPG
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```python
dataset = tf.keras.preprocessing.image_dataset_from_directory(
    "PlantVillage",
    shuffle=True,
    image_size = (IMAGE_SIZE,IMAGE_SIZE),
    batch_size = BATCH_SIZE
)
```

```
Found 2152 files belonging to 3 classes.
```

```python
HTML("""

""")
```

```python
HTML("""

""")
```

Python

```python
Class_names = dataset.class_names
Class_names
```

Python

```
['Potato___Early_blight', 'Potato___Late_blight', 'Potato___healthy']
```

```python
len(dataset)
```

Python

```
68
```

As you can see above, each element in the dataset is a tuple. First element is a batch of 32 elements of images. Second element is a batch of 32 elements of class labels

Visualize some of the images from our dataset

```
for image_batch, labels_batch in dataset.take(1):
    print(image_batch.shape)
    print(labels_batch.numpy())
```

```
(32, 256, 256, 3)
[0 1 1 1 0 0 0 1 0 1 1 0 0 1 0 1 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 1]
```

```
plt.figure(figsize=(10, 10))
for image_batch, labels_batch in dataset.take(1):
    for i in range(12):
        ax = plt.subplot(3, 4, i + 1)
        plt.imshow(image_batch[i].numpy().astype("uint8"))
        plt.title(Class_names[labels_batch[i]])
        plt.axis("off")
```

#Function to Split Dataset

Dataset should be bifurcated into 3 subsets, namely:

1. Training: Dataset to be used while training
2. Validation: Dataset to be tested against while training
3. Test: Dataset to be tested against after we trained a mode

```
len(dataset)
```

```
68
```

```
train_size = 0.8
len(dataset)*train_size
```

```
54.400000000000006
```

```
train_ds = dataset.take(54)
len(train_ds)
```

```
54
```

```
test_ds = dataset.skip(54)
len(test_ds)
```

```
14
```

```
val_size=0.1
len(dataset)*val_size
```

```
6.800000000000001
```

```
val_ds = test_ds.take(6)
len(val_ds)
```

```
6
```

```
test_ds = test_ds.skip(6)
len(test_ds)
```

```
8
```

```
def get_dataset_partitions_tf(ds, train_split=0.8, val_split=0.1, test_split=0.1, shuffle=True, shuffle_size=10000):
    assert (train_split + test_split + val_split) == 1

    ds_size = len(ds)

    if shuffle:
        ds = ds.shuffle(shuffle_size, seed=12)

    train_size = int(train_split * ds_size)
    val_size = int(val_split * ds_size)

    train_ds = ds.take(train_size)
    val_ds = ds.skip(train_size).take(val_size)
    test_ds = ds.skip(train_size).skip(val_size)

    return train_ds, val_ds, test_ds
```

```
train_ds, val_ds, test_ds = get_dataset_partitions_tf(dataset)
```

```
train_ds, val_ds, test_ds = get_dataset_partitions_tf(dataset)
```

```
len(train_ds)
```

```
54
```

```
len(val_ds)
```

```
6
```

```
len(test_ds)
```

```
8
```

Cache, Shuffle, and Prefetch the Datase

```
train_ds = train_ds.cache().shuffle(1000).prefetch(buffer_size=tf.data.AUTOTUNE)
val_ds = val_ds.cache().shuffle(1000).prefetch(buffer_size=tf.data.AUTOTUNE)
test_ds = test_ds.cache().shuffle(1000).prefetch(buffer_size=tf.data.AUTOTUNE)
```

Pyth

New sectionBuilding the Model

Creating a Layer for Resizing and NormalizationCreating a Layer for Resizing and Normalization

Before we feed our images to network, we should be resizing it to the desired size. Moreover, to improve model performance, we should normalize the imag pixel value (keeping them in range 0 and 1 by dividing by 256). This should happen while training as well as inference. Hence we can add that as a layer in o Sequential Model.

You might be thinking why do we need to resize (256,256) image to again (256,256). You are right we don't need to but this will be useful when we are done with the training and start using the model for predictions. At that time somone can supply an image that is not (256,256) and this layer will resize it

```python
resize_and_rescale = tf.keras.Sequential([
  layers.experimental.preprocessing.Resizing(IMAGE_SIZE, IMAGE_SIZE),
  layers.experimental.preprocessing.Rescaling(1./255),
])
```

Data Augmentation

Data Augmentation is needed when we have less data, this boosts the accuracy of our model by augmenting the data

```python
data_augmentation = tf.keras.Sequential([
  layers.experimental.preprocessing.RandomFlip("horizontal_and_vertical"),
  layers.experimental.preprocessing.RandomRotation(0.2),
])
```

Applying Data Augmentation to Train Dataset

```python
train_ds = train_ds.map(
    lambda x, y: (data_augmentation(x, training=True), y)
).prefetch(buffer_size=tf.data.AUTOTUNE)
```

Pyth

```python
HTML("""

""")
```

Pyth

Model Architecture

We use a CNN coupled with a Softmax activation in the output layer. We also add the initial layers for resizing, normalization and Data Augmentation.

We are going to use convolutional neural network (CNN) here. CNN is popular for image classification tasks. Watch below video to understand fundamentals CNN

```
HTML("""

""")
```

```
input_shape = (BATCH_SIZE, IMAGE_SIZE, IMAGE_SIZE, CHANNELS)
n_classes = 3

model = models.Sequential([
    resize_and_rescale,
    layers.Conv2D(32, kernel_size = (3,3), activation='relu', input_shape=input_shape),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64,  kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64,  kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
```

```
Model: "sequential_2"
_________________________________________________________________
 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)
```

Compiling the Model

We use adam Optimizer, SparseCategoricalCrossentropy for losses, accuracy as a metric

```
model.compile(
    optimizer='adam',
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),
    metrics=['accuracy']
)
```

```
history = model.fit(
    train_ds,
    batch_size=BATCH_SIZE,
    validation_data=val_ds,
    verbose=1,
    epochs=50,
)
```

```
Epoch 1/50
54/54 [==============================] - 137s 2s/step - loss: 0.8840 - accuracy: 0.4850 - val_loss: 0.7981 - val_accuracy: 0.5417
```

Plotting the Accuracy and Loss Curves

```
history.params
```

```
{'verbose': 1, 'epochs': 50, 'steps': 54}
```

```
history.history.keys()
```

```
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```

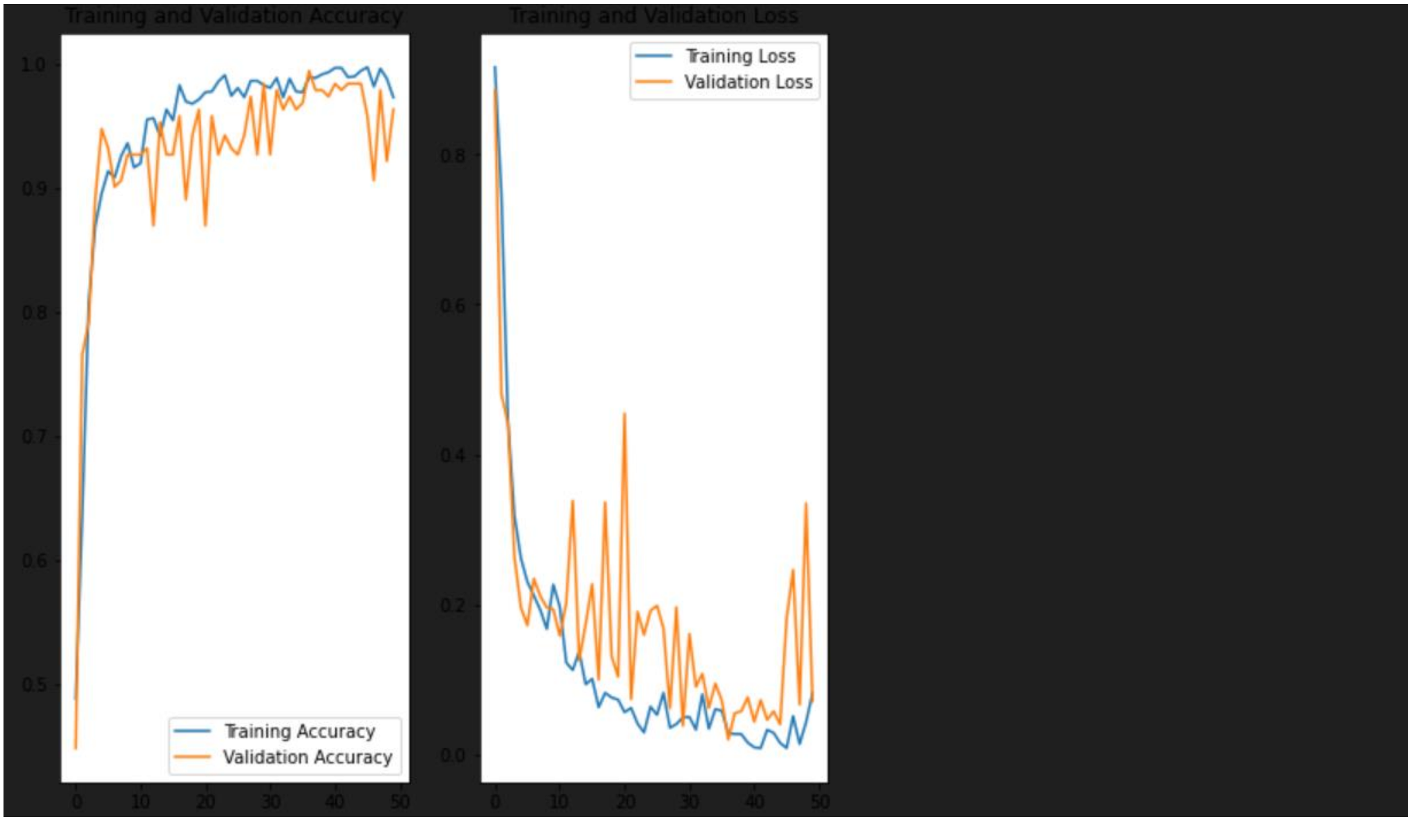
loss, accuracy, val loss etc are a python list containing values of los accuracy etc at the end of each epoch

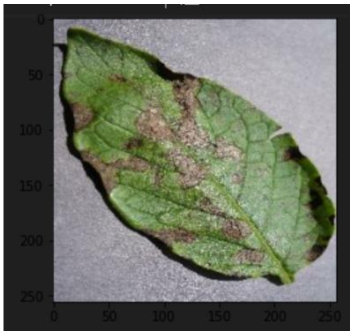
```
type(history.history['loss'])
```

```
0.31873801350593567,
0.261764258146286]
```

```
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

loss = history.history['loss']
val_loss = history.history['val_loss']
```

```
plt.figure(figsize=(8, 8))
plt.subplot(1, 2, 1)
plt.plot(range(EPOCHS), acc, label='Training Accuracy')
plt.plot(range(EPOCHS), val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.subplot(1, 2, 2)
plt.plot(range(EPOCHS), loss, label='Training Loss')
plt.plot(range(EPOCHS), val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()
```


Training and Validation Accuracy
Training Accuracy
Validation Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0
10
20
30
40
50
Training and Validation Loss
Training Loss
Validation Loss
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50



Write a function for inference

Empty markdown cell, double-click or press enter to edit.

```
def predict(model, img):
    img_array = tf.keras.preprocessing.image.img_to_array(images[i].numpy())
    img_array = tf.expand_dims(img_array, 0)

    predictions = model.predict(img_array)
```

```
    predicted_class = Class_names[np.argmax(predictions[0])]
    confidence = round(100 * (np.max(predictions[0])), 2)
    return predicted_class, confidence
```

Now run inference on few sample images

```
plt.figure(figsize=(15, 15))
for images, labels in test_ds.take(1):
    for i in range(9):
        ax = plt.subplot(3, 3, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))
        predicted_class, confidence = predict(model, images[i].numpy())

        actual_class = Class_names[labels[i]]
        plt.title(f"Actual: {actual_class},\n Predicted: {predicted_class}.\n Confidence: {confidence}%")
        plt.axis("off")
```

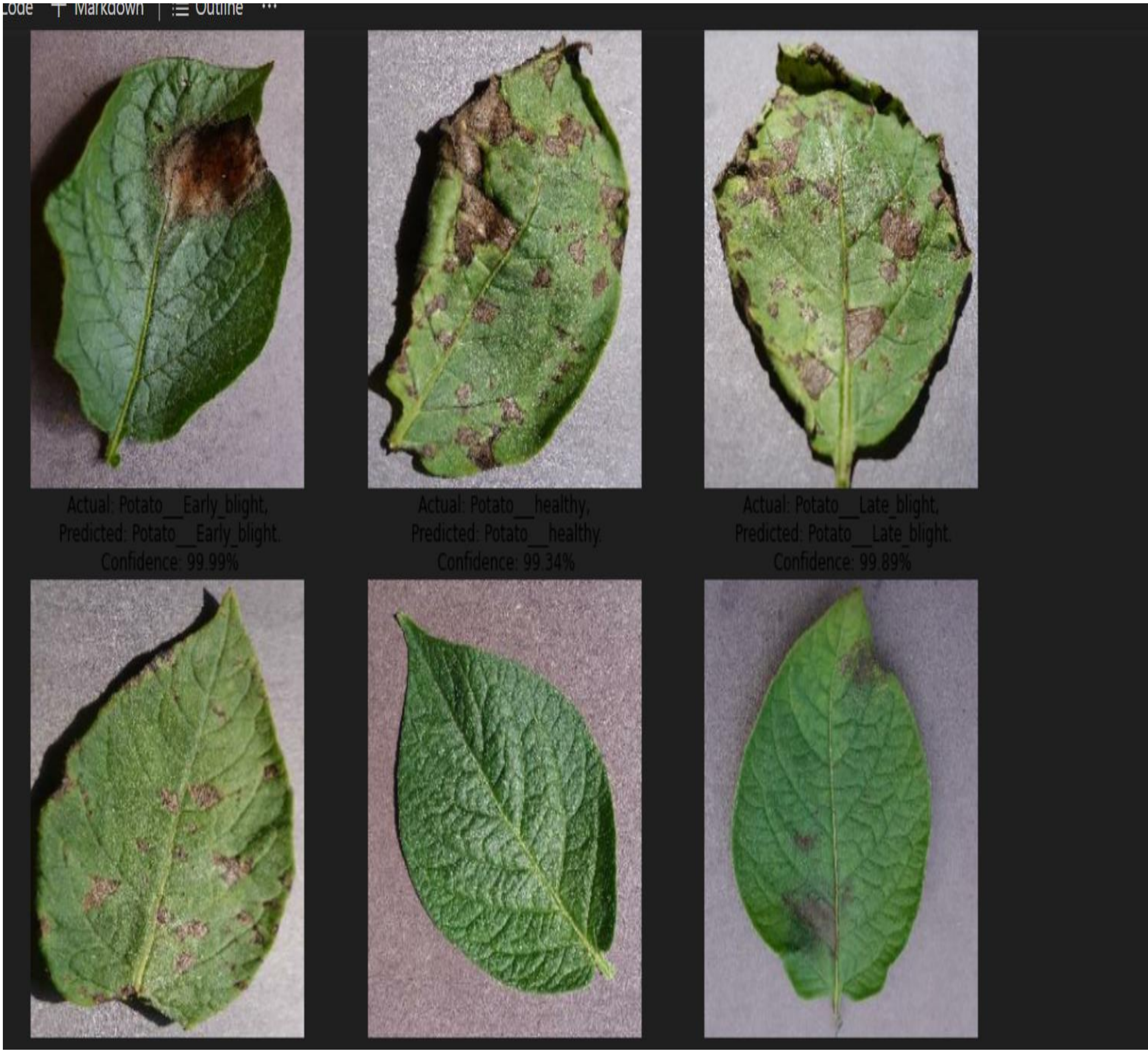
```
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 43ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 36ms/step
```

Run prediction on a sample image

```python
import numpy as np
for images_batch, labels_batch in test_ds.take(1):

    first_image = images_batch[0].numpy().astype('uint8')
    first_label = labels_batch[0].numpy()

    print("first image to predict")
    plt.imshow(first_image)
    print("actual label:",Class_names[first_label])

    batch_prediction = model.predict(images_batch)
    print("predicted label:",Class_names[np.argmax(batch_prediction[0])])
```

```
first image to predict
actual label: Potato___Early_blight
1/1 [==============================] - 1s 755ms/step
predicted label: Potato___Early_blight
```

```
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 43ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 41ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 37ms/step
```



Markdown
Outline
Actual: Potato__Early_blight,
Predicted: Potato__Early_blight.
Confidence: 99.99%
Actual: Potato__healthy,
Predicted: Potato__healthy.
Confidence: 99.34%
Actual: Potato__Late_blight,
Predicted: Potato__Late_blight.
Confidence: 99.89%

Chapter-5
Conclusion

In this paper, we suggested a model for categorising illnesses of potato leaves based on convolutional neural networks. We evaluated the accuracy and loss of the performance while taking potato leaf diseases into account. According to the evaluation results, our recommended classification model can identify particular potato leaf diseases. With 99.91% accuracy during training, 99.80% accuracy during validation, and 100% accuracy during testing, it was found that the suggested model outperformed the competition. This model will assist in classifying the specific potato leaf disease so that appropriate remedial treatment can be taken. We shall carry on our research in an effort to stop Bangladesh's annual significant loss in potato production. A web-based or Android app that will help potato producers will be the subject of future research. We anticipate a wide range of favourable consequences on agricultural production and global food security as a result of our efforts.

The global coronavirus pandemic has greatly raised global demand for potatoes. In this essay, we have categorised the leaf diseases that affect potato plants. Effective features for the picture classification of leaf diseases may be studied using the VGGNetwork (VGG16 and VGG19). A more reliable system would result from the dataset's addition of the data augmentation process. An average accuracy of 91–93% can be attained with our suggested strategy, according to experiments. In terms of agriculture and global food security, we think this effort has a lot to offer.

References ::

[1] UNICE, Food Loss And Waste; 2017.

[2] Editor; Mahfuz Anam, The Daily Star. Available:https://www.thedailystar.net/business/economy/news/potato-farmers-lose-tk-2500cr-year-fao2987086), access date 28/22/2022

[3] Drewnowski A, Rehm CD. Vegetable cost metrics show that potatoes and beans provide most nutrients per penny. PLoS One. 2013;8(5):e63277.

[4] Ilbery B, Maye D, Little R. Plant disease risk and grower–agronomist perceptions and relationships: an analysis of the UK potato and wheat sectors. Appl. Geogr. 2012;34:306–315.

[5] TM P, Alla P, Ashirta KS, Chittaragi NB, Koolagudi SG. Tomato leaf disease detection using convolutional neural network. International Conference on Contemporary Computing (IC3); 2018.

[6] Hokkaido Agricultural Research Center, N.: HARC brochure; 2017. Available:http://www.naro.affrc.go.jp/publicity report/publication/files/2017NARO english 1.pdf

[7] Faria FA, dos Santos JA, Rocha A, da S. Torres, R. Automatic classifier fusion for produce recognition. In: Proceedings of the 25th International Conference on Graphics, Patterns and Images (SIBGRAPI2012). 2012;252–259.

[8] Geetharamani G, Arun Pandian J. Identification of plant leaf diseases using a nine-layer deep convolutional neural network. Comput. Electr. Eng. 2019;76:323–338.

[9] B. Mitschang et al. (Hrsg.): BTW– Workshopband, Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2017 79 A Deep Learning-based Approach for Banana Leaf Diseases Classification Jihen Amara,1 Bassem Bouaziz, 2 and Alsayed Algergawy; 2017.

[10] Shen Weizhong, Wu Yachun [at.el]. Grading method of leaf spot disease based on image processing. IEEE. 2008;491-494.

[11] Appasaheb Gargade SA. Khandekar. A review: Custard apple leaf parameter analysis and leaf disease detection using digital image processing. 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC). 2019;267-271.

[12] Srichocksittikul P, Nimsuk N. Design of Deep Learning Architecture for Classification of Orchid Diseases. 2021 18th International Conference on Electrical

Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). 2021;904-907. DOI: 10.1109/ECTI-CON51831.2021.9454869

[13] Rabbia Mahum, Haris Munir, Zaib-Un-Nisa Mughal, Muhammad Awais, Falak Sher Khan, Muhammad Saqlain, Saipunidzam Mahamad & Iskander Tlili. A novel framework for potato leaf disease detection using an efficient deep learning model, Human and Ecological Risk Assessment: An International
Journal; 2022.

DOI: 10.1080/10807039.2022.2064814

[14] Chakraborty KK, Mukherjee R, Chakroborty C, Bora K. Automated recognition of optical image based potato leaf blight diseases using deep learning. Physiological and Molecular Plant Pathology. 2022;117: 101781.

[15] Chen W, Chen J, Zeb A, Yang S, Zhang D. Mobile convolution neural network for the recognition of potato leaf disease images. Multimedia Tools and Applications. 2022;1-20.

[16] Hinton GE, Osindero S, The YW. A fast learning algorithm for deep belief nets. Neural Comput. 2006;18(7):1527–1554.

[17] Le QV. Building high-level features using large scale unsupervised learning. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing. 2013;(ICASSP-2013):8595–8598.

[18] Kang L, Kumar J, Ye P, Li Y, Doermann D. Convolutional neural networks for document image classification. In: Proceedings of 22nd International Conference on Pattern Recognition 2014;(ICPR2014):3168–3172.

[19] Lee H, Grosse R, Ranganath R, Ng AY. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th Annual International Conference on Machine Learning. 2009;609–616.

[20] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436.