Identifying Diseases In Mulberry Leaves That Affects Silk Production: A Deep Learning Approach

MSc Research Project
Data Analytics

Himanshu Ashok Duragkar
Student ID: X20210639

School of Computing
National College of Ireland

Supervisor: Qurrat Ul Ain

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Himanshu Ashok Duragkar
Student ID:	X20210639
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Qurrat Ul Ain
Submission Due Date:	15/12/2022
Project Title:	Identifying Diseases In Mulberry Leaves That Affects Silk Production: A Deep Learning Approach
Word Count:	5241
Page Count:	17

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:	Himamnshu Duragkar
Date:	29th January 2023

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):	

Identifying Diseases In Mulberry Leaves That Affects Silk Production: A Deep Learning Approach

Himanshu Ashok Duragkar
X20210639

Abstract

In many places around the globe, sericulture is recognized as one of the most important segments of the agriculture industry, and it has achieved success. It is believed that the best species in the sericulture ecology is the mulberry. In order to boost output as much as possible, the use of pesticides and fertilizers was widespread, which negatively affected the condition of the land, air, and water and made it challenging to protect plants from a variety of pests and diseases. The use of image classification allows for this suggestion gives us a formula, tactics, and comprehensive detection of diseases in mulberry leaves using deep learning. This model is able to identify and find mulberry plant illnesses.

1 Introduction

Any nation, no matter how big or small, has a portion of its population that works in the agriculture industry, either to meet basic food needs or as a main source of export revenue. Nowadays, agriculture provides a lot more than simply food; it also significantly boosts the economies of so many countries. The usage of fertilizer, pesticides, and herbicides to boost yield has a significant impact on soil, water, and air. Sericulture, often referred to as "silk farming," is quite prevalent in several nations in the Middle East and south of Europe, as well as in the nations of Southeast Asia.

As the climate is tropical, this is the main reason these plants are grown in these regions. Silk is produced from mulberry plants, which are prone to failure due to a variety of diseases. Silk which is included in some of the finest natural fabrics in the world is only 9–10 micrometers thick. The silk story is simple to understand. Silkworms mostly consume mulberry leaves as food. They finally convert into cocoons. The cocoons are harvested. The string-like cocoon fiber can have a maximum length of one kilometer. 1 kg of silk is created from nearly 5500 cocoons. 1 kg of silk costs around $50 to $55 in the international market. Hence the ecosystem of sericulture depends heavily on the leaves. At a normal rate, if the quality of the silk produced is not up to par, the farmer has to bear up to 40% of its total expenditure.

The design of the cocoon is closely related to the quantity and quality of the leaf. The elements of a cocoon leaves and silkworms—all rely on one another. The majority of farmers neglect the soil structure and other lifeless elements, which have an influence on the effectiveness of the plants' responses. The plant became starved, which caused the outcomes to be unsatisfactory. An appropriate and calculated amount of organic nutrients can provide the best final product possible. This can be achieved when the farmers have an idea about "Are there any kinds of diseases found on the plant?". If yes, then what treatment can be given to improve the quality of the leaf, as a result of which the silk quality could be at its best? If the harvest is of poor quality, it will be impacted. Agriculturalists and cultivators have to bear their losses, which will indirectly affect the next cultivation as they won't be able to make money to invest in the next stage as it is a cycle.

Early diagnosis of these conditions is crucial to preventing further loss. The usual method of recognizing these scenarios is observation, although expert guidance, ongoing supervision, and an in-depth understanding of such conditions are also helpful. This can only result in 50% to 60% accuracy. Farmers in many nations lack the means and accessibility to access professionals who can advise them on how to handle their produce. With the help of this research, a solution will be provided that will ease the process. A cutting-edge technique called deep learning has been used to perform image processing, grouping, and analysis in this proposed research. The following proposed architecture will help cultivators and farmers classify whether the leaf is affected by the disease or not, which will result in saving a ton of money and making high profits.

1.1 Research Question

How accurately a capsule neural network model can predict diseases in a mulberry leaf?

1.2 Research Objective

The main objective of this proposal is to assist farm owners or cultivators in determining whether they allow silkworms to feed a specific mulberry plant in order to extract silk from it by finding out which leaf is infected with the disease or not. Deep learning will be used to create a model that will categorize the leaves as infected or noninfected. If an illness is discovered, it is important to treat it effectively. This classification will help the cultivator decide how to make up for the losses and improve the silk's quality by giving proper treatment to the infected plants, which will directly result in good quality silk and high profits.

2 Related Work

Hiary et al. (2011) provided a brief analysis of K-means clustering for many plants, including cotton, blueberries, apples, and others. Clusters of different photographs were produced using a collection of features split into K classes. K stands for clustering, which divides the image into four clusters, each of which has the potential to get infected. The boundaries were hidden using pixels. This methodology, in the author's opinion, provides the most accurate and quick classification and recognition of plant diseases.

Bashish et al. (2010) has proposed a framework where they have classified diseases on various plants with respect to their leaves and stems. The proposed algorithm will classify the images into five different diseases. The proposed framework is divided into a few steps. Images are segmented using k means clustering, where the image is divided into 4 clusters. Each cluster is checked for one of the five diseases. Feature extraction is done using the color co-occurrence method to get more details about the images. Next, the segmented images are passed into a trained neural network model with 10 layers.

93% accuracy is achieved using this model.

Chaudhari and Malathi (2021) conducted research and came to the conclusion that it is difficult for farmers and product manufacturers to identify and classify diseases of rice plants. A summary of numerous procedures and methods used in the identification and classification of illnesses that affect rice grains and leaves. Researchers came to the conclusion that artificial neural networks and support vector machines provide the best accuracy for disease classification.

(Jaware et al.; 2012) found out that traditional K means clustering has some shortcoming and it is influenced by noise present in the images. Hence, a noise filter is introduced in this proposed method. This filter is applied at the image pre-processing stage. SGDM matrices are generated for each pixel. K means is introduced in such a manner that it can identify clusters weather the input is original space or subspaces. The outcomes of the experiment demonstrate the effectiveness and high clustering accuracy of the proposed method.

In order to forecast the yield of the mulberry plant using the soil characteristics and potential for disease, Srikantaiah and Deeksha (2021) examined random forest regression (RF), multiple linear regression, and ridge regression. The results of the experiment using the Random Forest regressor with 94.6% accuracy, which is the highest.

With image processing and the use of various leaves, Sivabalaselvamani et al. (2019) has identified and classified many plant species. This study looks at sophisticated picture management frameworks for recognizing, measuring, and regulating plant poisonings using digital photos in the undeniable range. Fluffy C-implies clustering and a support vector machine classifier have actually been built to classify and arrange plant diseases. The test findings show the value of the suggested methodology as a theory that might help with quick and accurate diagnosis of leaf diseases.

Francis et al. (2016) observed that farmers were unable to make decisions on what to do when a peppers plant became infected based solely on observations made with the unaided eye. Since pepper is one of the most significant contributors to the Indian spice industry, it needs to be constantly monitored. MATLAB was used for the image processing, which included converting the image's colours between RGB to HSV and then to masking. The plant's injured ratio was computed. If the affected ratio is much less than 5%, an infection was considered to have impacted the plant. Neural networks are also used by the algorithm to determine the type of sickness.

A convolution neural network (CNN) is used by Hema et al. (2019) to detect diseases in mulberry plants. A model was trained with the labeled data, and a comparison was done using the neural network model in a sequential form. To increase the processing of the model the images were converted into black and white format.

It was discovered by Liu et al. (2018) that four different apple tree diseases were suddenly becoming more prevalent in China. These illnesses immediately impacted the harvest of apples, causing a dramatic drop in apple quantity and quality as well as economic losses. Regular classifiers like the Support Vector Machine, Random Forest, and

K-Nearest Neighbor (KNN) were producing fewer accurate results. As a result, the authors provided a model of a convolutional neural network to identify viruses at an early stage of growth. Techniques for digital image processing, including brightness modification, picture rotation, and principal component analysis filtering, were used. The outcome was 97.62%, which was better than the predictions of the conventional models.

Capsule neural network (capsnet) was proposed by Sabour et al. (2017) as an alternative neural network system to address the shortcomings of Convolution neural network (CNN) . In addition to requiring a lot of data to train, CNN also loses a lot of data in the space between it's maximum pooling layers. CNN also needs additional components and is unable to directly connect the relationships between the pieces. CNN does not place much emphasis on the spatial arrangement and alignment of picture components. Capsnet is used to solve these issues.

Kurup et al. (2020) examined the performance of two datasets based on plants using capsule nets. Plant disease diagnosis and the classification of plant leaves are the two duties that are taken into consideration in this work. They examine 54,306 pictures of 14 different crop species to diagnose plant diseases. There are 38 classes in total, with 26 disorders and apparently healthy cases in each class. Classifying plant leaves is the second duty. This collection includes 2997 pictures of 11 different plants. For We conducted our research on a limited number of photos because extensive computer power was needed. Accuracy achieved was 94% and 85% each respectively.

In Nigeria, it was found by the Oladejo and Ademola (2020) that the banana plants were affected by some types of diseases at a large scale. A comparison was done between Capsnet, CNN, LeNet5 and ResNet50 where it was found that Capsnet achieved 95% of accuracy which is highest against all for classification of diseases on leaves. Convolution layer with kernel size was defined as 9. Images were rotated for 30,60 and 90 degrees with 90 degrees rotation of image.

2.1 Identified Gaps and conclusion

The orientation and position of the objects are not encoded, making it difficult for CNN to identify entities when their positions change, according to the majority of the studies mentioned above. CNN struggle to categorize photos with various positions. Deep learning model like CNN requires a lot of training data. CNN is slower due to max pooling operation. As a result, the following proposed research attempts to bridge these gaps by replacing the max pooling algorithm with dynamic routing.

3 Methodology

Following a thorough evaluation of the literature, this model is implemented using a KDD (Knowledge Discovery in Databases)-inspired methodology (Fig. 1). There are six different steps in it. In order to arrive at the final results, which classify photographs of mulberry leaves as infected or not, a series of events must be followed. A deep learning technique called a "capsule neural network" is used to feed the model with scientific data.

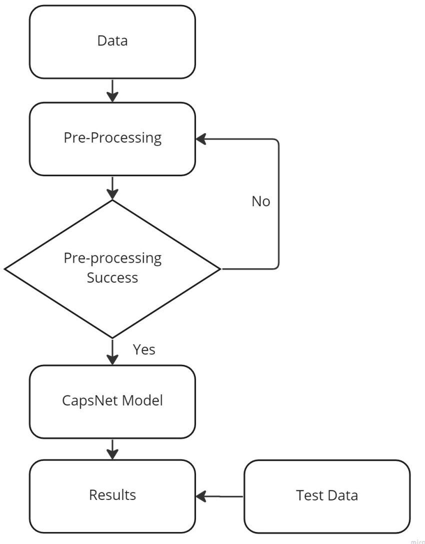

Figure 1: KDD approch which describes the proposed project flow

3.1 Data

The data used for this research is taken from the Multi-agent Intelligent Simulation Laboratory, which is publicly available on the internet. [1] The photographs (Figure 2) of the mulberry leaves were taken in their natural context. This data repository contains images from 10 cultivators from Taiwan, Turkey, and Australia. Images are taken from different perspectives. As the data was not labeled, the annotation, i.e., the separation of the leaves into infected and non-infected categories, was performed manually. Every leaf image is captured in JPEG format. The photos of mulberry leaves were downsized to 224 * 224 pixels. The data collection did not involve any sensitive information or human subjects that would have contravened moral or legal rules.

Figure 2: Mulberry leaves pictures from there natural acclamation present in the dataset.

3.1.1 Data Pre-processing

Before the data is applied to the model, it needs to be pre-processed. Data normalization is the initial stage in pre-processing, and it is crucial because it ensures that every

[1] http://misl.it.msu.ac.th/?page_$id = 225$

input element, such as an image's cells, have an uniform data distribution. The model would train more quickly if an image is normalized. Image Augmentation is the next pre-processing stage, when image size is enhanced without using additional new photos to train a model. This will enable a model to be trained on a huge amount of data, allowing it to be effectively trained on the training set of data and so improving model efficiency. This has been followed by a thorough explanation of image enhancement and normalization.

Normalization: In order to ensure precision, data normalization is a fundamental process that is often used in deep learning programs. By using this step paradigm, training will go more quickly and lead to improved results. Every pixel value is normalized within 0 and 1 during data normalization. Training, validation and testing data are re-scaled by 1./255 in this study by Nayak et al. (2021) .Grey scale and black-and-white photos are the two main categories of images. These Black-White pictures are entirely black, or "0," and entirely white, or "1." The values range from "0" to "1." Images in gray scale have many tones of gray. These numbers fall within the exact range from 0 to 1 range as well. While 0 is a black , 0.1 is a somewhat darker shade, and 1 is white. Each input pixel of the picture data set used by the Artificial intelligence system is standardized, which will improve the accuracy of the training model. Results are produced using the scaled standard deviation values that are obtained by subtracting and separating the pixel distributions by standard deviation.

Data Augmentation: The model must be properly trained if higher performance is to be obtained. More data to learn to model, can make this achievable. Despite adding additional images to the model, image augmentation enlarges existing ones. Image or data augmentation is performed only on training data . It plays a very important roles in enhancing the final results Nayak et al. (2021) Image augmentation involves changing an image by specifying different parameters, including rotation, scaling, shifting, and zooming. The below figure 3 shows the example for image augmentation.

Figure 3: Image after applied data augmentation method

3.2 CapsNet (Capsule Neural Network)

Capsule network is a group of neurons together referred to as a "capsule" which indicates the formation parameters of a certain sort of thing, such as an asset or an item. Here, the word "capsule" refers to a layer that is nested within another layer of capsule networks (Figure 4). Each single capsule reports only to a particular single capsule in the layer next.

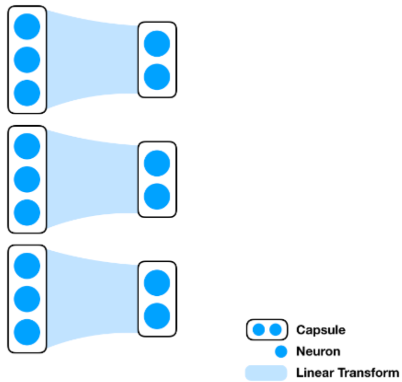

Figure 4: Architecture of a capsule inside a neuron

A capsule can retrieve geographical information and other crucial properties to prevent information loss during pooling operations of CNN. The specifications of an object's features are determined by capsules. The pooling layer in convolution neural networks brings important input from the background to the front line. Since the data is pooled for transmission to the following layer, the network might not be able to pick up on minute details.

Rendering is a concept defined under computer graphics. It means to consider different internal representation of an entity like it's scale, rotation and position, and converting image on screen of theirs. Our brain works opposite to this called as inverse graphics. When a human brain looks at any object, it on it's own deconstruct the object into different sub parts and tries to develop a relational between them to make a whole object out of that. This is the way a human tries to recognize an object and as a result, our ability to recognize items doesn't at all depend on how the orientation is. This is the base of Capsule neural network.

The neural output of CNN also includes a scalar value. Due to the fact that capsules hold numerous neurons, they provide vectorial output with the same size but various routings. The attributes of the photographs are represented by a vector's routings. Activation Function of scalar input like Tangent, Sigmoid, and ReLu are used by CNN. Inversely Capsnet uses vectorial activation function called squashing(eq. 1) Sabour et al. (2017).

$$v_j = \frac{||S_j||^2}{1 + ||S_j||^2} \frac{S_j}{||S_j||} \tag{1}$$

Here, For the capsule j, v_j the output. Total input for the capsule is indicated by S_j. If there is an entity in the image, v_j reduces long vectors to 1 and squeezes short vectors towards 0. Capsule S_j's input value is obtained by symmetric sum of the prediction vectors $U_(i|j)$ for the capsules which are located in the lower level . $U_(i|j)$ (prediction vector) is estimated by multiplying lower level capsule to it's output O_i and weight matrix W_{ij} (eq. 2) Sabour et al. (2017).

$$S_{j=} \sum_i b_{ij} u_{j|i}$$

$$u_{j|i=} W_{ij} O_i \tag{2}$$

Dynamic Routing is called as coefficient b_{ij} and calculated as follows (eq. 3) Sabour et al. (2017):

$$b_{ij} = \frac{\exp(a_{ij})}{\sum_k \exp(a_{ik})} \tag{3}$$

In the above equation, b_{ij} is the log probability. Softmax is the probability of the log prior from the summation of the correlations between capsule 1 and the capsules on the top layer, which is equal to 1. To identify whether entities of a specific class are available in capsule, a margin loss that can be estimated using the following (eq. 4) Sabour et al. (2017) . has been proposed.

$$L_k = T_k \max\left(0,\ m^+ - \|v_k\|\right)^2 + \lambda(1 - T_k)\ \max\left(0, \|v_k\| - m^-\right)^2 \tag{4}$$

If the class K is available then only the value of T_k will be 1. m + = 0.9 ve m = 0.1 are the hyper-parameters. They indicate degradation of the loss. The directions of the vector contain the parameter information, such as colour, position, size, text, etc, while the lengths of the vectors produced in the capsule shows the chance of being in that section of the image.

4 Design Specification

4.1 Capsule Neural Network

The proposed architecture has 5 convolutional layers. The intention for setting up 5 layers is to allow the primary layer to gather more features from the input data. The given figure 5 below shows the proposed architecture.

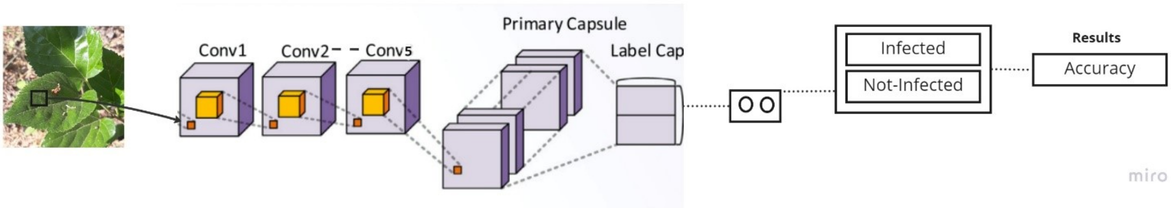

Figure 5: Capsule Neural network design Architecture

The proposed design architecture has different layers such as data, convolutional, capsule layer, and classification which will be together held to get the final results.

1st layer - Data : On this stage all the operation related to data collection, pre-processing and EDA is performed in this layer. The is collected from the Multi-agent Intelligent Simulation Laboratory(MISL).

2nd layer - deep learning Layer : Transfer learning model such as VGG16 is built on python language combing it with deep learning model Capsule neural network.

Output Layer: Final evaluation is reflected in this layer with the identification of diseases on the mulberry leaves which will be helpful to the cultivators and the farmers for cultivation.

4.2 (VGG) Visual Geometry Group

VGG or Visual Geometry Group is a basic Convectional Neural Network whose aim is to work as a transfer learning model and also for feature extraction. Deep here refers to the number of layers in the model. In this proposed model, VGG16 is implemented. VGG16 can support up to 16 convolutional layers. For the proposed model 5 convolutional layers have been added. VGG16 can classify images into 1000's different categories. VGG19 is also used in some preferred techniques where the model can support up to 19 convolutional layers. Visual Geometry Group is also called VGG. It could be used for feature extraction, fine-tuning, and prediction, It has 16 weighted layers. Following below is figure 6 general Architecture for VGG. [2].

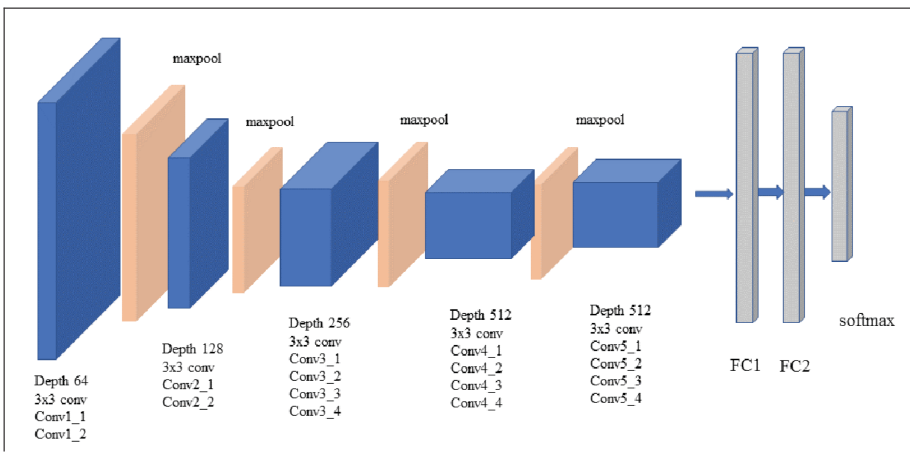

Figure 6: VGG Model Architecture

[2]https://towardsdatascience.com/ extract-features-visualize-filters-and-feature-maps-in-vgg16-and-vgg19-cnn-models-d2da6333edd0. Accessed: 2010-09-30

5 Implementation

The working of the Capsnet is divided into three important parts, which have sub-operations in each.

5.1 Convolutional layer

VGG model consists of a fully connected layer, model weights, a convolution layer, different types of optimizers, a dense layer, and a network layer, which helps perform feature extraction. The below proposed model is implemented in the Python programming language. A multiclass categorical classification is considered for this study, in which data is divided into infected and noninfected images for training and testing purposes, respectively.

The first step in VGG-16 is to obtain Imagenet's weights. For this, we have input an image size of (299,299,3). The model will be trained once, it will not be trained twice. Hence the layer of the VGG16 is set to 'FALSE'. A classification layer would be introduced at the conclusion of the model to train the dataset images. As a result, all the layers up to this point have been stopped, and the output layer, the classification layer, will be removed. To do this, we will smooth the layer and create a 512-layer system that is fully connected and has the activation function "ReLu." The last max pooling layer (designated by 9 x 9 x 512) is considered to constitute the feature extraction phase of the model, starting with the input layer. A dropout layer with a rate of 0.5 is implemented to avoid overlapping. For classifications, the output layer is inserted with the help of the Softmax function. The final results of the VGG model are used as input to the capsule neural model. The features that have been extracted by the VGG are in terms of pixels, which are different in both categories, i.e., infected and noninfected. Each convolutional filter has a 3x3 kernel, a stride of 1, and an unoverlapping 2x2 pooling region. You should take note of the fact as we decrease the twin 4096-dimension completely linked layer.
When a transfer learning model is applied to a large data set, the first step is to reduce the model's overfitting. The most practical way to do this is to drop out. As the model is 16 layers deep and the data set is huge, dropout is defined as 0.5 for a fully connected layer. The outcome indicates that this method has significantly reduced the error rate. The working of the VGG model is easily understandable in the below-given figure 7. [3]

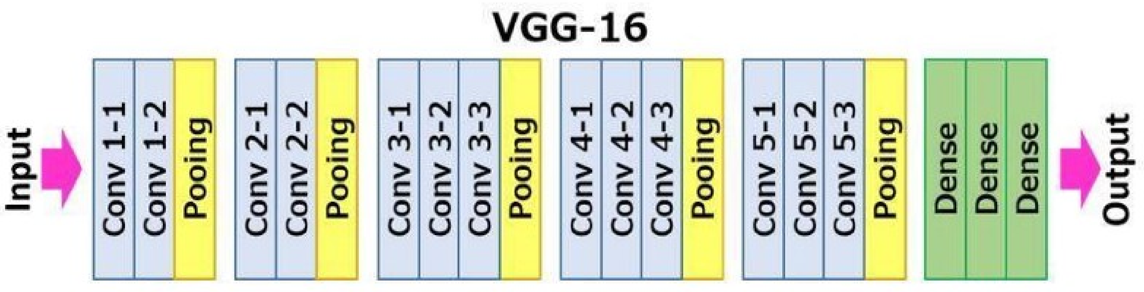

Figure 7: VGG Model block diagram

[3]https://medium.com/analytics-vidhya/face-recognition-using-transfer-learning-and-vgg16-cf4de57b9154. Accessed: 2010-09-30.

The below table gives the summary for the model summary of the VGG (Visual Geometry Group) with 16 layers.

Layer (type)	Output Shape	Param#
input_1($InputLayer$)	(None, 299, 299, 3)	0
block1_$conv1$($Conv2D$)	(None, 299, 299, 64)	1792
block1_$conv2$($Conv2D$)	(None, 299, 299, 64)	36928
block1_$pool$($MaxPooling2D$)	(None, 149, 149, 64)	0
block2_$conv1$($Conv2D$)	(None, 149, 149, 128)	73856
block2_$conv2$($Conv2D$)	(None, 149, 149, 128)	147584
block2_$pool$($MaxPooling2D$)	(None, 74, 74, 128)	0
block3_$conv1$($Conv2D$)	(None, 74, 74, 256)	295168
block3_$conv2$($Conv2D$)	(None, 74, 74, 256)	590080
block3_$conv3$($Conv2D$)	(None, 74, 74, 256)	590080
block3_$pool$($MaxPooling2D$)	(None, 37, 37, 256)	0
block4_$conv1$($Conv2D$)	(None, 37, 37, 512)	1180160
block4_$conv2$($Conv2D$)	(None, 37, 37, 512)	2359808
block4_$conv3$($Conv2D$)	(None, 37, 37, 512)	2359808
block4_$pool$($MaxPooling2D$)	(None, 18, 18, 512)	0
block5_$conv1$($Conv2D$)	(None, 18, 18, 512)	2359808
block5_$conv2$($Conv2D$)	(None, 18, 18, 512)	2359808
block5_$conv3$($Conv2D$)	(None, 18, 18, 512)	2359808
block5_$pool$($MaxPooling2D$)	(None, 9, 9, 512)	0

5.2 Primary capsule

The inverse graphics procedure is carried out on the first layer of the capsule. When a leaf image is fed to a primary capsule, it is divided into n number of different parts. Each component is regarded as a capsule in this context. For better understanding, let's assume that to depict the outcome of the capsules, arrows are used. These arrows (Fig. 9 [4]) are positioned in each section of the image, and their output indicates whether or not the infection is present there or not. The output of one subpart is denoted by black arrows, while the output of another subpart is denoted by blue arrows. The below-shown figure 7 indicates that the length of the arrow is longer where the infection is found in the input picture and shorter where it is not. The posture of the arrow specifies the alignment of that disease (rotation, scale, position, etc.) in the provided image, while the length denotes whether the disease is there.

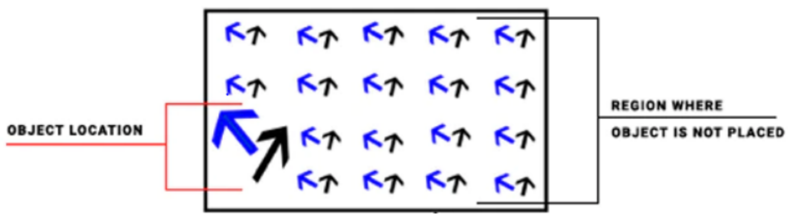

Figure 8: Infection detection in the image

[4]https://www.intel.com/content/www/us/en/developer/ articles/technical/understanding-capsule-network-architecture.html. Accessed: 2010-09-30.

Bizarrely, if the infection is gently rotated or shifted in the image, the arrows that represent the infection will also rotate. Equivariance is a minor change in the input that results in a minor change in the output of the matching capsule. That makes it possible for the capsule networks to identify the disease in a given image together with its exact rotation, scale, position, and other positional properties. This process is called "dynamic routing." It finds the feature as well as the location of the disease.

5.2.1 Squash function

The primary capsule layer of the model has 32 primary capsules whose aim to take features extracted from the convolutional layer of the VGG model and construct various combinations of the extracted features. Every capsule generates a 6x6x8 output tensor by applying eight 9x9x256 convolutional kernels (with stride 2) to a 20x20x256 input volume. The output volume is 6x6x8x32, as there are 32 of these capsules. By doing calculations identical to those in the previous layer, we obtain 5308672 parameters in this layer. The output received is reshaped into an 8-dimensional vector. Hence the shape would be a 6*6*32 capsule. After this, every capsule will be passed through squash (a nonlinear function) to maintain the length of the output vector either 0 and 1

5.2.2 Loss function

Margin loss means that if a disease is present on the leaf image provided then the squared length of the equivalent vector of that disease's capsule must not be less than 0.9 and if the disease is not present then the vector should be less than 0.1.

5.3 Higher Level Capsule

Here, dynamic routing or routing by agreement takes place. This layer makes sure that the output of every capsule is sent to the respective parent layer above. At first, the output is routed to every possible parent, but it is scaled to the coupling coefficient that gives the sum 1 Sabour et al. (2017). The capsule calculates the prediction vector through the multiplication of its own weight matrix for every possible parent. Top-down feedback occurs when this prediction vector has a significant scalar product with the output of a potential parent, increasing the coupling coefficient for that parent while decreasing it for other parents. This raises the capsule's participation with that parent, which in turn increases the scalar product of the prediction of the capsule and the output of the parent.

6 Evaluation

This section describes the complete evaluation of the implemented deep learning model, the capsule neural network. Three experiments were carried out with 1, 10, and 15 epochs each, which are discussed below. The metrics that are considered here for evaluation are classification report, precision, accuracy, recall, F1-score, and loss function.

6.1 Hyperparameter tuning

To avoid the problem raised by the vanishing gradient, an activation function called 'relu' is used at the input stage. 'Softmax' function is used at the output stage. To avoid overfitting, 'dropout' function is implemented, whoes range should always be between 0.2 and 0.8. For the VGG model, it should be 0.5. Adam optimizer is used to change weights and also to reach to the global minima while model training. If the model is stuck in local minima while training, adam's job is to help the model get out of local minima and get into global minima. This ultimately helps reduce losses. As in the proposed research, multiclass categorical crossentropy is used as a loss function to reduce prediction error.

6.2 Experiment 1

In the first experiment, the capsule neural network model is built using 5 convolutional layers. Initially, the model was run with 1 epoch and 120 iterations per epoch to determine whether the model was correctly built and executed. The validation accuracy that was achieved was 61.88% with a data loss rate of 0.62%.

6.3 Experiment 2

In the second experiment, the pre-trained model was trained with an increase in the number of epochs to 10, with 120 iterations per epoch. This experiment performed relatively well as compared to experiment 1 with a resulting accuracy of 82.04% and a data loss rate of 0.37%. Figures 10 and 11 show the graphs for the achieved accuracy and loss graphs with respect to the given number of epochs. Table 1 shows the classification report.

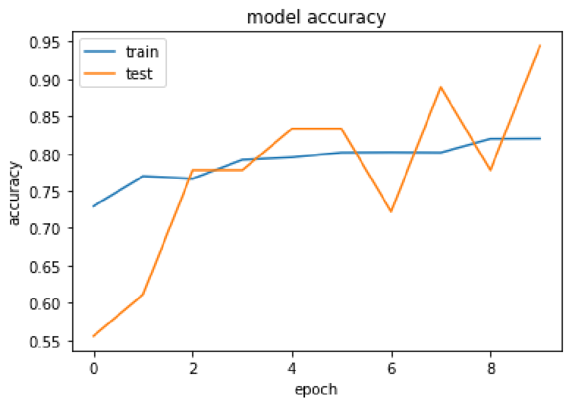

Figure 9: Graph for Accuracy vs Epoch

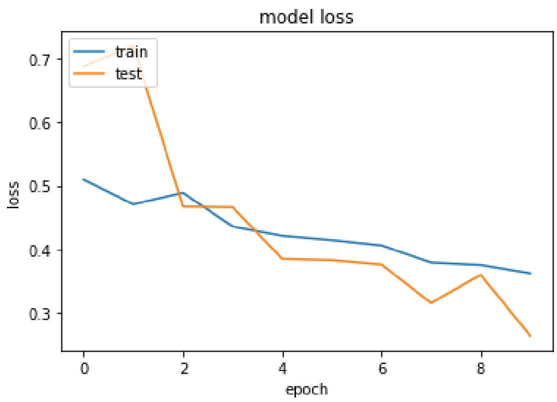

Figure 10: Graph for Loss vs Epoch

	precision	recall	f1-score	support
Infected	0.82	0.90	0.86	299
Non infected	0.82	0.71	0.76	202
Accuracy			0.82	501
Macro Avg	0.82	0.80	0.81	501
Weighted Avg	0.82	0.82	0.82	501

Table 1: Classification report for experiment 2

Noticing the below figure 12 for the confusion matrix for a model with 10, 268 leaf images are correctly classified as infected while 31 are not, though they are named infected. In terms of noninfected leaves, 59 images are classified as infected while they are not. And 143 images are noted as noninfected correctly.

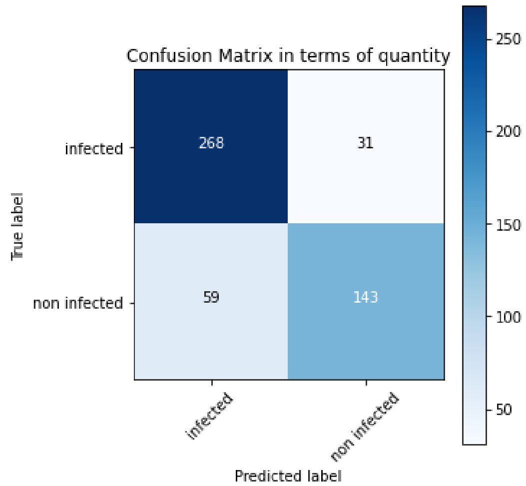

Figure 11: Confusion Matrix in terms of quantity for experiment 2

6.4 Experiment 3

The third experiment was carried out while increasing the number of epochs to 15 and 120 iterations per epoch. The accuracy noted was less than the 2nd experiment i.e 81.64% but the data loss measured a less amount of 0.12%. The below Figures 13 and 14 depict the graphs for accuracy and loss with their respective epochs. Also, table 2 explains the classification report.

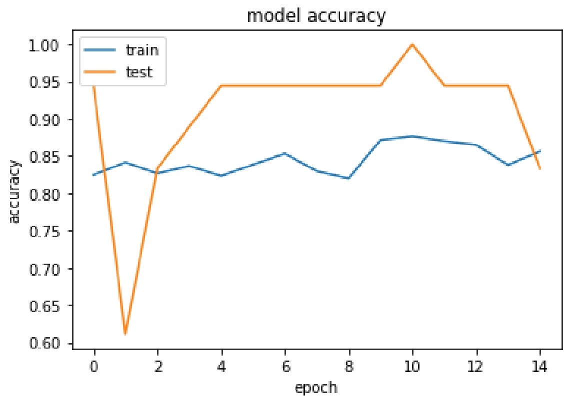

Figure 12: Graph for Accuracy vs Epoch

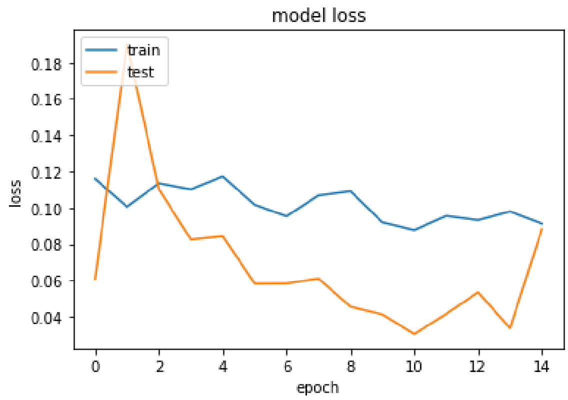

Figure 13: Graph for Loss vs Epoch

	precision	recall	f1-score	support
Infected	0.77	0.98	0.86	299
Non infected	0.96	0.57	0.71	202
Accuracy			0.82	501
Macro Avg	0.86	0.78	0.79	501
Weighted Avg	0.85	0.82	0.80	501

Table 2: Classification report for experiment 3

The confusion matrix for experiment 3 (Fig. 15) demonstrates that the model is classifying 294 photos as infected, which are indeed infected, and 5 images as non-infected, yet they are infected, as opposed to the actual infected 294 images. While 115 images were found to be completely infected, 87 images were classed as infected but they are not.

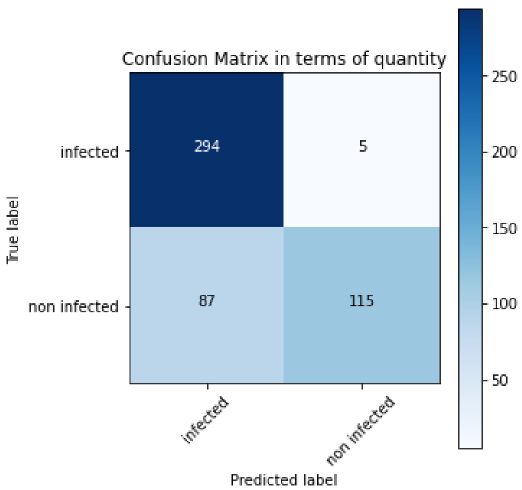

Figure 14: Confusion Matrix in terms of quantity for experiment 3

6.5 Discussion

With the results found in the 6th section, all the pre-trained models achieved different results on a different number of epochs. It is noted that the proposed deep learning technology is enhancing the accuracy of the proposed model as the number of epochs increases to the 10th epoch (Experiment 2). At the 11th epoch, it has been seen that, though the data loss is less, the accuracy gradually decreased (Experiment 3) as compared to the 10th epoch. In the point of view of image classification, models operating at 15 epochs have incorrectly identified some photos, whereas models running at 10 epochs have successfully classified the majority of the images provided to them. The curve structures of experiment 2 denote that the test data set is unrepresentative, i.e., it is not able to learn/capture more statistical characteristics as compared to the train. This could happen if there aren't enough instances in the testing dataset compared to the training dataset. The probable reason for the model to perform differently is due to illumination, background clutter, and the viewpoint of the images. The purpose of the research is to classify the leaves on the basis of whether they have any diseases or not. The above-mentioned experiments state that they resulted in accuracy and recall for the given inputs. There are a number of metrics that can be used to solve the problem of the proposed research, including precision, recall, and F1-score. However, since the objective of the problem is to reduce the number of diseased leaves that are incorrectly categorized as healthy, recall is the more crucial metric to pay attention to. There aren't many data sets available. The proposed model can perform better if the images provided are of good quality in terms of background noise and brightness levels.

7 Conclusion and Future Work

Capsule neural network is a newly proposed deep learning technique introduced as an alternative to CNN. Finally, the classification of mulberry leaves was the primary goal of this research. To achieve this research question and the objective, an innovatory approach is followed, which includes data pre-processing, as mentioned in Section 3.1.1. Section 5 mentions the proposed model for building a capsule neural network with the help of the transfer learning model VGG16, which is the second step, Lastly, all the results and evaluation performed are explained in Section 6. As stated in Section 6.2, the model is able to classify mulberry leaf images with 82.04% accuracy. This will help the farmers decide whether they should cultivate the respective crop in order to increase silk production and gain more profit. Hyperparameter such as relu is used to avoid any problem that took place by raising of vanishing gradient. For over-fitting, the dropout function is used, which should always range between 0.2 and 0.8. To change weights and reach global minima, the Adam optimizer is used. If in case the the model gets stuck in local minima, Adam's function is to help model get out of local and put into global. Which indirectly reduces loss. To enhance the model's performance, more data can be used to train it. More transfer learning models can be explored. Other data augmentation techniques can be included. Nevertheless, in the future, this model can be utilized to create a mobile application that will enable more growers to easily complete the above-described process wherever they are using a smartphone.

Acknowledgment

I want to express my gratitude to Professor Qurrat Ul Ain for giving me the opportunity to work on this project. She helped me do this task, for which I am really appreciative, and her guidance and aid also enabled me to effectively complete this proposal. I'm grateful to all the people and things that made this research possible.

References

Bashish, D. A., Braik, M. and Bani-Ahmad, S. (2010). A framework for detection and classification of plant leaf and stem diseases.

Chaudhari, D. J. and Malathi, K. (2021). A survey on rice leaf and seedlings disease detection system, *Indian Journal of Computer Science and Engineering* **12**.

Francis, J., D, A. S. D. and K, A. B. (2016). Identification of leaf diseases in pepper plants using soft computing techniques.

Hema, D. D., Dey, S., Krishabh and Saha, A. (2019). Mulberry leaf disease detection using deep learning, *International Journal of Engineering and Advanced Technology* **9**.

Hiary, H. A., Ahmad, S. B., Reyalat, M., Braik, M. and ALRahamneh, Z. (2011). Fast and accurate detection and classification of plant diseases, *International Journal of Computer Applications* **17**.

Jaware, T. H., Badgujar, R. D. and Patil, P. G. (2012). Crop disease detection using image segmentation, *World Journal of Science and Technology* **2**.

Kurup, R. V., Anupama, M. A., Vinayakumar, R., Sowmya, V. and Soman, K. P. (2020). Capsule network for plant disease and plant species classification, Vol. 1108 AISC.

Liu, B., Zhang, Y., He, D. J. and Li, Y. (2018). Identification of apple leaf diseases based on deep convolutional neural networks, *Symmetry* **10**.

Nayak, S. R., Nayak, D. R., Sinha, U., Arora, V. and Pachori, R. B. (2021). Application of deep learning techniques for detection of covid-19 cases using chest x-ray images: A comprehensive study, *Biomedical Signal Processing and Control* **64**.

Oladejo, B. F. and Ademola, O. O. (2020). Automated classification of banana leaf diseases using an optimized capsule network model.

Sabour, S., Frosst, N. and Hinton, G. E. (2017). Dynamic routing between capsules, Vol. 2017-December.

Sivabalaselvamani, D., Selvakarthi, D., Rahunathan, L. and Leninpugalhanthi, P. (2019). Detection and classification of plant pathology with image processing technique, *International Journal of Scientific and Technology Research* **8**.

Srikantaiah, K. C. and Deeksha, A. (2021). Mulberry leaf yield prediction using machine learning techniques, Vol. 4.