Comparative Study Between Different Architecture Of CNN For Potato Diseases Classification

April 18, 2023

Project Report submitted for

Modelling Workshop

by

TEAM 4

Abstract

At the present, industry professionals are frequently required to identify potato infections directly, which is a process that is both time demanding and prone to errors.[2] It requires a significant amount of time and effort to examine each plant for illness and, if it does, to identify the specific type of illness that the plant is suffering from. In addition, not every form of illness can be accurately recognised by simply using a person's naked sight. As a direct consequence of this, producers are already making use of artificial intelligence in order to fully systematise the process of diagnosing plant diseases. An automated system that makes use of computer vision to provide producers with assistance in recognising plant illnesses.As a result, we have recommended using artificial intelligence to diagnose plant diseases and to automate the complete system. To do this, we have proposed a deep learning model as well as several Transfer Learning models. In order to determine which of CNN's two alternative architectures delivers the most accurate results, we will carry out a comparative research study contrasting the two of them. A CNN could be educated using a sizeable data set consisting of photographs of healthy and diseased potato leaves in order to gain an understanding of the characteristics of each type of sickness. CNN would become skilled at recognising typical symptoms that indicate illness in the photographs, such as withering, spot patterns, or discoloration.Because agriculture is the primary source of revenue for approximately 70 percent of India's population and the potato is one of the most widely grown and consumed vegetables on a global scale, the initiative is significant because of these two factors. The fact that numerous diseases can have a detrimental influence on the quantity and quality of potato crops highlights the importance of timely disease discovery and efficient disease management. The application of CNNs for the purpose of picture classification has seen widespread adoption across a variety of industries, including agribusiness. However, there is a dearth of research on the quantitative analysis of various CNN architectures for the categorization of potato diseases. The purpose of this project is to close this knowledge gap by analysing how well VGG-16 and AlexNet execute on this specific assignment. The findings of this initiative will allow farmers and researchers to more accurately pinpoint potato diseases, which will ultimately lead to disease management techniques that are more efficient and effective.

Contents

1 Introduction

Agriculture is India's most important economic sector. Agriculture is the principal means of subsistence for approximately 70 percent of the world's inhabitants[17]. Agriculture is absolutely necessary to the functioning of the economics of the countries. Additionally, it has an impact on the Gross Domestic Product. Therefore, it is of the utmost importance to detect infections in the agricultural industry as quickly as humanly feasible. [1]. Infectious diseases affecting plants are just one of the many challenges that producers face. The process of diagnosing the illness before it worsens and extends to other plants across the field is a massive undertaking in and of itself. Even though there is an ever-increasing demand for food due to a rising population, the number of people electing to make their livings in agriculture has been gradually falling over the course of history. Potatoes are one of the vegetables that are produced on the most land and eaten the most frequently all over the globe [15].A variety of sicknesses, on the other hand, have the potential to have a negative impact on both the quantity and quality of the potato harvest. The timely diagnosis of sicknesses is essential to the efficient administration of potato production because of its dependence on this factor.
The use of Convolutional Neural Networks (CNNs) has become widespread in many different industries, including agriculture, due to their effectiveness as instruments for the categorization of images [9]. The purpose of this research is to investigate and evaluate the efficacy of various CNN architectures in the categorization of potato diseases. We are going to investigate how well various models categorise potato illnesses and try to determine which model is the most suitable for tackling this problem.
This project's objective is to assist farmers and researchers in more accurately identifying potato diseases in the field, which will enable for disease management techniques that are both more efficient and successful. We expect to be able to shed light on the most effective method for disease categorization in potatoes using CNNs by analysing the differences and similarities between various CNN architectures.
The majority of conventional machine learning algorithms were developed in a laboratory setting, and as a result, their level of robustness is insufficient to satisfy the requirements of practical agricultural applications. These days, deep learning (DL) techniques, particularly those that are based on convolutional neural networks (CNNs), are getting widespread application in the agricultural field for detection and classification tasks. Some examples of these types of tasks include weed detection crop pest categorization, and plant disease identification likewise Apple Leaf Disease Identification and Classification [4], Identification of rice diseases [8], Mango Leaf Diseases Identification [12]. The study of DL falls under the umbrella of the field of machine learning. Traditional methods of machine learning have issues with poor performance, a dearth of actual picture, and segmented operation . These issues have been addressed or partially solved by this technique. The ability of DL models to identify features without making use of segmented operations while still achieving satisfactory performance is one of the most significant benefits of these models. The original data is mined for the purpose of automatically extracting the characteristics of an object. In 1980, Kunihiko Fukushima presented the Neocognitron [18], which would later serve as an inspiration for CNNs . The development of CNNs has resulted in a marked improvement in the accuracy and speed of the technology used to classify plant diseases.
Throughout the course of this research, we will investigate and evaluate the efficacy of a variety of CNN architectures, including VGG-16 and AlexNet, for the categorization of potato diseases. We will train and evaluate each model using the standard evaluation criteria, and the dataset of potato disease images that we will use is one that is publicly accessible.

1.1 Deep learning

DL stands for deep learning and is a subfield of machine learning. Its primary applications are in the areas of picture categorization, object detection, and natural language processing. DL (or Deep Learning) is an algorithm for the automated selection of data features that is built on a neural network. There is not a great deal of manufactured feature engineering that is required for it. It does this by combining low-level features into abstract high-level features in order to identify distributed characteristics and properties of sample data. Image identification and target detection are two areas in which it excels over more conventional approaches, both in terms of its precision and its capacity for generalisation. The multilayer perceptron, the CNN, and the recurrent neural network are the three most common kinds of networks used today. CNN is the method that is utilised the most frequently for the categorization of plant foliage diseases. Other types of deep learning networks, such as completely convolutional networks (FCNs) and deconvolutional networks, are typically used for picture segmentation or medical diagnosis, but not for the categorization of plant leaf diseases.

1.2 General overview of CNN

CNN stands for Convolutional Neural Network, which is a type of deep learning architecture commonly used for image recognition, classification, and other computer vision tasks.

- Generally following layers are in CNN:

  1. **Convolutional layer:** This layer applies a convolution operation to the input data, which involves sliding a filter (a small matrix of weights) over the input and computing dot products between the filter and local patches of the input. The resulting feature maps highlight different aspects of the input, such as edges, corners, and textures. Convolutional layers learn these filters through backpropagation during training, and they can have multiple filters to capture different features.
  2. **Pooling layer:** This layer reduces the spatial size of the feature maps by downsampling them, typically using max pooling or average pooling. Max pooling takes the maximum value in each non-overlapping region of the feature map, while average pooling takes the average value. Pooling helps to make the features more invariant to small translations and distortions in the input, and it also reduces the number of parameters in the model.
  3. **Fully connected layer:** This layer connects every neuron in the previous layer to every neuron in the current layer, like a traditional neural network. The fully connected layers are typically used at the end of the CNN to transform the high-level features learned by the convolutional and pooling layers into class scores or regression outputs. The fully connected layers can have a large number of parameters, which makes them more prone to overfitting and requires regularization techniques such as dropout.

- Overview of process:

  1. The first layer is typically a convolutional layer, which applies filters to the input image to extract features
  2. The resulting feature maps are then passed through a pooling layer, which reduces their size by downsampling.
  3. The process of applying convolutional and pooling layers is repeated several times to extract increasingly abstract and high-level features from the input image.
  4. Finally, the feature maps are flattened and fed into one or more fully connected layers, which perform the classification or regression task.

One of the key advantages of CNNs is that they can learn hierarchical representations of the input data, where lower-level features such as edges and corners are combined to form higher-level features such as shapes and textures. This allows CNNs to achieve high accuracy on image recognition tasks, often surpassing human performance.

1.3 Image Classification Problem

Image classification is a popular task in deep learning, where the goal is to classify an input image into one of several predefined categories or classes. This task has many applications, such as in medical diagnosis, self-driving cars, and security systems. To solve an image classification problem, a deep learning model is trained on a labeled dataset of images, where each image is labeled with its corresponding class. The model learns to extract relevant features from the input image and use them to predict the correct class. model is designed to automatically learn hierarchical representations of the input image, from low-level features such as edges and corners, to high-level features such as shapes and textures.During training, the model is optimized to minimize a loss function that measures the difference between the predicted class and the true class of the input image. This optimization is typically performed using a variant of gradient descent called backpropagation. Once the model is trained, it can be used to classify new images that were not included in the training set. The input image is fed into the model, and the output is a probability distribution over the possible classes. The predicted class is the one with the highest probability.

2 Literature survey

A wide variety of research projects have been carried out in the areas of picture categorization and identification. For example, numerous studies have been carried out on the classification of plant diseases using CNNs. The CNN architecture known as Vgg-16 performed the best when put through a series of tests designed to classify illnesses that can affect plant leaves, as measured by the results of one research [1]. Vgg-16 was found to have the greatest accuracy out of all the pre-trained models that were tested using transfer learning to identify plant diseases that were compared in [3]. examined several different deep learning algorithms for the identification and diagnosis of plant diseases. ResNet models were utilised in [4] for the purpose of identifying and classifying apple foliage diseases.[5] examined the effectiveness of CNN and AlexNet in the identification of illness in potato and mango leaf samples, and discovered that CNN performed more effectively.[5] The effectiveness of three different convolutional neural network architectures for the categorization of plant leaf illnesses is analysed and compared in this scholarly article.The authors trained Vgg-16, MobileNet, and ConvNext models on a dataset comprising 20,639 images that were categorised into 15 different illnesses using transfer learning. The dataset was divided into 15 different classes.During the training process, they discovered that Vgg-16 had the greatest accuracy, which was measured at 0.97, while MobileNet was the most efficient in terms of time, requiring only 62 seconds.The findings of this research indicate that artificial intelligence may have application in the diagnosis and treatment of illnesses that affect plant life.[6] The research article authored by B. Chellapandi and S. Chopra analyses and contrasts the performance of several different pre-trained models that make use of transfer learning to identify plant diseases.In order to differentiate between pictures of healthy and diseased vegetation, the research project made use of five pre-trained models, including Inception-v3, ResNet50, VGG16, DenseNet121, and Xception.The dataset that was utilised was the PlantVillage dataset, which includes over 54,000 pictures of plant foliage affected by 38 distinct classifications of diseases.According to the findings, the Inception-v3 and DenseNet121 models performed significantly better than the rest of the models in terms of precision and F1-score respectively.The researchers came to the conclusion that using pre-trained models in conjunction with transfer learning can be an efficient method for illness identification in plants. [7] In the research article authored by M. N. Saranya, the topic of utilising neural networks for the purpose of illness detection in banana leaves and products is discussed.A multilayer perceptron neural network and input pictures of banana foliage and fruits with various diseases were used in the research to achieve an average accuracy of 95 percent in identifying the illnesses.This presents a discussion on the significance of disease detection in agricultural settings, the components and stages of the suggested system, and the benefits of employing image processing techniques in conjunction with artificial neural networks for disease categorization.[8] In conventional approaches to object recognition, the complete picture is typically analysed at a variety of scales and locations in order to locate the target objects, which can be a computationally intensive process. On the other hand, ROI pooling and RPN scans enable selective scanning of regions of interest within a picture, thereby decreasing the number of regions that need to be processed. This can result in faster overall processing times.In order to generate prospective object bounding boxes within a picture, RPN scans are utilised.The RPN starts with the image that is being fed into it and produces a collection of region proposals, each of which symbolises a possible object that is contained within the image. A succeeding stage in the object recognition pipeline will then refine these region suggestions in order to generate more accurate bounding boxes for the objects being searched for.On the other hand, ROI pooling is a technique that is utilised in order to extract features from particular sections within a picture. It accepts as input both the suggested object bounding boxes that were generated by the RPN and the feature map that was generated by the CNN. [9] Our system uses the K-Means Clustering Method just like existing systems, but it has a new added feature called alerting. After the disease is detected, our system informs the farmer as soon as possible to take care of the disease and try to minimises the disease spreading. In order to accomplish this, a buzzer is connected to the existing system. As a result, the spreading of diseases is controlled as early as possible, which saves time and improves the yield production of the crop and also reduces This has the potential to be of great assistance to Indian agriculture. [10]Existing systems use the K-Means Clustering Method, but our system has a new added feature called alerting; after a disease is detected, our system informs the farmer as soon as possible to take care of the disease and try to minimises the disease spreading; in order to accomplish this, a buzzer is connected to the existing system; as a result, the spreading of diseases is controlled as early as possible, which saves time and improves the yield production of the crop and the overall quality of the crop. This has the potential to be of great assistance to Indian agriculture.

S.No	**YOP**	**Citation**	**Remark**
1	2022	F. Al Heeti and M. Ilyas, ”Comparative analysis of convolutional neural network architectures for classification of plant leaf diseases,” 2022 2nd International Conference on Computing and Machine Intelligence (ICMI), Istanbul, Turkey, 2022, pp. 1-5, doi: 10.1109/ICMI55296.2022.9873752.	The authors talks about transfer learning to train Vgg-16, MobileNet, and ConvNext models.
2	2020	X. Li, L. Rai, and I. Engineering, “Apple Leaf Disease Identification and Classification using ResNet Models,” pp. 738–742, 2020.	discusses the use of ResNet models for identifying and classifying apple leaf diseases
3	2019	S. Arya and R. Singh, ”A Comparative Study of CNN and AlexNet for Detection of Disease in Potato and Mango leaf,” 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), Ghaziabad, India, 2019, pp. 1-6, doi: 10.1109/ICICT46931.2019.8977648.	study indicate that the CNN model outperformed AlexNet in terms of accuracy and speed in both potato and mango leaf disease detection
4	2019	arif, I. M. Dheir, A. Soliman, A. Mettleq, and S. S. Abunaser, “Potato Classification Using Deep Learning,” Int. J. Acad. Pedagog. Res., vol. 3, no. 12, pp. 1–8, 2019.	The paper demonstrates the effectiveness of using deep learning techniques for classifying different types of crops based on their visual features, which has potential applications in precision agriculture and food processing industries.

5	2018	B. Chellapandi and S. Chopra, "	The authors compare the performance of different pretrained models for detecting plant diseases on a dataset of plant images with and without diseases.
6	2018	S.Arivazhagan, S.Vineth Ligi, "Mango Leaf Diseases Identification Using Convolutional Neural Network", International Journal of Pure and Applied Mathematics" vol.120, No. 6,11067-11079, ISSN: 1314-3395, August 2018.	The authors propose a system for identifying various diseases that affect mango leaves using a convolutional neural network (CNN). The system involves the acquisition of high-quality images of mango leaves affected by various diseases, which are then preprocessed and fed into the CNN.
7	2018	K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis", Computers and Electronics in Agriculture, vol. 145, pp. 311- 318, 2018.	Deep learning models have shown promising results in plant disease detection and diagnosis, and that they have the potential to revolutionize the field of precision agriculture by enabling early disease detection and targeted treatments.
8	2018	S. Zhang, W. Huang and C. Zhang, "Tree-channel convolutional neural networks for vegetable leaf disease recognition", Cognitive Systems Research, 2018.	The authors propose a new type of convolutional neural network (CNN) architecture called the tree-channel CNN for recognizing and classifying vegetable leaf diseases.
9	2018	H. Yanagisawa, T. Yamashita, and H. Watanabe, "A Study on Object Detection Method from Manga Images using CNN," ieee-2018 Int. Work. Adv. Image Technol., pp. 7–10, doi: 10.1109/IWAIT.2018.8369633.	The paper shows the applicability of deep learning techniques beyond traditional computer vision tasks and highlights the potential of using them for analyzing and understanding various forms of visual media.
10	2017	A. A. M. Al-Saffar, H. Tao and M. A. Talab, "Review of deep convolutional neural network in image classification," 2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET), Jakarta, Indonesia, 2017, pp. 26-31, doi: 10.1109/ICRAMET.2017.8253139.	The paper provides a review of deep convolutional neural networks used for image classification

11	2017	Yang Lu, Shujuan Yi, Nianyin Zeng, Yurong Li, Yong Zhang, "Identification of rice diseases using deep convolutional neural networks" , Neurocomputing 267 (2017) 378–384.	The article discusses the use of deep convolutional neural networks for identifying diseases in rice plants.
12	2017	W. Rawat and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," in Neural Computation, vol. 29, no. 9, pp. 2352-2449, Sept. 2017, doi: 10.1162/neco$_{a0}$0990.	The authors provide a comprehensive review of deep convolutional neural networks for image classification. The paper discusses various techniques used in deep learning and the different architectures of deep convolutional neural networks that have been developed for image classification.
13	2017	Neena Aloysius and Geetha M , "A Review on Deep Convolutional Neural Networks", International Conference on Communication and Signal Processing, April 6-8, 2017.	The paper provides a comprehensive review of the current state of CNNs and their impact on the field of machine learning.
14	2016	Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton "ImageNet Classification with Deep Convolutional Neural Networks", International Conference on Neural Information Processing Systems - Volume 1, June 2016	It presents the architecture of a deep convolutional neural network (CNN) that achieved state-of-the-art performance on the ImageNet large-scale visual recognition challenge.
15	2007	Arpita Patel, Mrs. Barkha Joshi, "A Survey on the Plant Leaf Disease Detection Techniques", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 6, Issue 1, ISO 3297:2007	The article discusses various techniques used for detecting diseases in plant leaves.

3 Problem Definition

The aforementioned research shows that CNNs have seen widespread application in the field of plant disease categorization, and a number of different architectures for CNNs have been suggested. On the other hand, to the best of our knowledge, there is a dearth of research on the comparative analysis of various CNN architectures for the categorization of potato diseases. Comparing the efficacy of Vgg-16 and AlexNet for potato illness classification is the focus of this research project, the overarching goal of which is to close this knowledge deficit.

3.1 Research Question

The purpose of this project, which is labelled "Comparative Study Between Different Architecture Of CNN For Potato Diseases Classification," is to investigate and evaluate the efficacy of a number of different convolutional neural network (CNN) architectures with regard to the categorization of potato illnesses. The following items make up the particular goals of this project:

- **RQ1** Which is the better CNN architectures for classifying potato diseases VGG-16 or AlexNet?
- **RQ2** What is the behavior of each CNN architectures is it Overfit, Underfit or Goodfit?
- **RQ3** What is the advantage of using these computer vision techniques then tranditional techniques?

4 Domain Knowledge

Potato crops are susceptible to various diseases, including early and late blight, which can cause significant yield losses.

4.1 Diseases

Figure 1: Early Blight

Figure 2: Late Blight

4.1.1 Early Blight

The parasite Alternaria solani is responsible for causing early blight, which is a fungal illness.The leaves develop a pattern of concentric rings that begin as small, dark patches and progressively grow larger over the course of the disease.Leaves that have been infected may turn yellow and ultimately perish.Additionally, the pathogen is able to attack the petioles, stalks, and tubers of the plant.In the absence of appropriate management, early blight can result in substantial production losses.

4.1.2 Late Blight

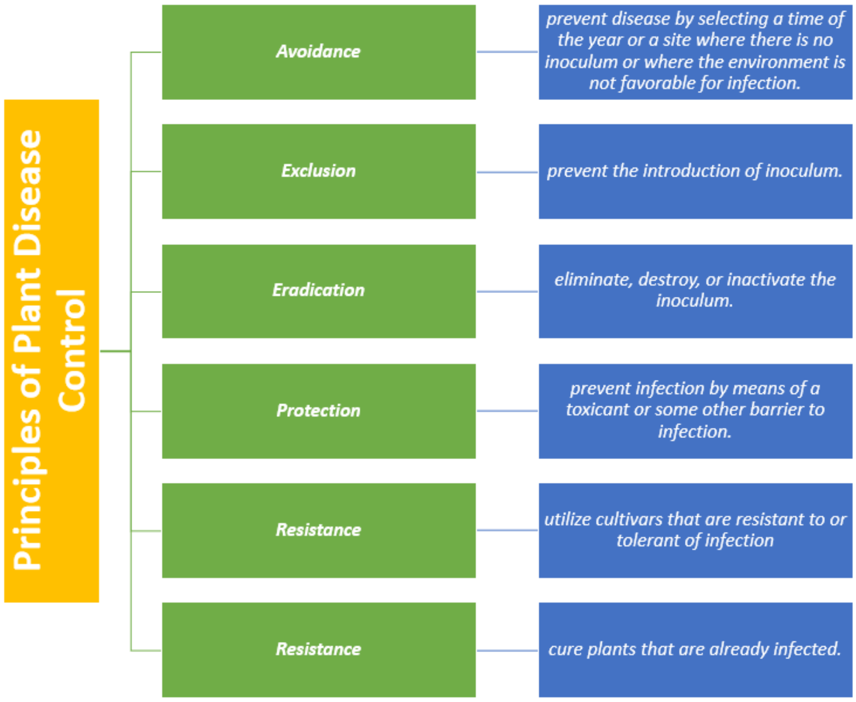
Late blight is a destructive disease caused by the oomycete pathogen Phytophthora infestans. Symptoms include dark, water-soaked lesions on the leaves that rapidly expand and turn brown. Infected leaves may die within days. The pathogen can also infect the stems, petioles, and tubers. Late blight can cause significant yield losses and can spread rapidly under favorable environmental conditions

4.2 Management Strategies

The accurate diagnosis of illnesses at the appropriate time is absolutely necessary for the effective administration of potato production.Consequently, the following are some of the managerial techniques that are presently being used:

- The control of early and late blight calls for a coordinated strategy that incorporates societal, chemical, and biological practises.
- The practise of agricultural rotation, the elimination of infected plant detritus, and the use of resistant varieties are all examples of cultural techniques. item The use of fungicides, which can be either prophylactic or curative depending on the situation, is one example of a chemical technique.

- The use of biocontrol agents, such as bacteria and fungus, which can inhibit the development of the pathogens as well as their ability to disseminate is included in the category of biological techniques.



5 Challenges

- Symptoms of a complicated disease: Potato plants are susceptible to a wide variety of illnesses, many of which exhibit symptoms that are interchangeable, making it difficult to differentiate between them. For instance, the signs and symptoms of late blight, early blight, and various other fungal conditions can coincide, which makes it difficult to accurately categorise them.

- The indications of plant diseases can change depending on a number of environmental variables, including temperature, humidity, and the amount of light that is present in the area. As a result, it is absolutely necessary to have a reliable algorithm that can take into consideration the various aspects of the surrounding environment and provide precise categorization.

- There are only so many datasets available, and finding ones of sufficient quality to use in the training and evaluation of AI algorithms can be a substantial challenge. To ensure that the algorithm can accurately recognise a variety of illnesses, the datasets need to be exhaustive, diversified, and accurately labelled.

- Overfitting is a problem that can occur in artificial intelligence models, in which the models do well on the training dataset but are unable to generalise their performance to new data. It is crucial to prevent overfitting by utilising appropriate methods such as data supplementation, regularization, and cross-validation. This is an important step.

- Hardware and computational requirements: Training deep learning models for illness categorization requires a substantial amount of computational resources, including high-end GPUs and large quantities of memory. These requirements must be met before the models can be used. Researchers who have limited access to these materials may therefore find this to be a substantial challenge.

6 Methodology

Below is the flowchart for the methodology used:

Now each step of methodology is explained in detail:

6.1 Dataset Collection:

The first thing that needs to be done is to gather a collection of potato disease pictures, which will later be used to both train and evaluate the CNN models. The dataset needs to have the appropriate amount of images for each illness categorization, and those pictures need to have the right names attached to them. The name of the information that we are making use of for this endeavour is "PlantVillage."`https://www.kaggle.com/datasets/emmarex/plantdisease` It includes 2152 pictures that are divided into three categories, which are titled "Potato healthy," "Potato early blight," and "Potato late blight." The information was separated into 23 folders, with one designated for instruction, another for validation, and a third for testing.

1. Training Dataset that will be utilised during training for the object in question
2. Validation Dataset: A dataset that will be evaluated in comparison to the object being trained evaluate:
3. Test Dataset that will be used to evaluate a model against after it has been trained.

6.2 Data Preprocessing:

The dataset should be preprocessed to ensure that the images are of the same size, and the background is removed. The images should also be normalized to ensure that they are in the same range of values. installed the Keras library, which contains various transfer learning models such as Vgg-16 and AlexNet. After installing the Keras library, we are augmenting the data that we use in this work, Data Augmentation is needed when we have less data, this boosts the accuracy of our model by augmenting the data. which consists of about 2000 images distributed across 3 classes.

6.3 Model Selection:

We will compare the performance of different CNN architectures for potato disease classification, such as VGG-16, and AlexNet. These architectures are known to work well for image classification tasks. convolutional neural network (CNN) architectures used in deep learning for image classification tasks. However, they differ in terms of their architecture and performance

6.3.1 VGG16

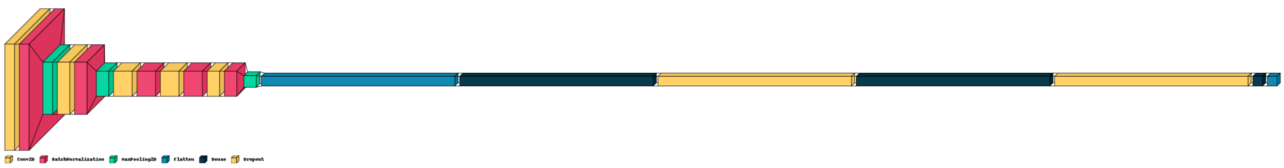
is a deep and powerful CNN architecture,is a deep convolutional neural network architecture that was developed by the Visual Geometry Group at the University of Oxford. It was proposed as an entry in the 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and achieved state-of-the-art results at the time.

Layer	Output	Parameter
conv2d (Conv2D)	"(None, 227, 227, 64) "	1792
conv2d_1 (Conv2D)	"(None, 227, 227, 64)"	36928
max_pooling2d	"(None, 113, 113, 64)"	0
conv2d_2 (Conv2D)	"(None, 113, 113, 128) "	73856
conv2d_3 (Conv2D)	"(None, 113, 113, 128)"	147584
max_pooling2d_1	"(None, 56, 56, 128) "	0
conv2d_4	"(None, 56, 56, 256)"	295168
conv2d_5	"(None, 56, 56, 256) "	590080
conv2d_6	"(None, 56, 56, 256) "	590080
max_pooling2d_2	"(None, 28, 28, 256)"	0
conv2d_7	"(None, 28, 28, 512)"	1180160
conv2d_8	" (None, 28, 28, 512) "	2359808
conv2d_9	"(None, 28, 28, 512) "	2359808
max_pooling2d_3	"(None, 14, 14, 512) "	0
conv2d_10	"(None, 14, 14, 512)"	2359808
conv2d_11	"(None, 14, 14, 512) "	2359808
conv2d_12	"(None, 14, 14, 512) "	2359808
max_pooling2d_4	"(None, 7, 7, 512) "	0
flatten	"(None, 25088) "	0
dense	"(None, 4096) "	102764544
dropout	"(None, 4096) "	0
dense_1	"(None, 4096) "	16781312
dropout_1	"(None, 4096)"	0
dense_2	"(None, 5) "	20485

Table 2: VGG16 Model Summary

Architecture

The architecture of VGG16 consists of 16 layers, including 13 convolutional layers and 3 fully connected layers. The convolutional layers have small 3x3 filters and the depth of the network increases gradually from 64 to 512 filters in the last few layers. The network also uses max-pooling layers after every two or three convolutional layers to reduce the spatial dimensions of the feature maps.



Key Feautures

1. **Small convolutional filters**: VGG16 uses small 3x3 convolutional filters, which helps to capture finer details in the images
2. **Max-pooling layers**: VGG16 uses max-pooling layers after every two or three convolutional layers, which reduces the spatial dimensions of the feature maps while preserving the important features.
3. **Uniform architecture**: VGG16 has a uniform architecture where all the convolutional layers have the same number of filters and are followed by a max-pooling layer, which makes it easier to train and optimize the network.

Layer	Output	Parameter
conv2d (Conv2D)	"(None, 55, 55, 96)"	34944
batch_normalization	"(None, 55, 55, 96) "	384
max_pooling2d	"(None, 27, 27, 96) "	0
conv2d_1 (Conv2D)	"(None, 27, 27, 256) "	614656
batch_normalization_1	"(None, 27, 27, 256) "	1024
max_pooling2d_1	"(None, 13, 13, 256)"	0
conv2d_2	"(None, 13, 13, 384) "	885120
batch_normalization_2	"(None, 13, 13, 384) "	1536
conv2d_3	"(None, 13, 13, 384)"	1327488
batch_normalization_3	"(None, 13, 13, 384) "	1536
conv2d_4	"(None, 13, 13, 256)"	884992
batch_normalization_4	"(None, 13, 13, 256)"	1024
max_pooling2d_2	"(None, 6, 6, 256) "	0
flatten	"(None, 9216) "	0
dense	" (None, 4096) "	37752832
dropout	"(None, 4096)"	0
dense_1	"(None, 4096) "	16781312
dropout_1	"(None, 4096)"	0
dense_2	"(None, 5)"	20485

Table 3: AlexNet Model Summary

4. **Deep architecture**: VGG16 has a deep architecture with 16 layers, including 13 convolutional layers and 3 fully connected layers. This allows it to learn more complex and abstract features from the input images.

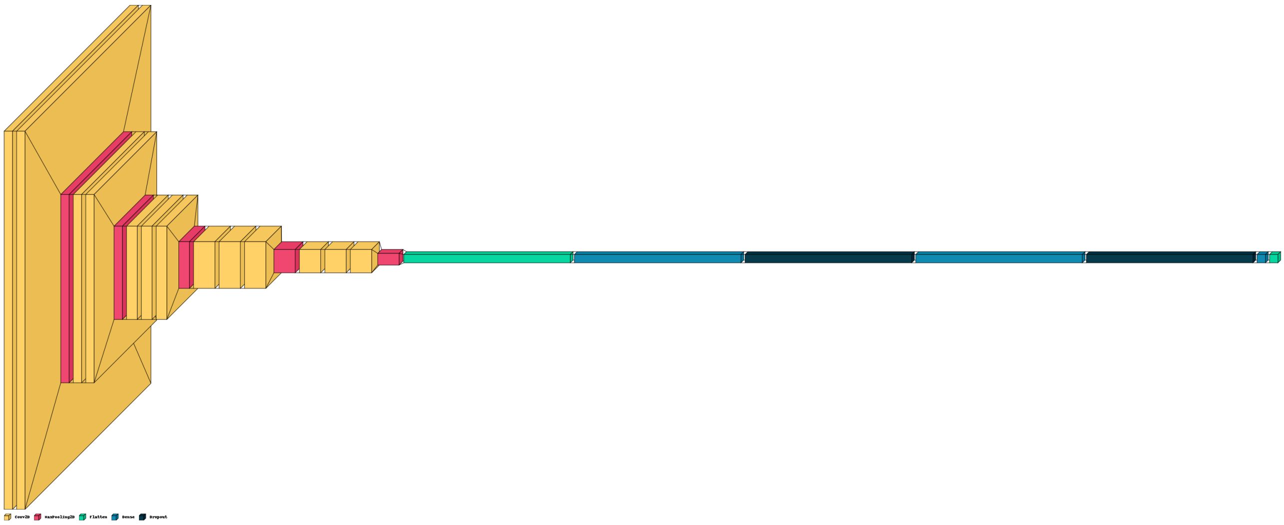
It is a powerful and effective neural network architecture that has made significant contributions to the field of computer vision.

6.3.2 AlexNet

is a deep convolutional neural network architecture that was developed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. It was the winning entry in the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC), where it achieved a significant improvement in the accuracy of image classification compared to previous methods.

Architecture

The architecture of AlexNet consists of eight layers, including five convolutional layers and three fully connected layers. The convolutional layers use larger 11x11 filters with a stride of 4 pixels, which allows for the detection of larger and more complex features. The network also uses overlapping max-pooling layers after every two convolutional layers to reduce the spatial dimensions of the feature maps.



Key Features

1. **Large convolutional filters**: Our AlexNet model consists of five convolutional layers and three fully connected layers. The architecture of the AlexNet model is as follows:

   (a) Input Layer: The input layer of AlexNet accepts a 227 x 227 x 3 image as input. The image is passed through a pre-processing step, where the pixel values are centered and normalized.

   (b) Convolutional Layers: The AlexNet model has five convolutional layers, with each layer having different filter sizes and numbers of filters[19]. The first convolutional layer has 96 filters of size 11 x 11 with a stride of 4. The second convolutional layer has 256 filters of size 5 x 5 with a stride of 1. The third convolutional layer has 384 filters of size 3 x 3 with a stride of 1. The fourth and fifth convolutional layers have 384 and 256 filters of size 3 x 3 with a stride of 1, respectively.

   Each convolutional layer is followed by a rectified linear unit (ReLU) activation function and a max-pooling layer. The max-pooling layer has a pool size of 3 x 3 with a stride of 2. Max-pooling reduces the spatial dimensions of the feature maps while preserving the important features.

   (c) Fully Connected Layers: After the five convolutional layers, the output is passed through three fully connected layers. The first fully connected layer has 4,096 neurons, while the second fully connected layer has 4,096 neurons. The last fully connected layer has 1,000 neurons, which correspond to the 1,000 categories in the ImageNet dataset. The final layer uses the softmax function to classify the input image into one of the 1,000 categories.

   Training:

   The AlexNet model was trained on NVIDIA GPU for one day. We used the adam optimizer to optimize our algorithm.We used this optimizer because of its fast convergence rate.We studied the comparison of adam optimizer with other optimizers and found that the adam optimizer outperform other. We also added dropout layer with dropout rate=0.4 to reduce overfitting.We have also added the max pooling layers as they help the model to learn translation-invariant features We used the ReLU activation function as it turned out to be most effective in achieving fast convergence rates and it also help to mitigate the problem of vanishing gradients that can arise in very deep networks.We used this ReLU activation function as it outperforms the other activation functions like Sigmoid,Tanh and LeakyReLU[18].Additionally, we used the batch normalisation technique as it normalizes the inputs to each layer of a neural network over the current mini-batch during training. This helps to alleviate the problem of internal covariate shift, where the distribution of inputs to a layer changes as the parameters of the previous layers are updated. We used the padding in our code in order to preserve the spatial dimensions of the input data, such as height and width, throughout the network. During the convolution operation, a filter is slid over the input data and performs element-wise multiplication with the input values within the receptive field. The resulting output is a smaller feature map than the input data. Without padding, the feature map would be even smaller than the input, resulting in a loss of spatial information.

Padding involves adding extra pixels or values around the border of the input data to ensure that the output feature map has the same spatial dimensions as the input. This allows the CNN to better preserve the spatial information of the input and maintain the size of the output feature map. AlexNet uses large 11x11 convolutional filters in the first layer, which helps to capture larger and more complex features in the images.

2. **ReLU activation functions**: AlexNet uses ReLU (Rectified Linear Unit) activation functions, which are computationally efficient and have been shown to improve the accuracy of deep neural networks.

3. **Dropout regularization**: AlexNet uses dropout regularization, which randomly drops out some of the units in the network during training to prevent overfitting.

4. **GPU acceleration**: AlexNet was one of the first deep learning models to be trained on GPUs (Graphics Processing Units), which significantly sped up the training time.

It is a groundbreaking neural network architecture that played a significant role in the advancement of deep learning and computer vision.

6.4 Model Training:

The selected CNN models will be trained on the preprocessed dataset. The training process will involve feeding the images into the models, adjusting the weights, and evaluating the model's accuracy using cross-validation techniques.
The training process typically involves the following steps:

1. **Initialization:** The weights and biases of the neural network are initialized randomly.

2. **Forward propagation:** input data is fed into the neural network, and the output is calculated using the current weights and biases.

3. **Calculation of loss:** The difference between the predicted output and the actual output is calculated using a loss function such as mean squared error.

4. **Backpropagation:** error is propagated backwards through the neural network, and the gradients of the loss function with respect to the weights and biases are calculated.

5. **Updating the weights:** The weights and biases of the neural network are updated using an optimization algorithm such as stochastic gradient descent or Adam.

6. **Repeat:** Steps 3-6 are repeated for a fixed number of epochs or until the model converges to a satisfactory level of accuracy.

6.5 Model Evaluation:

The performance of the trained models will be evaluated on a separate test set.

1. **Learning curve**
   is just a plot showing the progress over the experience of a specific metric related to learning during the training of a model. They are just a mathematical representation of the learning process.

2. **Training loss**

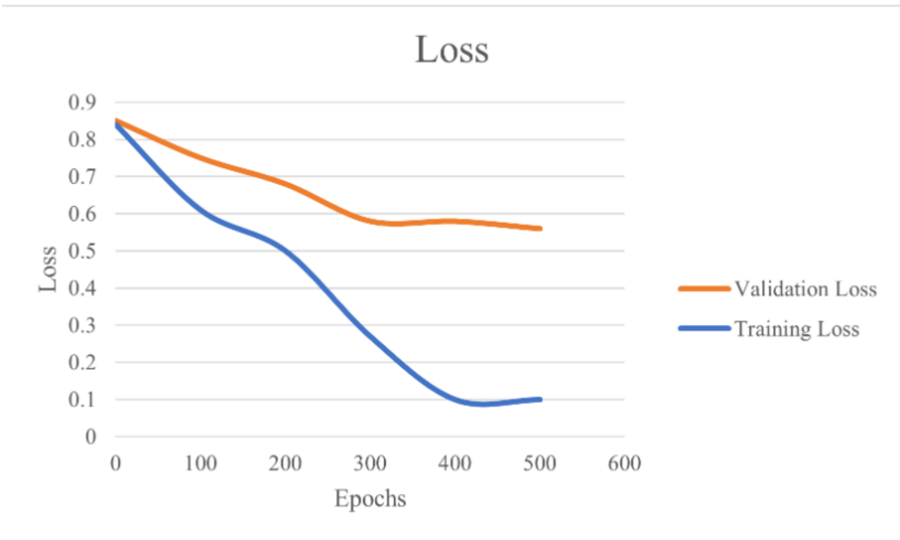
- is a metric used to assess how a deep learning model fits the training data.
- That is to say, it assesses the error of the model on the training set. Note that, the training set is a portion of a dataset used to initially train the model. Computationally, the training loss is calculated by taking the sum of errors for each example in the training set.
- It is also important to note that the training loss is measured after each batch. This is usually visualized by plotting a curve of the training loss.

3. **Validation Loss**

- is a metric used to assess the performance of a deep learning model on the validation set.
- The validation set is a portion of the dataset set aside to validate the performance of the model. The validation loss is similar to the training loss and is calculated from a sum of the errors for each example in the validation set.
- Additionally, the validation loss is measured after each epoch. This informs us as to whether the model needs further tuning or adjustments or not. To do this, we usually plot a learning curve for the validation loss

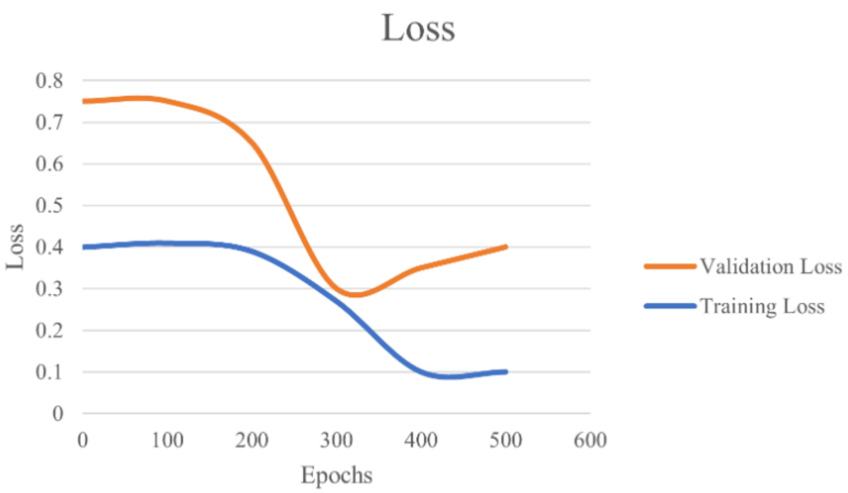
Implication of training and Validation Loss
The training and validation loss is usually visualized together on a graph. The purpose of this is to diagnose the model's performance. Now we will explain Model behaviour:

- **Under Fit Model**
  When the algorithm is not able to model either training data or new data, consistently

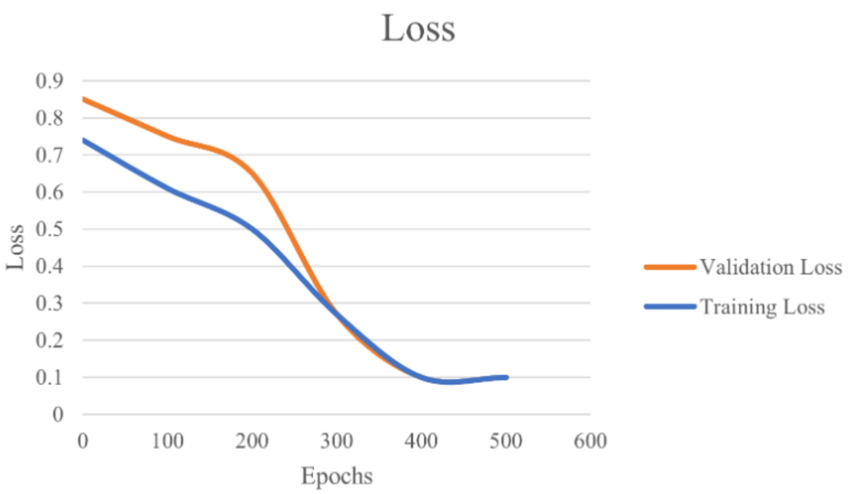


  obtaining high error values that don't decrease over time

- **Over Fit Model**
  The algorithm captures well the training data, but it performs poorly on new data, so it's not able to generalize



- **Good Fit Model**
  The algorithm captures well the training data, and also it performs good on new data, so it's able to generalize



6.6 Comparison and Interpretation:

- The performance of two different CNN architectures, namely VGG16 and AlexNet, will be evaluated and compared based on four different metrics: training loss, training accuracy, validation loss, and validation accuracy. According to the information presented in **Table 4** and **Table 5** on page **23**
- In addition to this, we will analyse the accuracy and loss caused by the various models by using visualisation approaches. As Illustrated in **Figure 1** , **Figure 2** , **Figure 3** and **Figure 4**
- Finally we will analyse the findings that were acquired from the tests and make advice on the most effective CNN architecture for the categorization of potato diseases. In addition to this, we will offer some suggestions on the project's potential future.

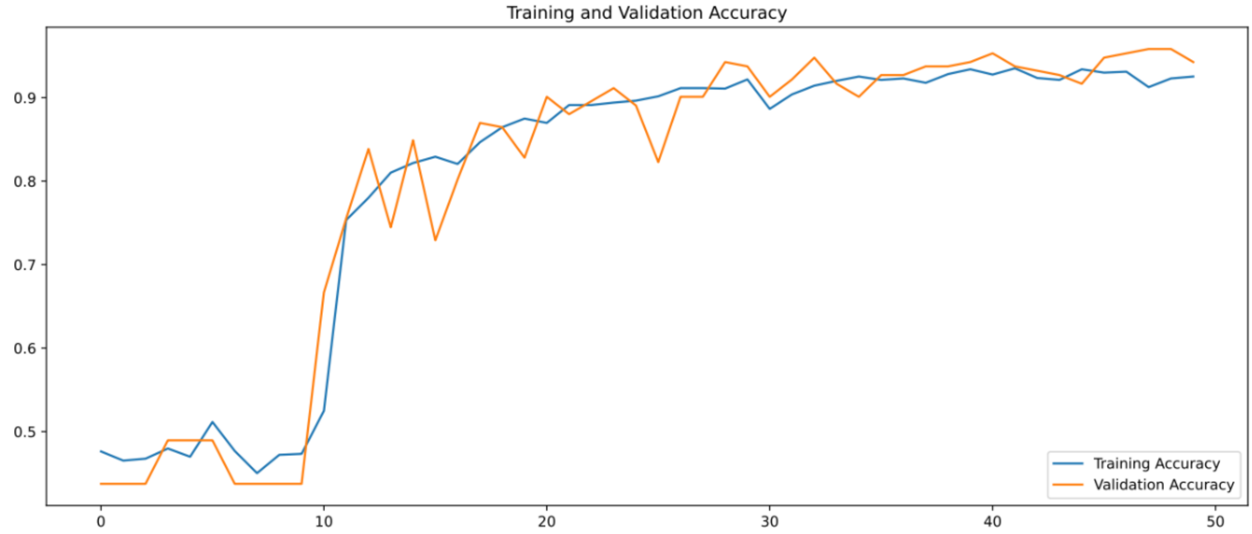


Figure 1: VGG16 Training And Validation Accuracy

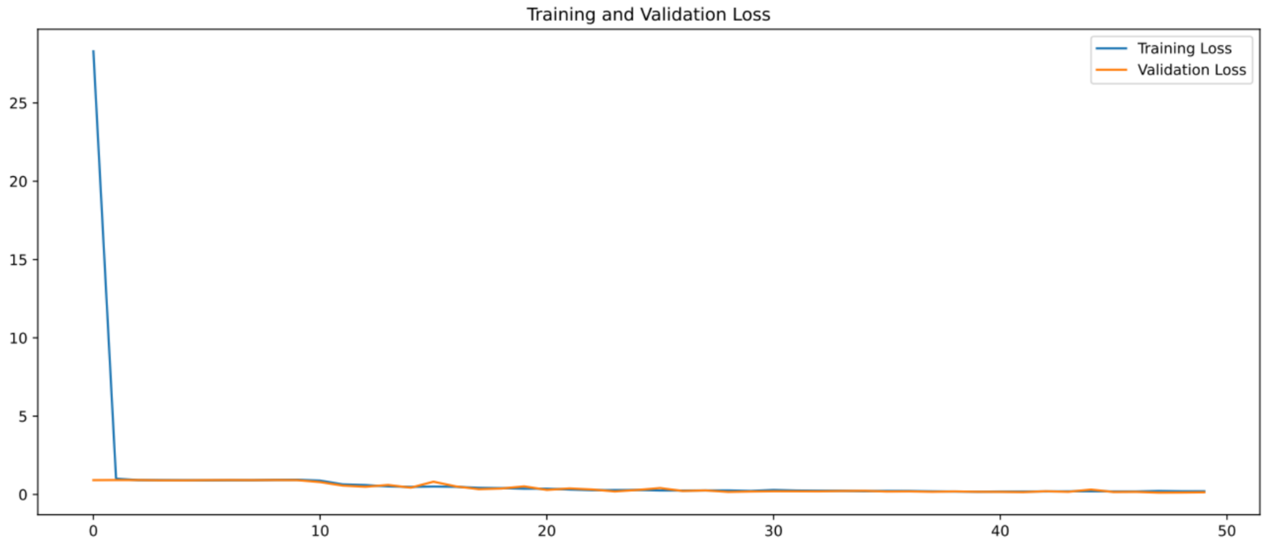


Figure 2: VGG16 Training And Validation Loss

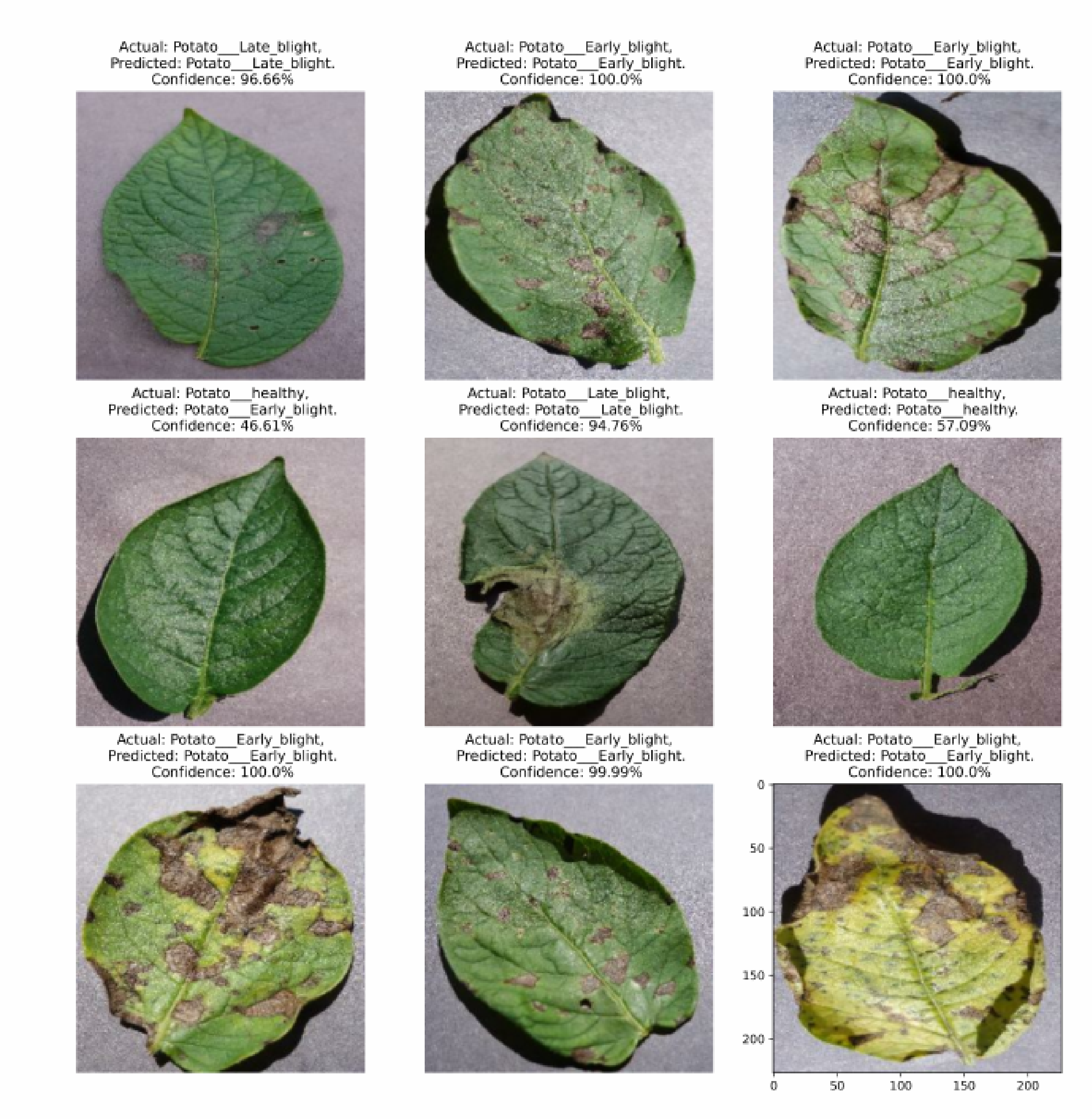


Figure 3: VGG16 Actual vs predicted with model Confidence

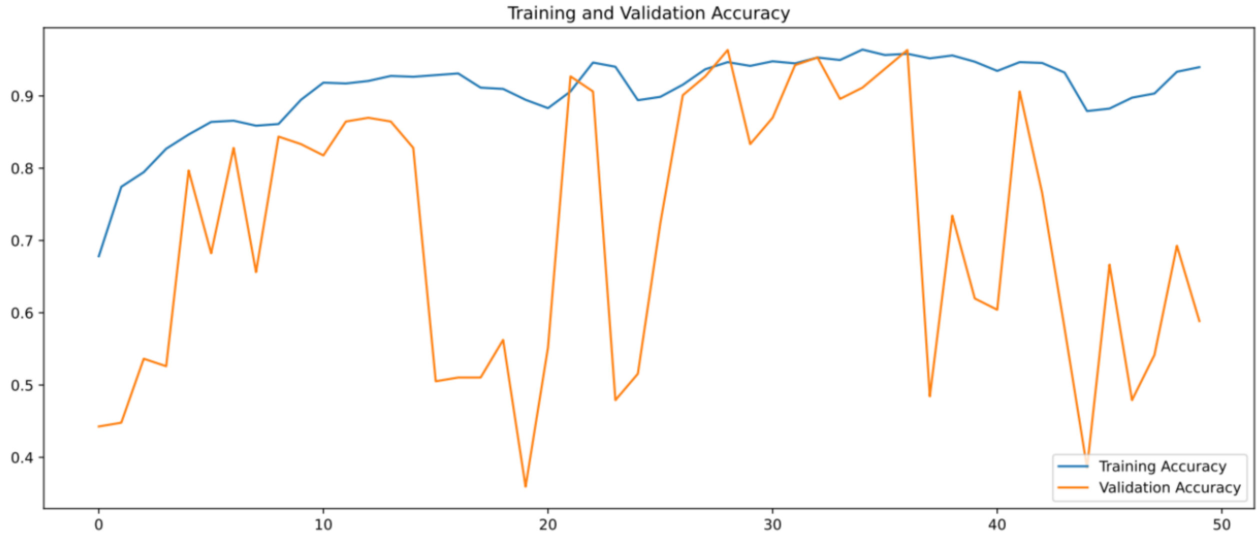


Figure 4: AlexNet Training And Validation Accuracy

Figure 5: AlexNet Training And Validation Loss

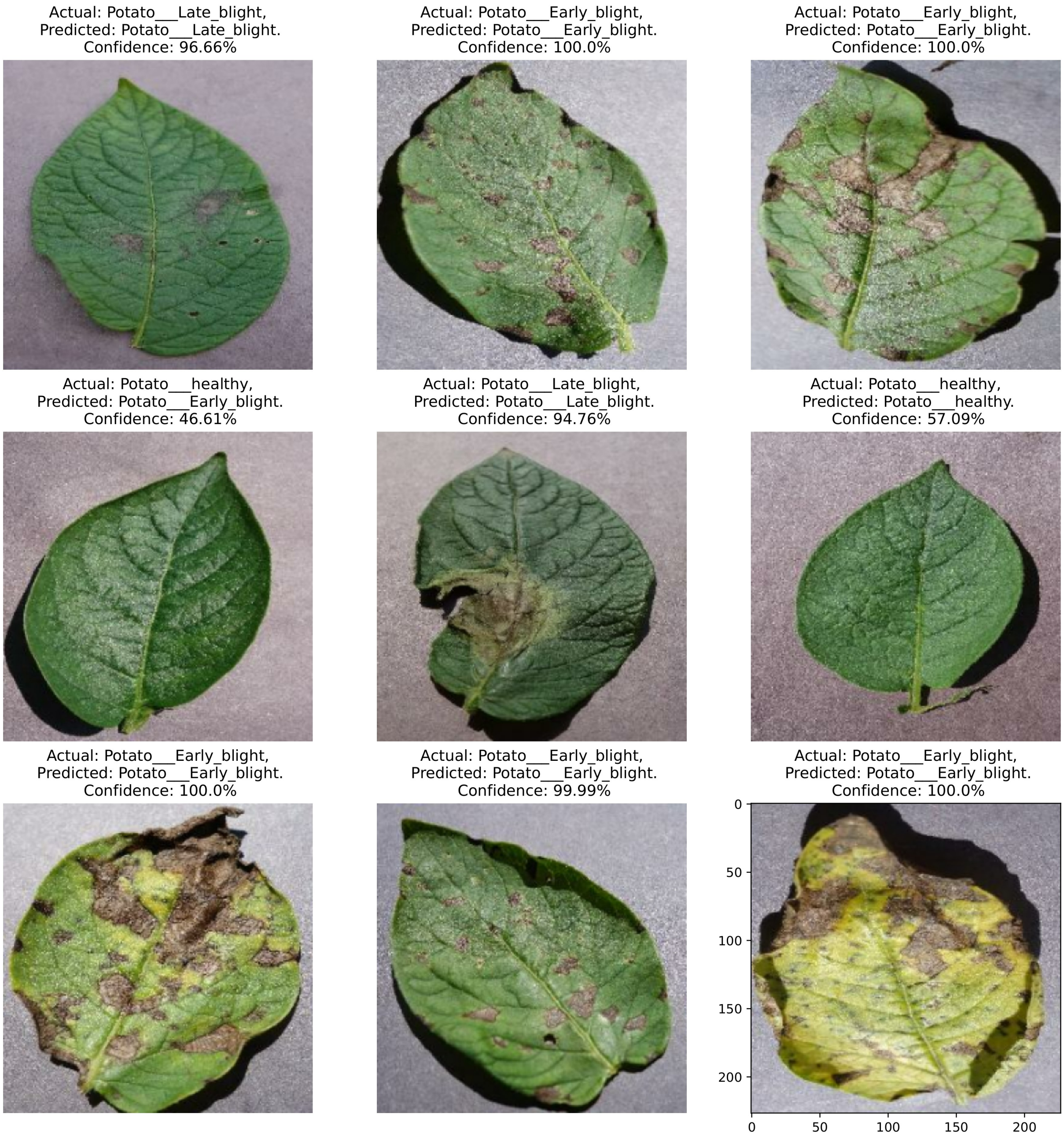


Figure 6: AlexNet Actual vs predicted with model Confidence

7 Results

The graphical depiction of all of the models' training loss compared to their validation loss as well as training accuracy compared to their validation accuracy. The model is better if it has a lower loss, and the classification results will be more satisfying if they have a greater accuracy..

Table 4 on page **23** shows the VGG16 model performance , such that at

1. At Epoch 0, we observe that the training loss is 28 percent and the Training Accuracy is 47 percent. Additionally, the validation loss is 91 percent and the validation accuracy is 43 percent, which is a very poor performance.

2. At the 25th percentile, the training loss is 20 percent and the training accuracy is 78 percent. Additionally, the validation loss is 18 percent and the validation accuracy is 74percent.

Epochs	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
0	28.29690170288086	0.47627314925193787	0.9129108786582947	0.4375
1	1.0081690549850464	0.4652777910232544	0.9190461039543152	0.4375
2	0.9124863743782043	0.46759259700775146	0.9169018864631653	0.4375
3	0.9091529846191406	0.47974535822868347	0.9019158482551575	0.4895833432674408
4	0.9037856459617615	0.46990740299224854	0.9006897807121277	0.4895833432674408
5	0.8967551589012146	0.5115740895271301	0.9126636981964111	0.4895833432674408
6	0.9037861824035645	0.47685185074806213	0.9174254536628723	0.4375
7	0.9045566916465759	0.45023149251937866	0.9128561019897461	0.4375
8	0.9268764853477478	0.4722222089767456	0.9183749556541443	0.4375
9	0.9355403184890747	0.47337964177131653	0.9125134944915771	0.4375
10	0.8855790495872498	0.5248842835426331	0.7891016602516174	0.6666666865348816
11	0.6439003348350525	0.7534722089767456	0.5639086961746216	0.7552083134651184
12	0.5985061526298523	0.7800925970077515	0.4915883541107178	0.8385416865348816
13	0.5145037174224854	0.8101851940155029	0.6044723987579346	0.7447916865348816
14	0.48263785243034363	0.8217592835426331	0.43388959765434265	0.8489583134651184
15	0.5060265064239502	0.8292824029922485	0.8128013014793396	0.7291666865348816
16	0.48714756965637207	0.8206018805503845	0.5105307698249817	0.8020833134651184
17	0.4221855401992798	0.8466435074806213	0.33777403831481934	0.8697916865348816
18	0.4020780622959137	0.8645833134651184	0.3770652115345001	0.8645833134651184
19	0.3646494448184967	0.875	0.5153966546058655	0.828125
20	0.3679255247116089	0.8697916865348816	0.29094019532203674	0.9010416865348816
21	0.3142886459827423	0.8912037014961243	0.385263055562973	0.8802083134651184
22	0.2698726952075958	0.8912037014961243	0.3168947696685791	0.8958333134651184
23	0.2808093726634979	0.8940972089767456	0.19076256453990936	0.9114583134651184
24	0.2843077480792999	0.8964120149612427	0.28765496611595154	0.890625
25	0.25179755687713623	0.9016203880310059	0.4085259437561035	0.8229166865348816
26	0.2494017332792282	0.9114583134651184	0.22050052881240845	0.9010416865348816
27	0.25613775849342346	0.9114583134651184	0.2557823359966278	0.9010416865348816
28	0.2567390203475952	0.9108796119689941	0.16232304275035858	0.9427083134651184
29	0.2234756201505661	0.921875	0.18726301193237305	0.9375
30	0.2873503565788269	0.8865740895271301	0.20188544690608978	0.9010416865348816
31	0.2498994618654251	0.9039351940155029	0.19943749904632568	0.921875
32	0.23743408918380737	0.9143518805503845	0.19601558148860931	0.9479166865348816
33	0.23028886318206787	0.9201388955116272	0.219138503074646	0.9166666865348816
34	0.20381605625152588	0.9253472089767456	0.22983978688716888	0.9010416865348816
35	0.2249804139137268	0.9212962985038757	0.18562321364879608	0.9270833134651184
36	0.21950708329677582	0.9230324029922485	0.19376647472381592	0.9270833134651184
37	0.20157290995121002	0.9178240895271301	0.16958938539028168	0.9375
38	0.1886078268289566	0.9282407164573669	0.1913088709115982	0.9375
39	0.17786607146263123	0.9340277910232544	0.17178362607955933	0.9427083134651184
40	0.18912765383720398	0.9276620149612427	0.1654125154018402	0.953125
41	0.1913708597421646	0.9351851940155029	0.14930829405784607	0.9375
42	0.19241927564144135	0.9236111044883728	0.2037370204925537	0.9322916865348816
43	0.20066353678703308	0.9212962985038757	0.1676207333803177	0.9270833134651184
44	0.1974041908979416	0.9340277910232544	0.3108105957508087	0.9166666865348816
45	0.19442996382713318	0.9299768805503845	0.15552519261837006	0.9479166865348816
46	0.19540713727474213	0.9311342835426331	0.16441206634044647	0.953125
47	0.22263452410697937	0.9126157164573669	0.11585044115781784	0.9583333134651184
48	0.20566123723983765	0.9230324029922485	0.12379684299230576	0.9583333134651184
49	0.20612405240535736	0.9253472089767456	0.1408984512090683	0.9427083134651184

Table 4: VGG16 Model Performance

Epochs	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
0	10.393170356750488	0.6782407164573669	129.3251495361328	0.4427083432674408
1	2.0464603900909424	0.7743055820465088	73.75043487548828	0.4479166567325592
2	1.1764262914657593	0.7945601940155029	13.945462226867676	0.5364583134651184
3	0.9415217638015747	0.8269675970077515	17.35091209411621	0.5260416865348816
4	0.7352074980735779	0.8466435074806213	16.56023406982422	0.796875
5	0.7467201352119446	0.8640046119689941	47.058780670166016	0.6822916865348816
6	0.6136613488197327	0.8657407164573669	14.16031265258789	0.828125
7	0.614230215549469	0.8587962985038757	15.19273853302002	0.65625
8	0.5480902194976807	0.8611111044883728	2.041987419128418	0.84375
9	0.3956541419029236	0.8946759104728699	0.9086384773254395	0.8333333134651184
10	0.36528465151786804	0.9184027910232544	0.8482351899147034	0.8177083134651184
11	0.2782757878303528	0.9172453880310059	0.45061957836151123	0.8645833134651184
12	0.28858980536460876	0.9207175970077515	1.4181569814682007	0.8697916865348816
13	0.22499980032444	0.9276620149612427	1.094497561454773	0.8645833134651184
14	0.24483869969844818	0.9265046119689941	1.4002585411071777	0.828125
15	0.3274518549442291	0.9288194179534912	15.820130348205566	0.5052083134651184
16	0.28838181495666504	0.9311342835426331	6.936670303344727	0.5104166865348816
17	0.7124971747398376	0.9114583134651184	7.6528801918029785	0.5104166865348816
18	0.4671914875507355	0.9097222089767456	137.00111389160156	0.5625
19	1.0715608596801758	0.8946759104728699	101.44417572021484	0.359375
20	0.8990366458892822	0.8831018805503845	4.298376083374023	0.5520833134651184
21	0.41203877329826355	0.90625	1.4562982320785522	0.9270833134651184
22	0.26409903168678284	0.9461805820465088	0.7320014834403992	0.90625
23	0.3310565948486328	0.9403935074806213	24.181676864624023	0.4791666567325592
24	0.7101272344589233	0.8940972089767456	34.73810958862305	0.515625
25	0.6562273502349854	0.8987268805503845	17.455398559570312	0.7239583134651184
26	0.4270302951335907	0.9155092835426331	0.705173909664154	0.9010416865348816
27	0.257717490196228	0.9369212985038757	0.21660639345645905	0.9270833134651184
28	0.210920050740242	0.9467592835426331	0.11058460921049118	0.9635416865348816
29	0.21214096248149872	0.9415509104728699	0.49504217505455017	0.8333333134651184
30	0.23224115371704102	0.9479166865348816	0.3936985731124878	0.8697916865348816
31	0.20287306606769562	0.9450231194496155	0.1562638133764267	0.9427083134651184
32	0.15811707079410553	0.953125	0.3942188024520874	0.953125
33	0.16243599355220795	0.9496527910232544	0.7536579966545105	0.8958333134651184
34	0.13687029480934143	0.9641203880310059	0.21515494585037231	0.9114583134651184
35	0.16447430849075317	0.9565972089767456	0.2082344889640808	0.9375
36	0.14361292123794556	0.9583333134651184	0.09793960303068161	0.9635416865348816
37	0.1959405541419983	0.9519675970077515	21.239736557006836	0.484375
38	0.19656629860401154	0.9560185074806213	1.0213228464126587	0.734375
39	0.2240128368139267	0.9473379850387573	2.474654197692871	0.6197916865348816
40	0.3028446137905121	0.9346064925193787	1.5532315969467163	0.6041666865348816
41	0.1942327320575714	0.9467592835426331	0.26140862703323364	0.90625
42	0.20901024341583252	0.9456018805503845	2.0542850494384766	0.765625
43	0.4675675630569458	0.9322916865348816	1.8567906618118286	0.578125
44	0.8415972590446472	0.8790509104728699	6.179410457611084	0.3854166567325592
45	0.9891258478164673	0.8825231194496155	5.771205425262451	0.6666666865348816
46	0.8151453733444214	0.8975694179534912	20.900373458862305	0.4791666567325592
47	0.48572447896003723	0.9033564925193787	3.299758195877075	0.5416666865348816
48	0.2618371546268463	0.9334490895271301	0.9764537215232849	0.6927083134651184
49	0.27461546659469604	0.9398148059844971	5.780017852783203	0.5885416865348816

Table 5: AlexNet Model Performance

3. Now, the training loss is 27 percent, and the accuracy of the training is 89 percent. Additionally, the validation loss is 28 percent, and the accuracy of the validation is 90 percent.

4. The training loss was twenty percent at the most recent epoch, and the training accuracy was ninety-two percent. The validation loss was fourteen percent, and the validation accuracy was ninety-four percent.

Now similary for **Table 5** on page **23** shows the AlexNet model performance , such that at

1. At item Epoch 0, we see that the training loss is ten percent while the training accuracy is sixty-seven percent. In addition, the validation loss is one hundred twenty-nine percent while the validation accuracy is forty-four percent, which is a very poor performance.

2. At the 25th percentile, the training loss is 22 percent and the training accuracy is 89 percent. Additionally, the validation loss is 73 percent and the validation accuracy is 53 percent.

3. Now, the training loss is 32 percent, and the training accuracy is 92 percent. Additionally, the validation loss is 200 percent, and the validation accuracy is 72 percent.

4. The training loss is now at 27 percent, and the training accuracy is currently at 93 percent. Additionally, the validation loss is currently at 570 percent, and the validation accuracy currently sits at 58 percent.

Thus final accuracy is 58 percent which is very less then VGG16 model which has 94 percent.

8 Conclusion

Thus this project provides the detailed analysis of the 2 models in CNN architectures considering the different parameters taken into consideration while predicting the output of the image. As discussed above in result section we get VGG16 outperform AlexNet in terms of accuracy. So, VGG16 is better for potato disease classification than AlexNet. also we see from **Figure 1** and **Figure 2** we see VGG16 model behaviour is Goodfit model that is The algorithm captures well the training data, and also it performs good on new data, so it's able to generalize and from **Figure 3** and **Figure 4** we see AlexNet is Overfit model behaviour that is The algorithm captures well the training data, but it performs poorly on new data, so it's not able to generalize. Computer vision techniques offer several advantages over traditional techniques, including speed, accuracy, consistency, adaptability, scalability, and cost-effectiveness, making them ideal for a wide range of applications in various industries.

- Speed and Efficiency: can analyze images and videos much faster and more efficiently than traditional techniques, thanks to advancements in hardware and algorithms. This enables real-time processing of large volumes of data, which is crucial for applications such as autonomous vehicles, surveillance, and robotics.

- Accuracy and Consistency:are more accurate and consistent than traditional techniques. They can detect and classify objects and patterns in images and videos with high precision, even in complex and noisy environments. This makes them ideal for applications that require high levels of accuracy, such as medical diagnosis, quality control, and security.

- Adaptability and Scalability:are highly adaptable and scalable. They can be trained to recognize new objects and patterns by simply adding more data to the training set. This makes them ideal for applications that require frequent updates and improvements, such as social media platforms and e-commerce websites.

- Cost-effectiveness: can be more cost-effective than traditional techniques. They can automate repetitive and labor-intensive tasks, reducing the need for human labor and lowering costs. This makes them ideal for applications that require large-scale image and video analysis, such as agriculture, manufacturing, and logistics.

9 Future Work

For the future scope of this project We can increase the dataset and get better accuracy of the models,also can increase the computation resources so we can use multiple CNN architecture for comparision to get a broader understanding of the comparision of CNN and could also use data augmentation by adding more samples and data replication, transfer learning along with better CNN models and hyperparameter tweaking can be used to im- prove results

References

[1] F. al Heeti, M. Ilyas. 2022 2nd International Conference on Computing and Machine Intelligence, ICMI 2022. Comparative analysis of convolutional neural network architectures for classification of plant leaf diseases. 2022 2nd International Conference on Computing and Machine Intelligence (ICMI), Istanbul, Turkey, 2022, pp. 1-5, doi: 10.1109/ICMI55296.2022.9873752.

[2] B. Chellapandi and S. Chopra, "Comparison of Pre-Trained Models Using Transfer Learning for Detecting Plant Disease," pp. 383–387, 2021. Computers and Electronics in Agriculture, vol. 145, pp. 311- 318, 2018. 3. K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Comput. Electron. Agric., vol. 145, no. September 2017, pp. 311–318, 2018, doi: 10.1016/j.compag.2018.01.009.

[3] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Comput. Electron. Agric., vol. 145, no. September 2017, pp. 311–318, 2018, doi: 10.1016/j.compag.2018.01.009.

[4] X. Li, L. Rai, and I. Engineering, "Apple Leaf Disease Identification and Classification using ResNet Models," pp. 738–742, 2020.

[5] S. Arya and R. Singh, "A Comparative Study of CNN and AlexNet for Detection of Disease in Potato and Mango leaf," 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), Ghaziabad, India, 2019, pp. 1-6, doi: 10.1109/ICICT46931.2019.8977648.

[6] Arpita Patel, Mrs. Barkha Joshi, "A Survey on the Plant Leaf Disease Detection Techniques", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 6, Issue 1, ISO 3297:2007

[7] Ahmed Ali Mohammed Al-Saffar, Hai Tao, Mohammed Ahmed Talab, "Review of Deep Convolution Neural Network in Image Classification", International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications, 2017.

[8] Yang Lu, Shujuan Yi, Nianyin Zeng, Yurong Li, Yong Zhang, "Identification of rice diseases using deep convolutional neural networks" , Neurocomputing 267 (2017) 378–384.

[9] Waseem Rawat, ZenghuiWang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review", unpublished.

[10] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton "ImageNet Classification with Deep Convolutional Neural Networks", International Conference on Neural Information Processing Systems - Volume 1, June 2016

[11] Neena Aloysius and Geetha M , "A Review on Deep Convolutional Neural Networks", International Conference on Communication and Signal Processing, April 6-8, 2017.

[12] S.Arivazhagan, S.Vineth Ligi, "Mango Leaf Diseases Identification Using Convolutional Neural Network", International Journal of Pure and Applied Mathematics" vol.120, No. 6,11067-11079, ISSN: 1314-3395, August 2018

[13] K. P. Ferentinos, “Deep learning models for plant disease detection and diagnosis”, Computers and Electronics in Agriculture, vol. 145, pp. 311- 318, 2018.

[14] S. Zhang, W. Huang and C. Zhang, “Tree-channel convolutional neural networks for vegetable leaf disease recognition”, Cognitive Systems Research, 2018.

[15] Arif, I. M. Dheir, A. Soliman, A. Mettleq, and S. S. Abunaser, “Potato Classification Using Deep Learning,” Int. J. Acad. Pedagog. Res., vol. 3, no. 12, pp. 1–8, 2019.

[16] H. Yanagisawa, T. Yamashita, and H. Watanabe, “A Study on Object Detection Method from Manga Images using CNN,” ieee-2018 Int. Work. Adv. Image Technol., pp. 7–10, doi: 10.1109/IWAIT.2018.8369633.

[17] Cagliarini, Adam, and Anthony Rush. ”Economic development and agriculture in India.” RBA Bulletin, June (2011): 15-22.

[18] Fukushima, Kunihiko. ”Neocognitron: A hierarchical neural network capable of visual pattern recognition.” Neural networks 1.2 (1988): 119-130.

[19] Karen Simonyan, Andrew Zisserman, Proceedings of the International Conference on Learning Representations (ICLR) in 2015.Very Deep Convolutional Networks for Large-Scale Image Recognition.

[20] Matthew D. Zeiler and Rob Fergus,Proceedings of the European Conference on Computer Vision (ECCV) in 2014 Visualizing and Understanding Convolutional Networks.

[21] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) in 2018 ,Exploring the Limits of Weakly Supervised Pretraining.

[22] Diederik P. Kingma and Jimmy Lei Ba,International Conference on Learning Representations (ICLR) in 2015,Adam: A Method for Stochastic Optimization.

[23] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar,International Conference on Learning Representations (ICLR) in 2018.On the Convergence of Adam and Beyond.

[24] Nitish Shirish Keskar and Richard Socher,Proceedings of the International Conference on Learning Representations (ICLR) in 2017.An Empirical Comparison of Optimizers for Machine Learning Models.

[25] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov,Journal of Machine Learning Research in 2014.Dropout: A Simple Way to Prevent Neural Networks from Overfitting.

[26] Matthew D. Zeiler and Rob Fergus,Proceedings of the European Conference on Computer Vision (ECCV) in 2014.Visualizing and Understanding Convolutional Networks.

[27] Vinod Nair and Geoffrey E. Hinton,Proceedings of the International Conference on Machine Learning (ICML) in 2010.Rectified Linear Units Improve Restricted Boltzmann Machines.

[28] Pramod K. Singh and Yongjian Hu,Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN),An Empirical Comparison of Activation Functions in Deep Neural Networks.

[29] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller.International Conference on Learning Representations (ICLR) in 2015.Striving for Simplicity: The All Convolutional Net.

[30] Sergey Ioffe and Christian Szegedy,International Conference on Machine Learning (ICML) in 2015,Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.