AppleCaps: A Capsule Model for Classification of Foliar Diseases in Apple Leaves

MSc Research Project
Data Analytics

Rakhi Ashok Sonkusare
Student ID: x20122314

School of Computing
National College of Ireland

Supervisor: Aaloka Anant

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Rakhi Ashok Sonkusare
Student ID:	x20122314
Programme:	Data Analytics
Year:	2021
Module:	MSc Research Project
Supervisor:	Aaloka Anant
Submission Due Date:	16/12/2021
Project Title:	AppleCaps: A Capsule Model for Classification of Foliar Diseases in Apple Leaves
Word Count:	7324
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	16th December 2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

AppleCaps: A Capsule Model for Classification of Foliar Diseases in Apple Leaves

Rakhi Ashok Sonkusare
x20122314

Abstract

Foliar diseases in apple leaves are characterized by distinct spots on the leaves that appear in different shapes and colours. Frog Eye Leaf spot, Scab, Rust and Powdery Mildew are the most common Foliar diseases affecting the productivity of apple orchards. This research proposes a novel deep learning network AppleCaps for accurate classification of multi-class disease classes and overcoming the limitation of spatial invariance in CNNs. Two baseline CNN models EfficientNet-B3 and ResNet152 are implemented, and their performances are compared and evaluated with the AppleCaps model when exposed to the augmented dataset. Experimental results showed that Capsule Networks provided better performance with an accuracy of 87.06% in classifying the disease type in apple leaves. Hyperparameter tuning with Random Search optimization is used to optimize the model performance, and the results are compared based on accuracy, precision, recall, F1-score and validation loss. Data augmentation techniques helped improve the performance of AppleCaps and CNN models. This research will help farmers identify apple leaf diseases at an early stage of diagnosis and prevent losses to agricultural fields.

Keywords — Foliar Leaf Disease Detection, Deep Learning, Capsule Network, EfficientNet-B3, ResNet152, Data Augmentation, HyperParameter Tuning, Image Processing

1 Introduction

Apples are amongst the most consumed temperate fruit crop grown at a large scale in many parts of the world. They are a rich source of essential Vitamins and nutrients that make them a healthy food option. Apple leaves have polyphenol antioxidants that can fight against chronic diseases. (El-Hawary et al.; 2021) investigated a domesticated apple tree known as Malus Domestica Borkh that is an apple tree with medicinal properties. Empirical results showed that the metabolite phlorizin extracts obtained from this apple tree proved to be neuroprotective agents that can be used for the treatment of Alzheimer disease and other recognition impairment disabilities. Due to its high antioxidant potential, a large number of apple fruits, as well as healthy leaves, are harvested during the summer season to enhance the fruit production quality. Through the harvesting activities, various by-products of these apple leaves are being produced and used in medicinal and cosmetic applications (Ben-Othman et al.; 2021).

1.1 Background and Motivation

The increasing number of apple leaf diseases are making the apples rot. The consumption of these rotten apples can affect human health and lead to a loss of productivity in the apple fields. This, in turn, impacts the economic value of the country and causes damage to agricultural sectors. Disease detection in plants is still an ongoing challenging task for farmers. The farmers can't identify these diseases by looking through the human eye and hence, assistance from the industry experts is needed. Manual detection of apple diseases is a time-consuming task for the experts. By the time, the disease could be classified, the apple tree is already affected by the disease. There is a need for an automated model that can identify these diseases at an early stage of infection through diagnosis.

The concern of the farmers related to the extensive damage of apple orchards and loss of productivity in apple fruits has inspired many researchers to develop and learn different machine learning and deep learning techniques to solve the problem of disease detection. This research is motivated towards diagnosing and classifying different Foliar diseases in apple leaves using a novel deep learning model. Taking into account the major challenges of capturing leaf images at different stages of the disease and different times of the week using digital camera, we focus on comparing the efficiency of the two baseline CNN models with the novel Capsule Network architecture. The visual apple leaf images in Figure 1 are taken from the 8th Plant Pathology FGCV (Fine Grained Visual Categorization) Kaggle Competition. [1]

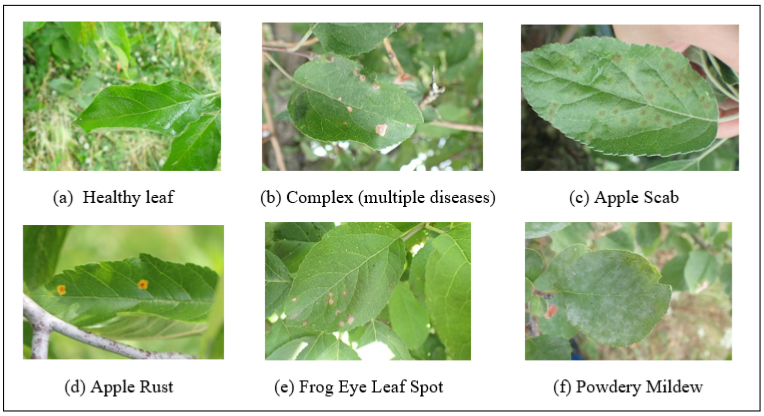

Figure 1: Apple Leaf Images infected by Foliar Diseases

1.2 Research Question and Objectives

To summarize the purpose of this research, I intend to answer the following research questions:

1. How effectively can a Capsule Network model as a feature extractor and classifier improve the identification of Foliar leaf diseases in Apple trees?
2. To what extent does the Capsule model overcome the limitations of two baseline CNN models (EfficientNet-B3, ResNet152)?

[1] https://www.kaggle.com/c/plant-pathology-2021-fgvc8/data

3. How does the fine-tuning of Hyperparameters impact the performance evaluation of the baseline CNN models?

With the intend to solve the mentioned research questions, the research objectives are set as below:

Table 1: Research Objectives

Obj.	Description	Evaluation Metrics
1	Literature Review and Understanding of Implemented Deep learning methodologies for plant leaf disease detection (2019-2021)	
2	Data Collection, Pre-processing of Leaf Images, Resizing, Data Augmentation with Horizontal Flipping, Rotation, Zooming, Train and Validation Split	
3	Implementation of baseline Efficient-B3 and ResNet152 CNN models on the Apple leaf dataset	Accuracy, Loss, Precision, Recall, F1-Score
4	Implementation of baseline Efficient-B3 and ResNet152 CNN models with hyper parameter tuning using Random Search method	Accuracy, Loss, Precision, Recall, F1-Score
5	Implementation of AppleCaps Capsule model using Gaussian Blur on Apple leaf dataset	Accuracy, Loss, Precision, Recall, F1-Score
6	Performance Evaluation and Comparison of results for each model implementation	

The research is organised into different sections as follows: Section 2 discusses the literature review on the previous researches carried out on the classification of plant disease detection and advancement of Capsule Networks in other applications. In Section 3, the methodology of the research is discussed including the steps taken to perform the proposed methodologies. Section 4 includes the design specifications and the architecture taken to attain the outcomes. Section 5 discusses the final steps for models implementation. Section 6 indicates the key findings of the research and comparison of models based on evaluation metrics to answer the research questions. Section 7 summarizes the research and defines the scope for future work.

2 Related Work

The literature review for this research can be broadly classified into three subsections. We will discuss how the image classification tasks have evolved from using machine learning techniques to switching onto deep learning methods for better image classification by addressing the pros and cons of the applicable techniques in the first two sections. In the third section, we will discuss the use of the proposed methodology Capsule Networks in image recognition tasks across multiple fields of research.

2.1 Overview of Machine Learning Techniques in Image Classification

The agricultural field has suffered a great loss in the past few years due to the increasing number of plant diseases in different agricultural crop species. Over 42% of the loss has been incurred due to these diseases. With the advent of new technologies, it was possible to diagnose the occurrence of these diseases at initial stage. The use of machine learning algorithms helped detect multiple disease categories in various plants. The experiment by (Tulshan and Raul; 2019) involved disease detection in apple leaves using K Nearest Neighbours (KNN) classification model. The dataset consisted of 75 images with seven different disease classes: Down Mildew, Early Blight, Mosaic Virus, Leaf Miner, White Fly. The leaf images were captured in their RGB format through a digital camera and then converted to grayscale using MATLAB picture creating a library. Image Segmentation methods were employed on the input images to select the region of interest from the leaves. Also, feature extraction techniques like Gray Level Co-occurrence Matrix (GLCM) were used to extract 13 discriminative properties to enhance the classification of a particular disease type. Experimental results indicated that the KNN classifier showed the exact disease name along with the amount of area affected due to the disease with an accuracy of 98.56%. Other important observations indicated the sensitivity of the model and the elapsed time of the disease.

India's economy is majorly dependent on agricultural production as it contributes to 17% of the country's GDP and provides employment to around 60% of the population. The increasing disease types in fruit plants are causing huge losses to the farmers. Manual detection of diseases is a tedious task for the experts and time consuming, hence automatic detection of disease in the real world is the need of the hour. A multi-class Support Vector Machine (SVM) was implemented in research (Agarwal et al.; 2019) to identify the extent of the disease and control the loss of yield. Gray Level Co-occurrence Matrix (GLCM) algorithm is a feature extraction technique that extracts 13 distinct features from the segmented leaf images obtained by K-means clustering. Following the pre-processing stages, the disease classification task was performed by the SVM model that achieved an accuracy of 98.387% in the identification of disease types in fruit trees.

A study by (Singh et al.; 2021) implemented a novel image segmentation technique to extract the diseased part of the apple leaves. For the images captured under complex backgrounds, segmentation plays an important role to separate the region of interest from the background noise. The leaf images were enhanced using the Brightness-Preserving Dynamic Fuzzy Histogram Equalization technique, and the diseased leaf part was extracted for the further classification task. Two different disease classes like Marsonina Coronaria or apple scab were detected from the extracted features, including colour and texture, classified by the KNN classifier. The novel segmentation method improved the performance of the KNN classifier with an accuracy of 96.4% when compared to other segmentation techniques. Future work suggests applying this improved model on a large dataset with more disease classes and leaf images captured at different times of the day.

2.2 Overview of Deep Learning Techniques in Image Classification

Machine learning techniques have been used widely in the detection of plant diseases, although after the advancement of a subset in the ML methods, Deep learning methods evolved with great potential and showed promising results in better accuracy. The research by (Saleem et al.; 2019a) created an improved version of the state-of-the-art methods using visualization techniques. For a better understanding of plant diseases, visualization activation filters were applied in the initial layers of AlexNet and GoogleNet models which enhanced the diseased spot on the tomato leaf images. With the help of the saliency map created by the visualizations, GoogleNet was able to identify the disease type and outperformed AlexNet. The segmentation and edge visual maps in the LeNet model provided good results in detecting diseases in olive plants. Other CNN models like ResNet-50, ResNet-101, VGG were implemented using Faster-RCNN and SSD (Single Shot Multibox Detector) detectors that created a bounding box to detect spots on banana leaves. Heat maps were introduced as an input to the diseased images and feature maps were evaluated to detect rice diseases. A method based on the hotspot technique was used to extract the spots from the diseased leaves, which described two features i.e.color information and texture of the spot. Gaussian noise and Jittering techniques were also performed on the entire dataset. A comparative study was carried out using different visualization, data augmentation and feature extraction methods on the PlantVillage dataset containing different plant species and the individual performances were evaluated based on accuracy and ROC curve. The study focuses on using hyperspectral imaging as an emerging technology in the detection of plant diseases before the symptoms are visible.

The research by (Jiang et al.; 2019) implemented an improved CNN model that involved a combination of GoogleNet Inception framework and Rainbow Concatenation to detect five different apple leaf diseases like Brown spot, Mosaic, Grey spot, Rust, Alternaria in a real-time environment. The initial dataset consisted of a total of 2,029 diseased images that had the following characteristics: multiple diseases infused into a single leaf and complex backgrounds of the captured images. The leaf images were converted to a total of 26,377 leaf images using data augmentation techniques to balance the disease classes and avoid overfitting. The model was developed for the faster detection of apple diseases. Each of the diseased classes has a discriminative feature that enables the model to distinguish multiple disease types on the apple leaves. The experiment resulted in a 78.80% mean average precision with a detection speed of 23.13 FPS. The proposed novel methodology was capable of extracting the distinct features from the diseased leaf images in real-time to enhance multi-class disease detection tasks.

The research by (Kumar et al.; 2020) implemented a Residual Network with 34-layers (ResNet-34) on the New Plant Disease dataset containing 15200 images with multiple crop types and 38 disease classes. ResNet architecture contains a residual block that skips the degradation problem occurring while increasing the number of layers in the network. Also, it is a pre-trained model on the ImageNet dataset that tends to provide higher accuracy on a small training dataset. The performance of the ResNet model was evaluated on two parameters: Average Weighted Precision (AVP) of 96.51% and accuracy of 99.40% was achieved. Future work aims at capturing images in panorama view and aerial images. The dataset can be extended to more disease classes by adding more images.

Traditional CNN models consist of extensive pooling layers to increase the accuracy of the model and reduce the number of parameters, although it leads to the loss of spatial features. A novel Deep Neural Network (DNN) architecture with EfficientNet and DenseNet was implemented in a study by (Srinidhi et al.; 2021) to detect apple leaf diseases. Both the DNN models can preserve spatial information and can overcome the shortcomings of the CNN models. EfficientNet provides better computational performance even in a smaller training dataset. The experiment was performed on a public dataset of the Fine-Grained Visual Categorization (FGCV) workshop with 3600 real-time images. Image Augmentation techniques like Blurring, Flipping, Canny Edge detection were employed on the original dataset to increase the number of images split into training and validation datasets. The model was evaluated on its performance accuracy with over 40 epochs and resulted in an accuracy of 99.8% by EfficientNet-B7 and 99.75% by DenseNet. These models with less number of parameters outperformed other CNN models like AlexNet, GoogleNet, VGG, consisting high number of parameters and more pooling layers.

2.3 Capsule Network : A Novel Deep Learning Technique for Image Classification

The world faced a novel life-threatening pandemic called the Coronavirus (COVID-19) disease at the end of the second decade of the 21st century. Due to the increasing spread of this virus, early diagnosis of COVID-19 was required to break the chain spread and reduce the growing cases. CNNs were capable of facilitating the detection of positive Covid-19 cases, but are prone to loss of spatial information and requires more training data. The research by (Afshar et al.; 2020) used an alternative deep learning framework Capsule Networks since it can handle smaller datasets consisting of X-ray images. The novel COVID-CAPS model showcased advantages over traditional methods with 95.7% accuracy, Area under Curve of 0.97, Sensitivity 90% with less trainable parameters. Pre-trained modelling and transfer learning were employed to enhance the diagnosis abilities of the COVID-CAPS architecture. Model pre-training enhanced the accuracy by 98.3%. With the increasing number of cases all around the world, the research aims at building a more robust model to facilitate larger training data and improve the diagnosis of the Covid-19 virus.

Disease detection of plant leaves has been successful in the past years due to the invention of Convolutional Neural networks. Although, there are a few drawbacks of CNN models like the inability of the max-pooling layer which cannot capture the pose, orientation and view of leaf images. Other drawbacks involve huge training data required for classification and failure to evaluate the spatial relationship of features. The research by (Oladejo and Ademola; 2020) implemented an optimized Capsule network for the detection of banana leaf diseases. The leaf dataset consists of 1000 images collected from a field with two classes of diseased leaves: Banana Bacterial Wilt and Banana Black Sigatoka along with healthy leaves. The images were used in the RGB colour format without grayscale and resized to the appropriate image size. Additional feature extraction and data augmentation techniques are used to balance the data amongst the two classes. The capsule network obtained an accuracy of 95.36% and was compared to the CNN model, LeNet5 and ResNet50 implemented from scratch to compare the performance. Leaky ReLu activation function was used to fine-tune the hyperparameters of the model. Future research aims at implementing a more robust model for disease classification and increasing the number of disease classes in the dataset.

The paper (Kwabena et al.; 2020) introduced Gabor filters in the Capsule network to enhance the disease detection in tomato and citrus plants. These Gabor filters are used for feature extraction, edge detection or texture analysis. The use of data augmentation techniques like rotation, deformation and Gaussian blur was employed on the training data to create an improved dataset and avoid the overfitting issue. Two baseline CNN models GoogleNet and AlexNet were implemented to compare the performance with the proposed framework. Hyperparameter tuning to change the batch size, learning rate, the dropout rate was used but it did not affect the performance of the Capsule model. Although, the number of routing iterations set to three gave significantly the best performance with an accuracy of 98.12%. Experimental results showed that CapsNet outperformed the baseline CNN models in terms of robustness, accuracy, complexity, etc. The proposed model can detect unhealthy and diseased leaves from the healthy leaves in challenging lighting conditions and images captured in diverse angles. Future work aims at reducing the number of trainable parameters for implementation on mobile devices for convenient use.

2.4 Summary of the key findings

After reviewing the literature and related work on the implementation of various machine learning and deep learning models, we have observed that deep learning methods have outperformed the traditional state-of-the-art machine learning models. Capsule Networks have emerged as a promising deep learning technique that is capable of preserving spatial information and overcoming the shortcomings of CNN models. They have shown exceptional results in the agricultural field by enhancing the early diagnosis of diseases in plant leaves as well as in medical applications by tackling the ongoing COVID-19 pandemic. Different data augmentation techniques help create an enhanced dataset that can improve the classification accuracy. The use of optimizers and hyperparameter tuning also play a significant role in the performance of Capsule networks.

3 Methodology

3.1 Understanding the application domain

Despite various researches conducted on plant disease detection using machine learning and deep learning techniques, an automated model that can detect the occurrence of a disease at an early stage of diagnosis with accurate prediction is still lacking. CNN models have been implemented in the past few years to detect Foliar diseases in apple leaves in real-time situations. With the advancement in technology and ongoing challenges related to the background of leaf images captured, lighting conditions, pose and orientation of the leaf images, CNNs fail to capture this information (Patrick et al.; 2019). The increasing number of foliar diseases in apple trees are hampering the productivity of the apple orchards. Hence, I will be developing a model that can deal with the challenges addressed and help in the accurate prediction of diseases in apple leaves.

3.2 Dataset Collection

The dataset for this research is taken from two sources: 8th FGCV Plant Pathology Kaggle Competition with labeled apple leaf images and manual capturing of unlabelled apple tree leaf images. The dataset derived from online competition contains 18,632 apple leaf images infected with different foliar disease categories: Apple Scab, Apple Rust, Frog Eye Leaf Spot, Powdery Mildew, Complex class with leaf images having multiple diseases, as well as leaves with a combination of one or more diseases. The labeled dataset consists of 12 different disease classes including healthy leaves. The unlabeled healthy and diseased leaf images are captured manually from the apple trees in a nearby locality at different times of the day and week with the help of mobile camera. A few sample leaf images taken manually are shown in Figure 2.

Figure 2: Manually Captured Unlabeled Apple Leaf images

The labeled leaf image dataset is used for training the baseline CNN models and the novel Capsule Network, whereas the images taken using a mobile camera are used to make predictions as testing images. The labeled data is further divided into training and validation data using a split-ratio of 80:20.

3.3 Image Pre-processing Methods

Certain issues related to complex image backgrounds, lighting conditions, class imbalance or leaf images with one disease superimposed on another disease symptoms can cause the model to make inaccurate predictions and complicate the model-building task. To simplify the model-building steps, pre-processing the input raw and inconsistent data into reliable and clean images is a good practice. In this research, I performed the following image pre-processing steps:

- **Resizing of images**: The input images were resized into a dimension of 512 * 512 for the baseline CNN models and 224 * 224 for the Capsule network model.
- **Noise Reduction**: Gaussian Blur is performed on the input images for Capsule network to remove noise from the images.
- **Image Augmentation**: The process of enhancing and creating new images into the existing dataset using techniques such as Flipping, Rotation, Shifting, Blurring, etc is referred to as Image Augmentation. It helps in improving the model's performance (Bansal et al.; 2021). Most of the images belong to Apple scab, Healthy and Frog eye leaf spot symptoms, whereas there are fewer leaf images in Rust, Powdery mildew, Complex and multiple disease symptoms (Arsenovic et al.; 2019). In this research, the following augmentation techniques are performed:

  1. Horizontal Flipping: The images are flipped or upturned by the number of rows. This is especially relevant in real-time disease detection tasks.
  2. Rotation_range: The images are rotated by an angle of 20 degrees.

3. Zoom_range: The images are zoomed out by a value of 0.2. When the value of the zoom range exceeds by 1, the images get magnified.

Figure 3: Image Augmentation on Apple Leaf Images

4. Shear_range: This applies a shearing transformation on the images by an angle of 20 degrees.
5. width_shift_range and height_shift_range: The images are shifted vertically and horizontally along X-axis and Y-axis by 20%.

3.4 Model Selection and Hyper Parameter Tuning

Convolutional Neural Networks have shown impressive results in various image classification tasks. They act as a backbone for most of the deep learning models and are used commonly in plant disease detection. The max-pooling layer in CNNs helps to extract features from the input image, although they cannot capture the orientation and pose of the images referred to as spatial information (Oladejo and Ademola; 2020). CNNs require huge annotated or labeled data to train the model which is a challenging task. To overcome these limitations, Capsule networks have originated as a challenging model for the image classification task. CapsuleNet can preserve spatial information of the images such as hue, pose, orientation and view. In this research, I have implemented and performed a comparative study amongst the underlying models:

- I implemented a novel architecture of Capsule Networks as a feature extractor and classifier for the classification of foliar diseases in apple leaves referred to as 'AppleCaps'. Adam optimizer is used with a learning rate of 0.001 to improve the accuracy of the model that results in a lesser training time with more efficiency (Melinte and Vladareanu; 2020).

- To answer the research question on the advantages of Capsule Networks over CNNs, I perform a comparative study by implementing two of the baseline CNN networks EfficientNet-B3 and ResNet152 with the use of Adam optimizer.

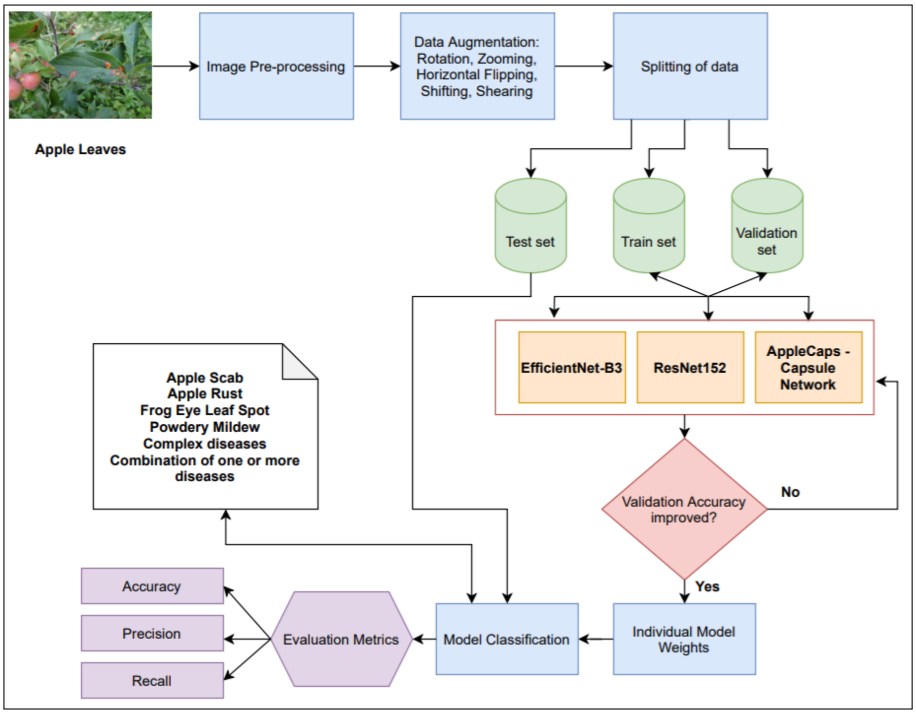

Figure 4: Framework of Proposed Methodology

The performance of the proposed CNN methodologies can be further enhanced by choosing optimum parameters for the best results. The batch size is set to 32 in our models. The parameters like number of epochs, learning rate, number of units in the Dense layer can be fine-tuned using optimization. The fine-tuning of hyperparameters is performed with random test runs to obtain respective model accuracies by training and validating the models (Bansal and Kumar; 2021). In this research, I have used Random Search optimization that finds a random combination of tuning parameters based on the objective defined in the model. Random Search takes lesser time to find the optimal combination of parameters and is proved to be faster than other optimization methods such as Grid Search and Bayesian Optimization (Firdaus et al.; 2021). In our experiments, I implemented the baseline CNN models with and without hyperparameter tuning to evaluate the performance of the models.

3.5 Model Evaluation Metrics

The implemented models are evaluated using various evaluation metrics like Accuracy, Precision, Validation loss, Recall, F1-score. These metrics are compared for each of the methodologies and the results are evident in the Evaluation section 6.

4 Design Specification

4.1 Capsule Network

A capsule consists of several neurons that differentiate information specific to a feature or an object of interest to be classified based on its texture, width, height and orientation and stores it in a high-dimensional 8 or 16 dimension vector. Capsule Networks have a distinct functionality of inverse graphics approach wherein it deconstructs an object into different sub-parts and develops an interconnection between the sub-parts. The architectural design of Capsule Network involves primary capsule layer, secondary capsules and loss calculation.

4.1.1 Primary Capsule Layer

There are 3 processes involved in the Primary Caps Layer: Convolutional, Reshaping and Squashing function as shown in Figure 5. In the convolutional layer, grayscale images are used, hence dimension of the image will be (28,28,1) where 1 represents the number of channels. For RGB images, the value is 3. The input image is convoluted with the filter kernel which is of dimension (9,9,1,256) where 256 refers to the convolutional filters. After the convolution is done, the resultant block is of size (20,20,256) with 256 channels.

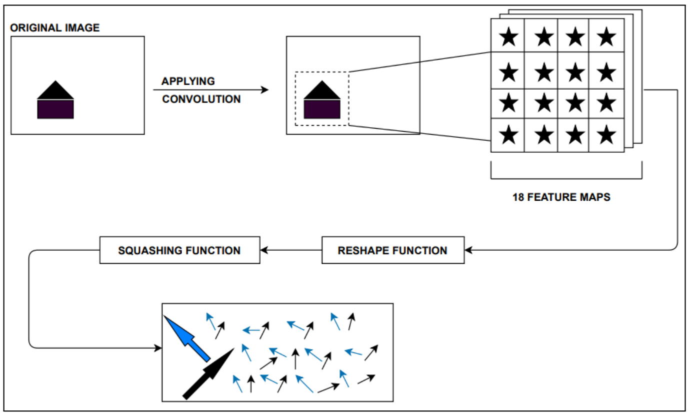

Figure 5: Primary Capsule Layer

The number of parameters that will be trained equals 20,992 which is obtained from the calculation 256 * (9 * 9 + 1). The resultant block is further convoluted with a filter block of dimension (9,9,256,256) to obtain a resultant block of (6,6,256) size. So, now the number of trainable parameters is equal to 256 * (256 * (9 * 9 + 1)) = 5,308,672. The reshaping process will reshape the final block of the (6,6,256) dimension into 32 blocks with the dimension of (6,6,8). These 6 * 6 blocks contain all the vector outputs that define the position, orientation, texture and height of the object in an 8-dimensional vector. The total 8D vectors will be equal to 6 * 6 * 32 = 1152 vectors. The length of each vector refers to the probability of an object if it is present or not in an image and should not exceed the value 1. The squashing function then comes into the picture which squashes all the vectors with a value more than 1 to a probability range between 0 to 1.

4.1.2 AppleCaps Layer

In this layer, each (1,1,8) vectors are converted to (1,1,16) dimension vectors and create 10 vectors amongst 1152 8D vectors. The transformation from 8-dimensional to 16-dimensional vectors will take place in the AppleCaps layer using the coefficients (C,W). These co-efficient parameters need to be trained equivalent to 8 * 16 * 1152 * 10 + 1152 * 10 = 1,486,080 parameters. The number of trainable parameters is calculated using a simple illustration. Suppose the input layer has 5 neurons and the output layer consists of 4 neurons connected in a feed-forward network, then the total trainable parameters will be 5 * 4 = 20. Now, suppose the input has 2 layers each with 5 neurons and output has 3 layers each with 4 neurons. The weights required to interconnect the neurons will be equal to 5 * 4 * 3 * 2. The 5 and 4 in this equation represent 1152 and 10 vectors. Similarly, the 3 and 2 in the calculation indicates 8 and 16-dimensional vectors respectively.

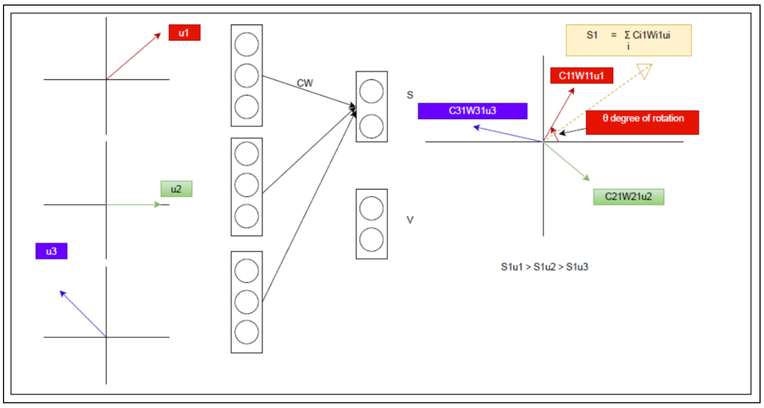

Figure 6: Routing by Agreement in AppleCaps

The C is the routing co-efficient that represents the interconnection between the input capsule and the output capsule. The building block of Capsule Network is the process referred to as routing by agreement. In this process, when a primary capsule agrees to interconnect with a secondary capsule, the signal is agreed upon making it easier for the network to make a prediction. Suppose there are 3 input capsules with output vectors as u_1,u_2 and u_3 as shown in Figure 6. When the u1 vector routes towards the parent capsule, let's say, S, the vector generated will be a product of C * W_{11} * u_1 where W_{11} indicates the first capsule in one layer connected to the first capsule in another layer. The S vector will be calculated as the summation of all the input vectors. The final vector derived from the parent capsule will be a summation of all the vectors. The value of the C co-efficient is generated using the similarity between the resultant vector S and each of the input vectors. The similarity between the two vectors can be indicated by the dot product which will let us know about the vector that contributes highest to the vector generated by S. Based upon this agreed-upon routing, the classification task will be performed.

4.1.3 Margin loss and Reconstruction Loss

Multiclass classification is referred to as a task that involves the classification of two or more classes. In this research, 12 disease classes are classified. Margin loss is used to calculate the probability of each disease class, which means if an object of that class is present in the input, then the square of the length of the corresponding vector of the respective capsule should not be lesser than 0.9, while in the reverse case, it should not be more than 0.1. Along with the margin loss, another component in the network i.e Decoder is connected to the AppleCaps layer with 3 fully connected layers where one is the sigmoid activation function and the other two are ReLu activation units. The decoder helps to reconstruct the input image by minimizing the squared difference between recreated image and the input image known as the reconstruction loss and preserves the features for the image to be recreated and behaves as a regularizer by avoiding overfitting problems while training the model. [2]

4.2 Baseline CNN Models - EfficientNet-B3 and ResNet152

The baseline CNN models EfficientNet-B3 and ResNet152 are implemented in this research to perform a comparative study with the Capsule network. EfficientNet-B3 performs uniform scaling of the number of layers, channels and size of the image, and thus, provides good performance by maintaining the model structure with lesser training parameters. EfficientNet performs the dimension scaling with a defined set of scale coefficients. The architecture of EfficientNet comprises of Mobile Inverted Convolution (MBConv) that follows the narrow to wide and back to narrow approach indicated by widening with 1 * 1 convolution, then 3 * 3 depth convolution to decrease the number of parameters and again 1 * 1 convolution to reduce the number of channels. This dimension scaling is performed with a high parameter accuracy making it a compact architecture (Srinidhi et al.; 2021).

ResNet-152 is a convolutional neural network that consists of residual blocks for training the model with 152 convolutional layers. The number of parameters in this model is around 60.2 million. It learns from its residual functions by skipping a few of the layers and thus, gains higher accuracy and are easy to optimize with depth. To perform downsampling, 7 * 7 convolutional layer is used and the layers contain 3 * 3 filters, followed by a global pooling layer and an FC layer along with the softmax activation function. For a specific feature map size, there will be an equivalent number of filters in the network (Hossain et al.; 2020).

5 Implementation

5.1 Experimental Setup

The implementation of deep learning models highly depends upon the system configurations. The code for the models was built and run using Python3.7. I made use of various frameworks and libraries for the implementation like TensorFlow, Keras that provide functions like ImageDataGenerator for image pre-processing, Dense, Flatten, Dropout functions for adding layers to the network and pre-defined functions like ResNet152 and EfficientNet-B3 for model building. Numpy and Pandas were utilised to deal with a data frame and NumPy arrays. Kaggle notebooks are used that provide a high processing GPU accelerator for up to 40 hours per week to run the code. Although, due to the limited run hours, I used Google Colab IDE which is a cloud-based publicly accessible platform to train our model and perform the execution as it provides a dedicated GPU for running deep learning models. The dataset size is around 16GB and was uploaded to Drive to make it accessible for the code run. Google Colab allows saving the outputs of the trained model in a CSV file that can be used as a reference. Matplotlib library was used to create interactive visualizations. The prime factors that are directly proportional to the computational time of the model are the number of epochs, batch size, steps

[2] `https://www.intel.com/content/www/us/en/developer/articles/technical/understanding-capsule-network-architecture.html`

per epoch and learning rate. This is handled using hyperparameter tuning of models to provide better results.

5.2 Implementation of AppleCaps: Capsule Network

For the implementation of Capsule Network, the dataset is fetched from the FGCV8 Plant Pathology Competition and a seaborn barplot is generated to check for the number of disease classes present in the training data. There are 12 disease categories of Foliar diseases. The input images are resized to a dimension of 224 * 224 and image augmentation methods like Horizontal Flipping, Rotation, Contrasting, Zooming, Shifting and Shearing are applied on 1000 training images to create an enhanced dataset as seen in Figure 3. Gaussian Blurring is used to reduce extra noise from the images. The images in their RGB format are converted to grayscale images to select distinct features from the leaf images. Since there is no pre-defined architecture present in the Tensorflow for Capsule networks, the model is built from scratch that involves building the convolutional layer with 256 filter kernels. The reshaping function will reshape the 32 features into two vectors of 9 dimensions each. The squashing function is created to squash the probabilities of the output vectors to a value between 0 to 1 if the vectors exceed more than 1. The routing by agreement block is created that will output the vector of the input capsule with the highest probability relevance corresponding to the target disease class to the decoder and the other vectors will be masked out. The decoder will calculate the margin loss and reconstruction loss for each round of the dynamic routing. The reconstruction layer will reconstruct the leaf image from the features extracted from the input image. The input dataset is split into training and validation set with a split ratio of 80:20. The model is trained with a batch size of 32 and the number of epochs as 8 and the accuracy and validation loss is generated and displayed at each epoch. Capsule Network requires high computational GPU for model training. The model testing was performed on the manually captured images from the apple trees in a nearby locality and the corresponding plots for Training Vs Validation accuracy and Training Vs Validation loss are displayed using ggplot library.

5.3 Implementation of Efficient-B3 and ResNet152 models

The architecture for EfficientNet and ResNet152 are available in the Tensorflow library. Keras is a high-level interface that provides functions to build deep learning models. Using the ImageDataGenerator, I performed various image pre-processing methods like resizing the input images to a dimension of 512 * 512 and data augmentation techniques like Horizontal Flipping, Zooming, Rotation, Shifting and Shearing are mapped to 32 training images. I have loaded both the models with the Sequential class of Keras and assigned weights as ImageNet. The model is run with 466 steps per epoch for a batch size of 32 and number of epochs as 8. The learning rate of the model is 0.001 obtained from Adam Optimizer. Early Stopping is a method in Keras that enables to specify a parameter, in our experiment, val_loss and mode value is set as min to minimize the validation loss and maximize accuracy. The model training will stop when the chosen metric stops improving. The results are derived from the history of the model run and visualizations are created using the ggplot library. The model prediction is performed on the manually captured apple leaf images and the predicted disease class index and category are displayed. Hyperparameter tuning using the Random search optimization technique is incorporated to find the optimal combination of the learning rate, number of epochs and batch size to evaluate the model's performance as compared to without parameter tuning of the model.

6 Evaluation

The key findings and performance evaluation of the proposed methodologies are explained in this section that answers the listed research questions. To understand and analyze the outcomes of the model implementation, several evaluation metrics are critically evaluated. For this research, the model evaluation is performed based on the metrics like Accuracy, F1-score, validation loss, validation accuracy, precision and recall (Saleem et al.; 2019b).

6.1 Evaluation 1 : Baseline EfficientNet-B3 and ResNet152 CNN models without Hyper Parameter Tuning

The baseline EfficientNet-B3 and ResNet 152 models are implemented on the training data of 8th FGCV Plant Pathology with a batch size of 32 and number of epochs as 8. The model is compiled using Adam optimizer with a learning rate of 0.001 and the accuracy is calculated. The loss calculation is done through Categorical Cross Entropy. Early stopping and Reduce Learning Rate on Plateau callbacks are used to stop the model training if the validation loss and the learning rate stops improving. EfficientNet-B3 model gave an accuracy of 74.26% with a validation accuracy of 75.88% whereas the ResNet152 model obtained an accuracy of 76.50% and a validation accuracy of 78.23%.

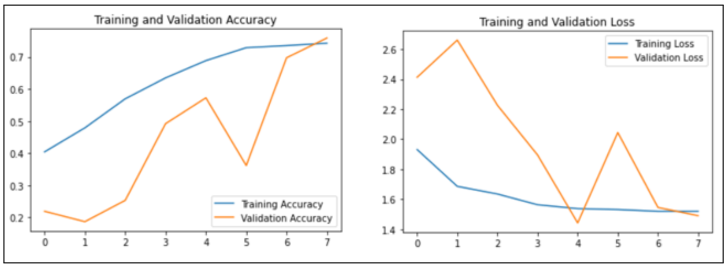

Figure 7: Plots for Accuracy and Loss using EfficientNet-B3 model

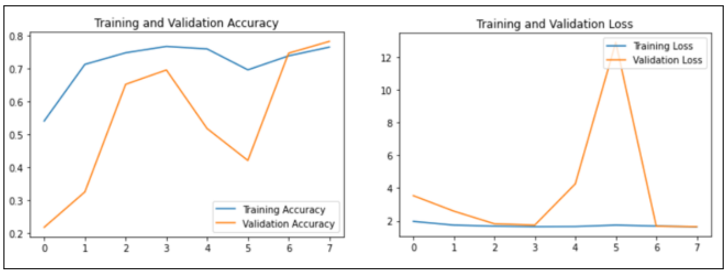

Figure 8: Plots for Accuracy and Loss using ResNet152 model

From Figure 7, it is observed from the Accuracy plot of EfficientNet model that with the increase in epoch, the training and validation accuracy also increased. Similarly, in the

Loss plot, it is observed that the loss has reduced at the 6th epoch and improved further in the last epoch for the validation data.

From Figure 8, it is observed from the Accuracy plot of ResNet152 model that the training and validation accuracy increases as the number of epochs increases. Similarly, the validation loss has decreased gradually in the Loss plot after the 5th epoch and remained constant till the last epoch.

6.2 Evaluation 2 : Baseline EfficientNet-B3 and ResNet152 CNN models with Hyper Parameter Tuning

The baseline CNN models are implemented using the Random Search fine-tuning method. The objective in the tuner is set as val_loss to minimize the validation loss and increase accuracy. Random Search tuner predicted the optimal parameter values for ResNet152 with learning rate as 0.0001 and number of units in the Dense Layer as 512 in 12 trial runs, whereas for EfficientNet-B3, the predicted learning rate is 0.001 and number of units in the Dense Layer as 768. The models are built using the tuned parameters and executed. EfficientNet-B3 provided an accuracy of 85.84% with a validation accuracy of 87.17% whereas ResNet152 provided an accuracy of 84.32% with a validation accuracy of 85.99%.

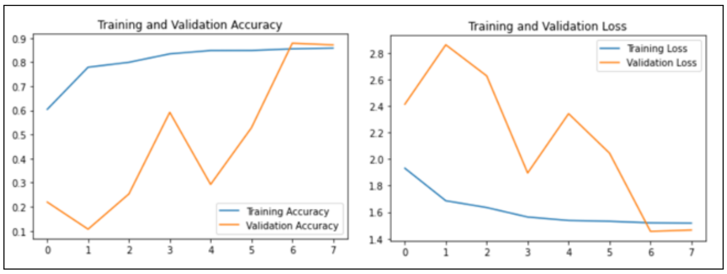

Figure 9: Plots for Accuracy and Loss using EfficientNet-B3 model

From Figure 9, it is observed from the Accuracy plot of EfficientNet model that with the increase in epoch, the training and validation accuracy also increased. Similarly, in the Loss plot, it is observed that the loss has reduced at the 6th epoch and remained constant till the last epoch for the validation data.

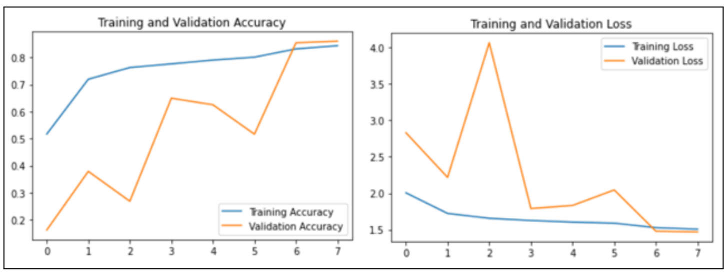

Figure 10: Plots for Accuracy and Loss using ResNet152 model

From Figure 10, it is observed from the Accuracy plot of ResNet152 model that the training and validation accuracy increases as the number of epochs increases. Similarly,

Table 2: EfficientNet-B3 Classification report

Disease Class	Precision	Recall	F1-Score	Support
complex	0.07	0.05	0.06	307
frog_eye_leaf_spot	**0.15**	**0.15**	**0.15**	**554**
frog_eye_leaf_spot complex	0.00	0.00	0.00	18
healthy	**0.20**	**0.23**	**0.21**	**814**
powdery_mildew	0.12	0.14	0.13	337
powdery_mildew complex	0.00	0.00	0.00	20
rust	0.10	0.11	0.10	399
rust complex	0.00	0.00	0.00	12
rust frog_eye_leaf_spot	0.00	0.00	0.00	24
scab	**0.30**	**0.31**	**0.31**	**1108**
scab frog_eye_leaf_spot	0.00	0.00	0.00	105
scab frog_eye_leaf_spot complex	0.00	0.00	0.00	28

Table 3: ResNet152 Classification report

Disease Class	Precision	Recall	F1-Score	Support
complex	0.07	0.08	0.07	307
frog_eye_leaf_spot	**0.15**	**0.16**	**0.15**	**554**
frog_eye_leaf_spot complex	0.00	0.00	0.00	18
healthy	**0.23**	**0.23**	**0.23**	**814**
powdery_mildew	0.10	0.11	0.10	337
powdery_mildew complex	0.00	0.00	0.00	20
rust	0.10	0.11	0.10	399
rust complex	0.00	0.00	0.00	12
rust frog_eye_leaf_spot	0.00	0.00	0.00	24
scab	**0.30**	**0.29**	**0.30**	**1108**
scab frog_eye_leaf_spot	0.04	0.02	0.03	105
scab frog_eye_leaf_spot complex	0.00	0.00	0.00	28

the validation loss has decreased gradually in the Loss plot after the 5th epoch and reduced further until the last epoch. Along with the model accuracy, I have calculated precision, recall, F1-Score and support for each of the Foliar disease classes in a classification report as shown in Table 2 for EfficientNet-B3 and Table 3 for ResNet152. Accuracy is an important measure, but it may lead to incorrect conclusions when the class distribution amongst the disease categories is not balanced. Thus, precision, F1-score and recall play a vital role in the performance evaluation of a model. Due to the class imbalance problem, both the CNN models are biased towards scab, healthy and frog_eye_leaf_spot disease classes.

6.3 Evaluation 3 : AppleCaps - Capsule Network Model

The CapsNet model is implemented on the training data of the 8th FGCV dataset with a batch size of 32 and number of epochs equal to 8. The model is compiled using the Adam optimizer to minimize the validation loss. There is no pre-defined function for Capsule in the Keras architecture, hence the model is built from scratch. With each epoch run, the corresponding loss and accuracy are displayed. The AppleCaps model obtained an accuracy of 87.06% with a validation accuracy of 88.85%.

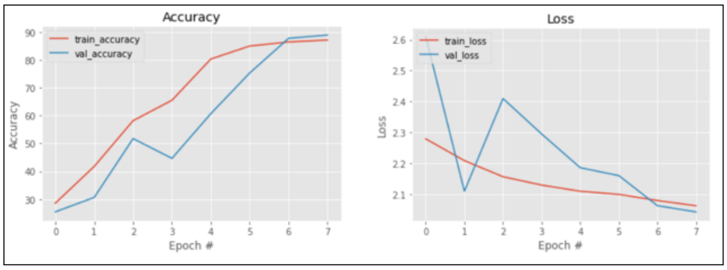

Figure 11: Plots for Accuracy and Loss using AppleCaps model

From Figure 11, we can see in the Accuracy plot that as the number of epochs increases, the accuracy also increases whereas in the loss plot, the loss initially reduced at epoch 1 and later increased but gradually started improving and was minimum at the 8th epoch.

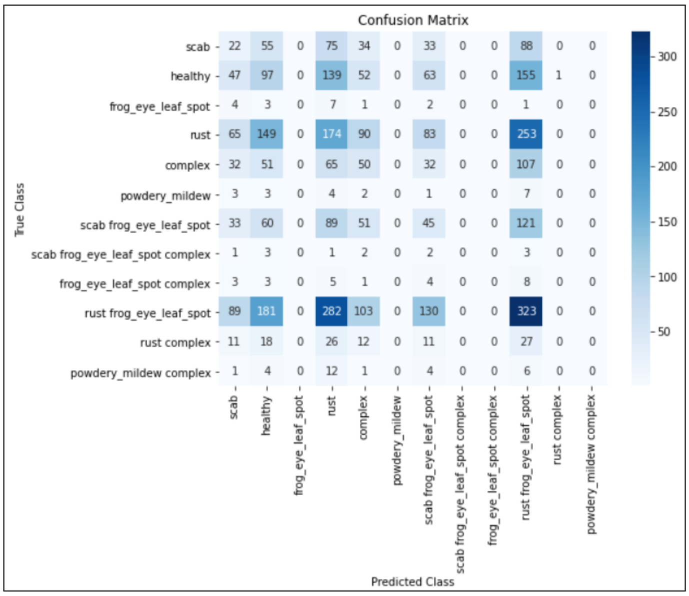

Figure 12: Confusion Matrix of AppleCaps model

The confusion matrix in Figure 12 shows that 323 images belonging to rust frog_eye_leaf_spot diseases are more accurately classified denoted by dark blue colour. 282 images belonging to rust frog_eye_leaf_spot are predicted as rust and inversely 253 images belonging to rust are classified as rust frog_eye_leaf_spot category.

6.4 Discussion

This research proposed and implemented a Novel Capsule Network Architecture AppleCaps to detect different categories of Foliar Leaf diseases in Apple trees. After a critical review of past researches and literature, many experiments have shown that Capsule Networks have a better performance in terms of preserving spatial information as compared to CNN models. The implementation of Capsule Networks in this research shows that it provides better performance in terms of Accuracy, Precision, Recall, F1-score and Validation loss when exposed to the augmented dataset. Accuracy of 87.06% was achieved for the Capsule model whereas the baseline CNN models EfficientNet-B3 and ResNet152 achieved an accuracy of 74.26% and 76.50% respectively. It is evident from the model evaluation 6.3 that CapsuleNet outperforms the baseline CNN models and provide better classification results as proposed in 1.2. Table 4 shows the comparison of results between the proposed methodologies.

Table 4: Comparison of Accuracy for Baseline CNN and AppleCaps Model

Model	**Accuracy**
EfficientNet-B3 without Hyper Parameter Tuning	74.26%
EfficientNet-B3 with Hyper Parameter Tuning	**85.84%**
ResNet152 without Hyper Parameter Tuning	76.50%
ResNet152 with Hyper Parameter Tuning	**84.32%**
AppleCaps (Caspule Network)	**87.06%**

In our experiment in 6.2, I tuned the learning rate and number of units in the Dense layer using Random Search optimization in both the baseline CNN models. The accuracy of CNN models obtained using parameter tuning can be seen in Table 4. From the results, we can see that baseline CNN models with parameter tuning perform better as compared to model implementation without tuning as proposed in 1.2. For Capsule network, I have not implemented fine-tuning of parameters since the standard approach of implementation is used, rather the model is trained again and again and the loss is calculated and minimized with each epoch. The above models were tested for their performance by passing an unexposed and unlabeled testing set consisting of manually captured diseased and healthy apple leaf images from a nearby locality. The Capsule model was able to classify the disease categories of the apple leaves more accurately. The models were saved and then retrieved with the help of checkpoints to make predictions on an unexposed dataset.

```
from IPython.display import Image
Image(filename='D:\\EfficentNet_Apple\\plant-pathology-
```

```
print("Predicted class:", predictions[0])
```

```
Predicted class: rust frog_eye_leaf_spot
```

Figure 13: Prediction result by AppleCaps model

In Figure 13, the AppleCaps model classifies the leaf as 'rust frog_eye_leaf_spot' which is a multiple disease type. In terms of computational time, Capsule Network took lesser time to run as compared to baseline CNN models EfficientNet-B3 and ResNet152 which takes more time with 466 steps per epoch.

7 Conclusion and Future Work

The research aims at solving a Multiclass classification problem for the detection of Foliar diseases in apple tree leaves. Taking major challenges under consideration related to image capturing backgrounds, pose and orientation of leaf images and lighting conditions, the 8th FGCV Plant Pathology dataset was enhanced using data augmentation techniques and exposed to Capsule Network and two baseline CNN models EfficientNet-B3 and ResNet152 and the corresponding results were evaluated. Experimental results indicated that AppleCaps outperformed the CNN models with an accuracy of 87.06% and in terms of Precision, Recall and F1-score, since CNNs have a limitation of preserving spatial information also known as spatial invariance. Although, fine-tuning of CNN models provided better performance for EfficientNet-B3 and ResNet152 with an accuracy of 85.84% and 84.32% respectively as compared to the models without hyperparameter tuning. Capsule network also outperformed the baseline CNN models implemented with hyperparameter tuning. Hence, this implementation will be useful for farmers and agricultural experts to detect the diseases in apple trees at an early diagnosis stage and avoid losses to the productivity of apple orchards. The entire implementation can be integrated into any service and used for the detection of plant diseases for other crop species.

Future work involves capturing more diverse images from different apple orchards and enhancing the model for the detection of pest symptoms in apple leaves. Data augmentation techniques can be implemented in the model to balance the disease class distribution.

Acknowledgement

I would like to express my gratitude towards my supervisor Aaloka Anant for his unwavering support and knowledge that helped me complete my research. His expertise and guidance were valuable to carry out the research experiments and tackle any issues related to code or report writing. He has been very supportive about the approach followed throughout the research in our weekly calls that helped me complete my project step by step. I would also like to thank National College of Ireland and the staff of MSc in Data Analytics for acknowledging me to continue my research. Finally, I would like to thank my family and friends who encouraged me throughout my masters in NCI.

References

Afshar, P., Heidarian, S., Naderkhani, F., Oikonomou, A., Plataniotis, K. N. and Mohammadi, A. (2020). Covid-caps: A capsule network-based framework for identification of covid-19 cases from x-ray images, *Pattern Recognition Letters* **138**: 638–643.

Agarwal, A., Sarkar, A. and Dubey, A. K. (2019). Computer vision-based fruit disease detection and classification, *Smart Innovations in Communication and Computational Sciences*, Springer, pp. 105–115.

Arsenovic, M., Karanovic, M., Sladojevic, S., Anderla, A. and Stefanovic, D. (2019). Solving

current limitations of deep learning based approaches for plant disease detection, *Symmetry* **11**(7).
URL: *https://www.mdpi.com/2073-8994/11/7/939*

Bansal, P., Kumar, R. and Kumar, S. (2021). Disease detection in apple leaves using deep convolutional neural network, *Agriculture* **11**(7): 617.

Bansal, S. and Kumar, A. (2021). Automatic deep neural network hyper-parameter optimization for maize disease detection, *IOP Conference Series: Materials Science and Engineering*, Vol. 1022, IOP Publishing, p. 012089.

Ben-Othman, S., Kaldmäe, H., Rätsep, R., Bleive, U., Aluvee, A. and Rinken, T. (2021). Optimization of ultrasound-assisted extraction of phloretin and other phenolic compounds from apple tree leaves (malus domestica borkh.) and comparison of different cultivars from estonia, *Antioxidants* **10**(2): 189.

El-Hawary, S. S., Hammam, W. E., El-Tantawi, M. E.-M., Yassin, N. A., Kirollos, F. N., Abdelhameed, M. F., Abdelfattah, M. A., Wink, M. and Sobeh, M. (2021). Apple leaves and their major secondary metabolite phlorizin exhibit distinct neuroprotective activities: Evidence from in vivo and in silico studies, *Arabian Journal of Chemistry* **14**(6): 103188.

Firdaus, F. F., Nugroho, H. A. and Soesanti, I. (2021). Deep neural network with hyperparameter tuning for detection of heart disease, *2021 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, pp. 59–65.

Hossain, M. S., Noor, F. I., Ali, M. A. et al. (2020). *Plant disease detection using convolutional neural network*, PhD thesis, Brac University.

Jiang, P., Chen, Y., Liu, B., He, D. and Liang, C. (2019). Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks, *IEEE Access* **7**: 59069–59080.

Kumar, V., Arora, H., Harsh and Sisodia, J. (2020). Resnet-based approach for detection and classification of plant leaf diseases, *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp. 495–502.

Kwabena, P. M., Weyori, B. A. and Mighty, A. A. (2020). Gabor capsule network for plant disease detection, *(IJACSA) International Journal of Advanced Computer Science and Applications* **11**(10).

Melinte, D. O. and Vladareanu, L. (2020). Facial expressions recognition for human–robot interaction using deep convolutional neural networks with rectified adam optimizer, *Sensors* **20**(8): 2393.

Oladejo, B. F. and Ademola, O. O. (2020). Automated classification of banana leaf diseases using an optimized capsule network model, *CS & IT Conference Proceedings*, Vol. 10, CS & IT Conference Proceedings.

Patrick, M. K., Adekoya, A. F., Mighty, A. A. and Edward, B. Y. (2019). Capsule networks–a survey, *Journal of King Saud University-computer and information sciences* .

Saleem, M. H., Potgieter, J. and Arif, K. M. (2019a). Plant disease detection and classification by deep learning, *Plants* **8**(11).
URL: *https://www.mdpi.com/2223-7747/8/11/468*

Saleem, M. H., Potgieter, J. and Arif, K. M. (2019b). Plant disease detection and classification by deep learning, *Plants* **8**(11): 468.

Singh, S., Gupta, S., Tanta, A. and Gupta, R. (2021). Extraction of multiple diseases in apple leaf using machine learning, *International Journal of Image and Graphics* p. 2140009.

Srinidhi, V. V., Sahay, A. and Deeba, K. (2021). Plant pathology disease detection in apple leaves using deep convolutional neural networks : Apple leaves disease detection using efficientnet and densenet, *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 1119–1127.

Tulshan, A. S. and Raul, N. (2019). Plant leaf disease detection using machine learning, *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–6.