

A COMPARATIVE ANALYSIS OF CONVOLUTIONAL NEURAL NETWORK MODELS FOR CHILI LEAF CURL VIRUS DETECTION

Muhammad Mureed^{*1}, Shumaila Afzal², Mughairat-Ul-Islam³, Nadar Hussain⁴, Fatima Haya⁵,
Muhammad Hammad⁶

^{*1,3,4,6}Department Information Technology Centre, Faculty of Agricultural Social Sciences, Sindh Agriculture University,
Tandojam, Pakistan

²Department Of Botany, University of Agriculture Faisalabad, Pakistan

⁵Department Of Computer Science, Islamia College Peshawar, Pakistan

^{*1}muhammadmureed02@gmail.com, ²shumailaafzal397@gmail.com, ³mughairatulislam@gmail.com,
⁴nadarhussain02@gmail.com, ⁵fatimahaya.pcr0011@gmail.com, ⁶shaikhhammad723@gmail.com

DOI: <https://doi.org/10.5281/zenodo.22723899>

Keywords

Chili Leaf Curl Virus,
Convolutional Neural Network
(CNN), Deep Learning, Image
Classification, Data
Augmentation, Adam Optimizer,
Binary Cross-Entropy, Plant
Disease Detection.

Article History

Received: 01 June 2026

Accepted: 23 August 2026

Published: 12 September 2026

Copyright @Author

Corresponding Author: *

Muhammad Mureed

Abstract

A Convolutional Neural Network (CNN) model specifically designed for the detection of the chili leaf curl virus, a serious hazard to chili plants, is developed and evaluated in this work. The research commenced with the assembly of a dataset comprising 80 images each of healthy and virus-afflicted chili leaves, which were meticulously labeled and resized for consistency. To overcome the scarcity of the data, data augmentation techniques were used to generate more data for the model. Data was divided into a training, validation and test set and a CNN architecture with three convolutional layers, ReLU activation function and max-pooling was designed. The model was optimized with the Adam optimizer and loss function as a binary cross-entropy was monitored throughout the training process along with key indicators of model performance or KPIs (loss, accuracy, etc.). A total of eight variants of Convolutional Neural Network (CNN) models were developed for the detection of chili leaf curl virus, which poses a great threat to chili plants. The CNN architectures were systematically developed with varying numbers of dense layers, dropout rate, activation function and other parameters to evaluate the accuracy of the classification of Healthy and Virus infested chili leaves into the CNN model. All variants were rigorously trained and evaluated, with additional testing on independent sets to get unbiased performance metrics. The results show that the results obtained by the different variants vary in their accuracy, which ranges from 50% to 100%, and the loss values are taken into account according to the models' predictions. Some variants showed potential performance with high accuracy while others showed some weakness either by underfitting or overfitting, showcasing the difficulty of the classification task. All things considered, this thorough investigation offers insightful information about how regularization strategies and architectural decisions affect CNN performance. The comprehensive documentation presented in this study offers transparency and clarity

regarding the model's performance, paving the way for its potential application in practical scenarios involving chili leaf curl virus detection.

1.1 INTRODUCTION

Chili (*Capsicum annuum* L.) is not only a highly regarded agricultural vegetable crop, but is well recognized for its peppery flavour, attributed to capsaicin, and as a source of essential minerals (K, Zn, Mg, Co), vitamins (C, A, K) and phytochemicals (alkaloids, phenolics, limonoids). In addition to culinary uses, chili is also used medicinally, with capsaicin as an active ingredient in Ayurvedic and allopathic medicines to treat several diseases, including cancer, diabetes, and obesity, and its relative, cayenne pepper, providing numerous other health benefits via the presence of isoprenoid compounds such as carotenoids. The production of chillies in Indonesia in 2021 was 1.39 million tons (Indonesia Central Bureau of Statistics [BPS], 2022), which has decreased in amount by 8.09% from 2020, mainly due to inefficient water and fertilizer use, and wasteful conventional irrigation. Production has since recovered: national output rose to approximately 1.48–1.55 million tons in 2022, and total Indonesian chili production across all types reached an estimated 3.05 million tons by 2023 (Statistics Indonesia, 2024, as reported via Statista).

To address these challenges, farmers are encouraged to adopt fertigation – the controlled, combined application of water and fertilizer within precision agriculture – which can significantly improve fertilizer efficiency and crop production, helping secure the future of chili cultivation (Padmini et al., 2023).

1.2 Symptoms associated with Chili leaf curl virus disease

The transmission of viral diseases is affected by any number of factors such as the type of virus, insect vector and environment factors. In tropical and subtropical regions, chili plants suffer devastating losses – sometimes as much as 100% – due to chili leaf curl virus disease, driven by the emergence of new viral strains, the rapid spread of the whitefly 'B' biotype, breakdown of host resistance,

ineffective insecticides, and an increased vector host range.

While Geminiviruses are generally not seed-transmitted, an exception is the Pepper Yellow Leaf Curl Indonesia Virus (PepYLCIV), which has been found to be seed-transmissible: molecular analysis of seedlings and embryos grown from PepYLCIV-infected chili seeds accumulated from various locations showed varying degrees of PepYLCIV DNA-A (25% to 67%) and DNA-B (50% to 100%), though the possibility of seed transmission warrants further investigation.

Infection results from a tripartite interplay among host plants, insect vectors, and viruses. Begomoviruses spread persistently and circulatively by whiteflies, Jasmonic acid (JA) is involved in host resistance to whitefly and begomovirus infection and begomovirus infection has been shown to down-regulate the transcription of JA-responsive genes; enhancing whitefly survival and reproduction. The virulence factors of Begomovirus also suppress terpene production, impairing resistance of the host and spreading the virus. Control of the whitefly vector using insecticides has been shown to be only partially effective and has high costs for the producers themselves, risks to the farmers, consumers and environment, and produces residues that reduce the export value of affected plants (Manoj Kumar Nalla et al., 2023). Chili leaf curl virus (CLCuV) is a disease which causes significant losses in chili crop worldwide and machine learning (ML) approaches have been proposed as a potential solution to the diagnosis and prediction of the disease. The following review explores current research employing diverse ML algorithms for the comparative evaluation of CLCuV.

To the best of the author's knowledge, Nageswara Rao Naik et al. (2022) used a Squeeze-and-Excitation based convolutional neural network (SECNN) for classification of the chili leaf disease symptoms by employing twelve different pretrained deep learning networks both with and without augmentation; DarkNet53 with

augmentation had the highest accuracy of 98.82% which indicates that deep learning can be used effectively to detect chili leaf diseases. In a similar study, Patil & Lad (2020) created an automated system for the prediction of chili plant leaf diseases using support vector machine (SVM) and k-nearest neighbours (KNN) classifiers. They used a technique of texture feature extraction known as GLCM and classified the images with competitive results to traditional image-processing techniques. In another comparative study, Saputra et al. (2024) were able to generate chili leaf images and detect them using CNN with feature extraction using the model VGG19 transfer learning and SVM classification to obtain training accuracy of 95% and testing accuracy of 90%. Hybrid CNN-SVM pipelines can outperform the use of classical machine-learning algorithms in chili leaf image data. Additionally, Vasavi Pallepati et al. (2023) assessed several machine learning algorithms for the prediction of crop diseases, concluding that hybrid and ensemble methods could be fruitful avenues for increasing the accuracy of the crop disease prediction using any one of the classifiers, and recommending their use in future research efforts. In addition to image-based techniques, molecular approaches have also been used to characterize CLCuV directly. For example, Amna Fayyaz et al. (2022) used CTAB-based DNA extraction and Rolling Circle Amplification (RCA) to analyze the genomic diversity of Chili Leaf Curl Virus (ChLCV) isolates infecting chili crops in Faisalabad, Pakistan, shedding light on the molecular structure and diversity underlying CLCuV infection.

Overall, this research underscores the potential of machine learning strategies in comparative evaluation of CLCuV. From deep learning-based image classification to ensemble learning and molecular data evaluation, ML techniques offer versatile tools for accurate diagnosis and understanding of CLCuV dynamics. But there are still some issues to be resolved in order to tackle mal-distribution in datasets, generalization across environments, and scalability for practical usage. Cultivation of chili pepper *Capsicum annuum* is an important agricultural enterprise and has

multiple economic, cultural and culinary significance. It has been grown since prehistory, when it was domesticated in the Americas (Kraft et al., 2014) and was used by preeminent civilizations like the Aztecs and the Mayans before being traded around the world. Genetically, chili has good diversity, resulting in chili cultivars with different characteristics (Bulle et al., 2025). Due to the development of molecular genetics and breeding, new varieties with higher yield potential, disease resistance, and flavor have been produced. The cultivation practices involve land preparation, selection of appropriate seeds, irrigation, fertilization, pest management and the implementation of integrated pest management and precision agriculture for optimizing productivity while reducing ecological impacts (Zahari et al., 2021). With the changing climate, technological innovations and consumers' preferences, chili cultivation is still facing a number of multifaceted problems of pest attack and diseases that demand sustainable control strategies (Sam-on et al., 2024).

1.3 Problem Statement

India's reduced output is generally due to crop diseases, which farmers frequently find hard to detect and control. Consequently, reducing losses calls for the development of sophisticated methods for early infection diagnosis (Vasavi Pallepati et al., 2023). As a great commercial crop, chili is grown in tropical and subtropical regions. Chili production is being threatened by the Chili Leaf Curl Virus (CLCuV), which is spread through whiteflies. The two primary factors influencing the epidemic process – vector migration rate and host-vector contact rate – have been linked to helping control it.

It has been shown that complete interception of migratory vectors soon after transplantation enhances the plants' life duration (to remain free of infection) by up to 80%, delaying the spread of the epidemic. About 800,100 tonnes are produced by growing it on 2,298,076 acres, according to Nageswara Rao Naik et al. (2022). The more recent data indicates that the chili cultivation area had been expanded in India: During 2023-24, the

total chili cultivation area is approximately 8.09 lakh hectares (About 809 thousand ha or 2 million acres) and the production of chili was estimated as 29.13 lakh tonnes (About 2.91 million tonnes) with an average productivity of 3,273 kg/ha (Chili is the highest producer in the world) (Spices Board of India, 2024). In this project, an image-based CNN model with multiple variants will be developed to forecast crop illnesses in chilies.

Scope:

The current study focussed on the development and evaluation of a Convolutional Neural Network (CNN) model to predict chili leaf curl virus in chili vegetation. This comprises (1) images are collected and preprocessed (80 healthy leaves images and 80 infected leaves images were labeled and all images were resized to 256x256 pixels); (2) Data augmentation (Rotation, flipping, zooming and shifting were used for data augmentation); (3) splitting the dataset into Training set, Validation set and Testing set (70%/15%/15%); (3) Proposal of a CNN architecture comprising of three CNN layers followed by ReLU activation and a max-pooling; (4) Training the proposed CNN and evaluation of it based on the binary cross-entropy loss, Adam optimizer and using accuracy/precision/recall/F1-scores along with early stopping; (5) evaluation of the proposed CNN on the testing set with visualization and (6) documentation of the process. In this work, the development, training, evaluation and documentation of a CNN-based chili leaf curl virus detection model are aimed at, with the focus on transparency, repeatability and applicability.

1.4 Objectives

- To create a machine learning model (CNN)
- To compare its results with other models

1.5 Methodology

The present study aimed to develop a Convolutional Neural Network (CNN) model using a dataset of 80 healthy and 80 chili leaf images to predict chili leaf curl virus after augmenting and splitting the dataset into training,

validation and testing sets. Eight CNN variants were trained using early stopping to avoid overfitting, trained with binary cross entropy loss, and compiled with Adam optimizer, and tested with accuracy, precision, recall, and F1 score. The entire methodology (data preparation, model architecture, training procedure and evaluation criteria) is provided in Chapter 3.

1. REVIEW OF LITERATURE

Seelam & Sai Abhishta Roy (2023) evaluated CNN, RNN, and MLP architectures and their variants for diagnosing chili plant illnesses. Among the RNN models, RNN 1 (90% training, 10% testing, single hidden layer) achieved the highest accuracy of 91%. The best-performing CNN, CNN 3 (70% training, 30% testing, 3 hidden layers), reached a test accuracy of 72.50%, while the best MLP model achieved only 20% test accuracy with an 80%/20% split. In their comparative analysis, RNN 1(0.1) was determined to be the best technique for detecting chili plant illnesses.

V. Krishna Pratap & N. Suresh Kumar (2023) evaluated multiple deep learning models – VGG-16, DenseNet-121, ResNet-50, MobileNet-V2, and EfficientNet-B4 – for monitoring chili leaf diseases. EfficientNetB4's adapted architecture addressed challenges posed by variation in leaf size, shape, and texture, as well as differing pathogens. In terms of accuracy, F1-score, precision, and recall, their customized EfficientLeafNetB4 outperformed the alternative models, enabling the timely response and early identification crucial for focused disease management strategies.

Dyah Kurniawati et al. (2023) reduced Fourier Transform Infrared (FTIR) spectra for chili plants separately using Principal Component Analysis (PCA) and Discrete Wavelet Transform (DWT), and classified the reduced data with classifiers: Support Vector Machine (SVM), Linear Discriminant Analysis (LDA), and Random Forest (RF) with a train-test split of 80%–20%. However the result of the reduction using the PCA method was of highest accuracy (79.17%) obtained when 5 principal components were employed with both

classifiers namely LDA and RF, while the result of the reduction using the DWT method was of higher accuracy (95.83%) at 4th decomposition level with the same classifiers.

Rudi Muslim et al. (2023) developed an Android-based disease detection system for rice and chili plants using CNN-based image classification, incorporating MobileNet, Sequential CNN, and TensorFlow Lite models. The MobileNet and Sequential CNN models achieved training, validation, and testing accuracies of 0.8827, 0.8493, and 0.8678 respectively, and field tests correctly identified rice and chili plant illnesses with 95% accuracy.

Achmad Naila Muna Ramadhani et al. (2023) classified 5 types of chili leaf diseases using Deep CNN and the MobileNetV2 pre-trained model. MobileNetV2 trained with a learning rate of 0.001 achieved a reasonably good accuracy of 91%, though misclassifications persisted due to damaged or biased sample data. A separate comparison found the CNN architecture reaching 84% accuracy versus MobileNetV2's 92%, a difference attributed to less suitable hyperparameter selection in the CNN; the authors suggest data augmentation and better hyperparameter tuning to improve the CNN's performance.

Fangfang Li et al. (2022) noted that TYLCV poses a significant and growing threat to tomato production in China and other temperate regions internationally due to its rapid worldwide spread, and recommended that future research focus on cutting-edge control strategies such as CRISPR/Cas9 systems and RNA interference technology to develop plant resistance.

It is observed that automated identification and detection of disease for G-4 chili leaves proves to be one of the effective methods which significantly reduces errors in disease detection from the G-4 (Guntur-4) chili leaves into G-4 chili leaf illnesses with 97.56% accuracy compared to manual disease detection by Das Chagas Silva Araujo et al. (2021).

Amna Fayyaz et al. (2022) mentioned that Chili Leaf Curl Disease (ChLCD) is an important disease of chili due to the Begomovirus complex

in worldwide. Begomovirus, known as Chili Leaf Curl Virus (ChLCV) has monopartite genome containing 6 ORFs (2754nt) and a betasatellite (1.3kb). Genomic DNA from infected chili leaf samples collected in Faisalabad was extracted using the CTAB technique, and circular molecules within it were then amplified using the Rolling Circle Amplification (RCA) technique.

2. MATERIALS AND METHODS

This chapter describes the materials and methods that were used to develop, train and evaluate the CNN model, described in the following steps.

3.1 Data Collection and Preprocessing

Collect 80 chili leaf healthy pictures and 80 chili leaf with chili leaf curl virus. Ensure all photos are labeled and scaled to similar sizes (such as 256x256) and are scaled correctly as are the other photos.

3.2 Data Augmentation

Due to the limited amount of data perform data augmentation techniques such as rotating, flipping, zooming and shifting the data to artificially augment the data set and improve model generalization.

3.3 Splitting the Dataset

Data Split Break down the Data to training, validation and test sets. The complete dataset was separated into three parts, with 70% allocated for training and 15% each for validation and testing.

3.4 Model Architecture

- Create a basic model of CNN (Convolutional Neural Network) model for the task
- Input Layer: Convolutional layer with 32 filters and kernel size (3x3) and activation of ReLU.
- Hidden Layers: Add more convolutional layers, with more filters (64, 128) and more filter size (3x3).
- Parameters of ReLU activation (End of each Conv Layer).
- Pooling Layers: To get down-sampling feature maps, the Max-pooling layer was applied after the convolution layer.

- Flatten Layer: In this Layer the Output of convolutional is flattened.
- Fully Connected Layers: It is create one or more fully connected (dense) layer(s) using an appropriate unit.
- Output Layer: In this layer the output neuron with a sigmoid activation function, for binary classification.

3.5 Model Compilation

The problem being a binary classification one, we will use binary cross entropy loss and Adam optimizer. Monitor other metrics of performance when training, including accuracy, precision, recall and F1 score.

3.6 Training the Model

Compile architecture and train the model on the training data. Prevent overfitting with early stopping – halt training after a set number of epochs if the validation loss does not decrease or stops improving. Try out different learning rates and batch sizes to see which ones work best when training.

3.7 Model Evaluation

Test trained model on validation set to check the performance and adjust hyper parameters if needed. Apply confusion matrix analysis and other classification metrics to get a detailed understanding of the model's performance.

3.8 Model Testing

After seeing the overall performance of the model on the validation set, assess it on the previously unseen testing set to get independent performance numbers. Evaluate results and provide accuracy, precision, recall, F1 score and other parameters as appropriate.

3.9 The model interpretation and further analysis will follow.

Use sample images from the test set, and visualize the model's predictions to understand how the model reacts to the images. Examine misclassifications for possible improvement areas. Look into it with transfer learning and/or fine-

tuning to larger datasets if needed to achieve good performance.

3.10 Documentation and Reporting

Record the entire system including dataset collection, preprocessing, model architecture, training procedure, and evaluation outcomes. Provide clear explanations and insights into the model's overall performance and potential implications for practical use in detecting chili leaf curl virus.

3.11 Understanding CNN Architecture

Description and explanation of Convolutional Neural Networks (CNNs / ConvNets) in detail. A deep learning model that is well suited for classification problems using images is called Convolutional Neural Networks (CNNs). CNNs consist of learnable weights and biases, as well as convolutional, pooling and fully connected layers which transform the pixel values of the input to class scores. The convolutional layers learn the filters to identify the local features, the pooling layers reduce the dimensionality of the feature maps and retain the most relevant information, and the fully connected layers combine the features to generate a classification output. Parameters are optimized through gradient descent so that the predicted class scores align with the training labels.

3.12 TensorFlow

In this study, the deep learning models implemented and trained in the CNN are represented by a TensorFlow framework, which is a computation and mutable state graph. The design enables efficient distributed training, places the ability to parallelize on available hardware, and is well suited with iterative experimentation on a variety of model architectures.

3.13 Rectified Linear Units (ReLU)

The CNN models used in this study adopt the ReLU activation function $f(x) = \max(0, x)$ which cuts off any outputs below zero. . Compared to earlier activation functions such as sigmoid, ReLU accelerates convergence during training and helps

mitigate vanishing-gradient issues, which is why it is combined with softmax or sigmoid at the classification layer in this work.

3.14 Data Augmentation for Improving the Model

Given the limited size of the dataset (80 images per class), data augmentation techniques including rotation, flipping, zooming, and shifting were applied to artificially expand the training data and narrow the gap between training and validation error. This brought about a decrease of overfitting and contributed to models generalizing better to other data, along with regularization techniques like dropout and early stopping.

3.15 Dataset

The Chili Leaf Curl Virus data is taken from Kaggle website and consists of images categorized

as healthy and Leaf Curl Virus Affected. Each class includes 80 pictures, all in JPG format. The data set is useful for image analysis and classification algorithm development and testing for disease detection in plant diseases and is significant for researchers and practitioners in the field of plant pathology. The number of pictures used in each class is equal, making the training and testing information balanced, which will be beneficial for good model performance. By including both healthy and diseased images, full learning is possible, and algorithms can learn to differentiate between healthy leaves and symptoms associated with the Leaf Curl Virus. The dataset can be used to enhance plant diseases' early detection and control and ensure better crop yield and sustainability for researchers.

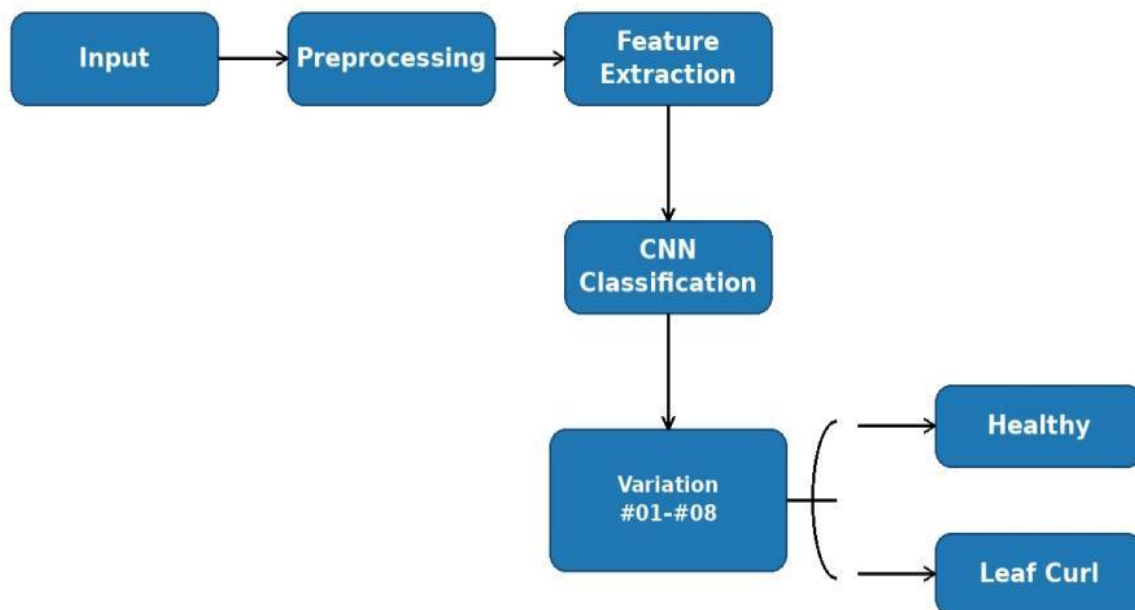


Figure 3.1: Architecture of CNN for Chili Leaf Curl Virus Prediction

The procedure given may be a systematical method of diagnosing possible disease such as Leaf Curl Virus in plant leaves. Starting with photo input, it progresses through preprocessing to refine photo quality, and feature extraction to identify relevant patterns. Utilizing CNN classification, with multiple variants, the system predicts whether the

leaf is healthy or affected, culminating in a comprehensive evaluation pipeline for accurate disease detection.

3.16 Prototype Explanation

As illustrated in Figure 3.1, the prediction pipeline proceeds through four stages. The first input to the

system is the image of a chili leaf obtained by capturing or loading it to a host computer. The image is resized to a standard dimension in the pre-processing stage, the intensity of each pixel is normalized and any outlier values are discarded from the image to be able to process it. Feature extraction then pinpoints important features in

the image, including texture, shape and colour patterns. Finally, a CNN model – one of the eight variants developed in this study, denoted “Variation #01” and so on – classifies the extracted features and predicts whether the leaf is healthy or affected by Leaf Curl Virus, based on patterns learned during training.

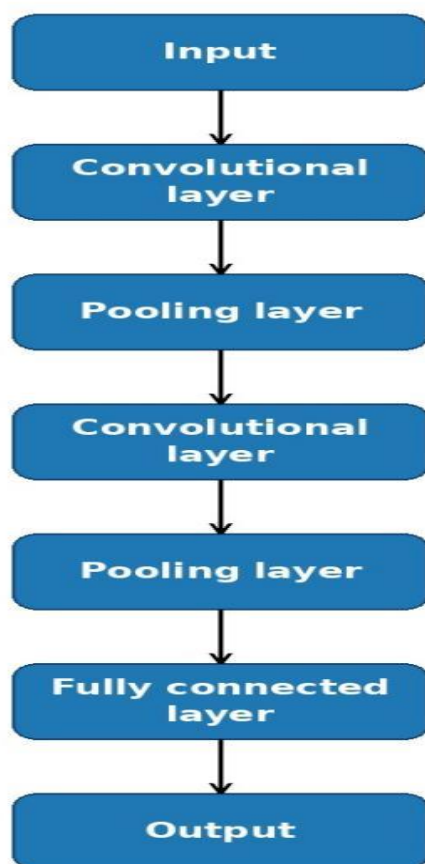


Figure 3.2: CNN Layers

Figure 3.2: CNN Layers Figure

Fully connected layer; POOL: A 2×2 max-pooling layer; CONV: A leaky ReLU three $\times$ three

convolutional layer; M: a collection of control factors. The range of the characteristic maps are listed underneath each block.

Table 3.1: CNN Variant

This is a table of the different variants of CNN with input shapes, dense layers, max pool and dropout for 8 variants.

Variant	Input shape	Dense layers	Max pooling	Dropout	Accuracy
1	100*100*3	16,32,64,64, & 128	(2, 2)	0.1,0.2,0.3	81.85%
2	100*100*3	32,64,128	(2, 2)	0.5	50%

3	100*100*3	32,64,128	-	0.5	50%
4	100*100*3	16,32,64,64,128	(2, 2)	0.5	80%
5	100*100*3	32, 64,128	(2, 2)	0.5	65%
6	100*100*3	32, 64,128	(2, 2)	0.5	70%
7	100*100*3	64,64 Output layer	(2, 2)	0.5	100%
8	100*100*3	32,32, & 32 (Output layer Sigmoid)	(2, 2)	0.5	100%

3.17 Sigmoid & Softmax

The activation functions in the output layer are Sigmoid, Softmax in all the eight CNN variants used in the study. The Sigmoid map is used for binary classification, has values as input, maps the value to a range of [0,1] and therefore, is ideally suited to represent the probability of being in a class. Multi-class problems are reconciled by using softmax functions to convert a vector of scores into vector of probabilities of sum 1. They are nonlinear, differentiable and both are essential in the CNN models to make accurate classification decisions based upon probabilities, and are appropriate for gradient-based optimization methods such as stochastic gradient descent (SGD).

3. RESULTS

This chapter presents the results obtained from training and evaluating the eight CNN variants described in Chapter 3, including their architecture, training parameters, and resulting loss and accuracy.

4.1 First Variant of CNN

This input shape refers to the main Convolutional variant that trains this shape. The dense layers in this variant are (16, 32, 64, 64, 128) and Max Pooling with (2, 2). For this purpose, 0.1, 0.2 and 0.3 dropouts will be tested on each layer of and for the dense layer on both the train and test sets.

Total params :> 4828481 (18.42 MB)

Trainable params :> 4828481 (18.42 MB)

Non-trainable params :> 0 (0.00 B)

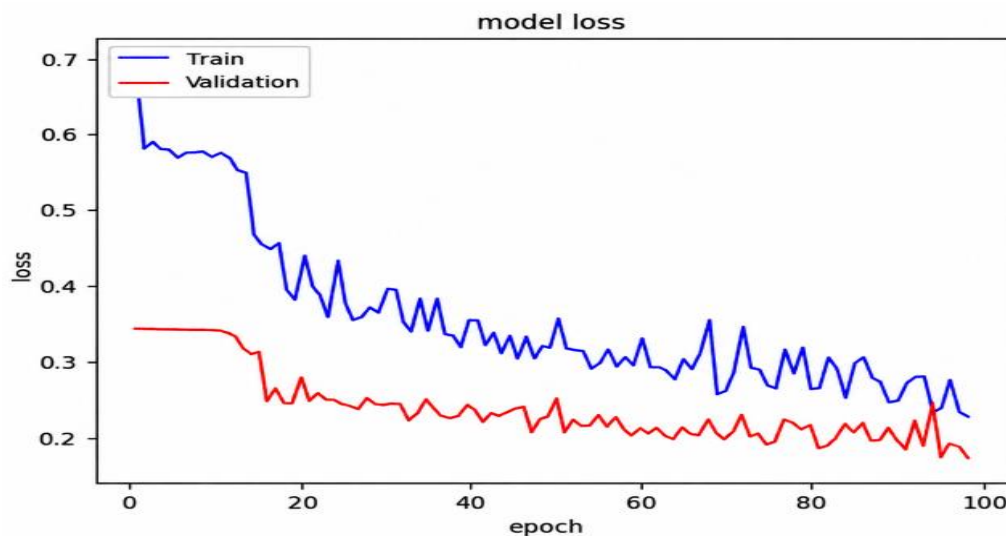


Figure 4.1: First Variant of CNN

The Figure illustrates the architecture of the first CNN variant described above, showing the

arrangement of its dense layers and pooling operations.

4.2 Second Variant of CNN

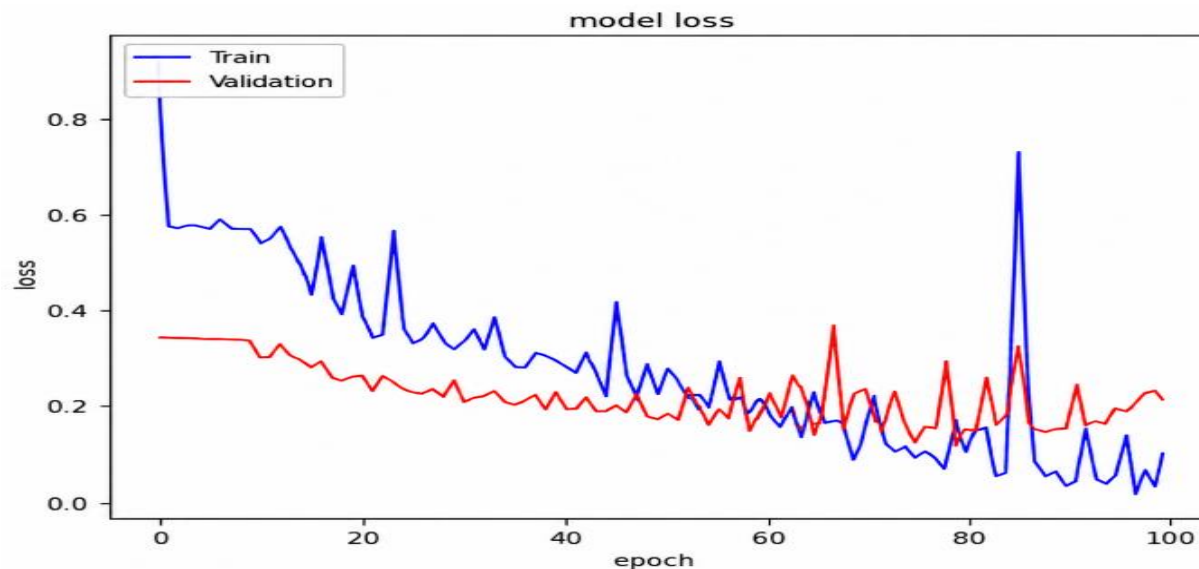


Figure 4.2: Second Variant of CNN

This is the second model of CNN. Another form of CNN2. The second CNN is composed of a 100x100 sized input and a dense layer of size 32,

64, 128. A dropout of 0.5 is applied to check whether it affects the accuracy of the model, this time mainly the flatten layer of the model.

4.3 Third Variant of CNN

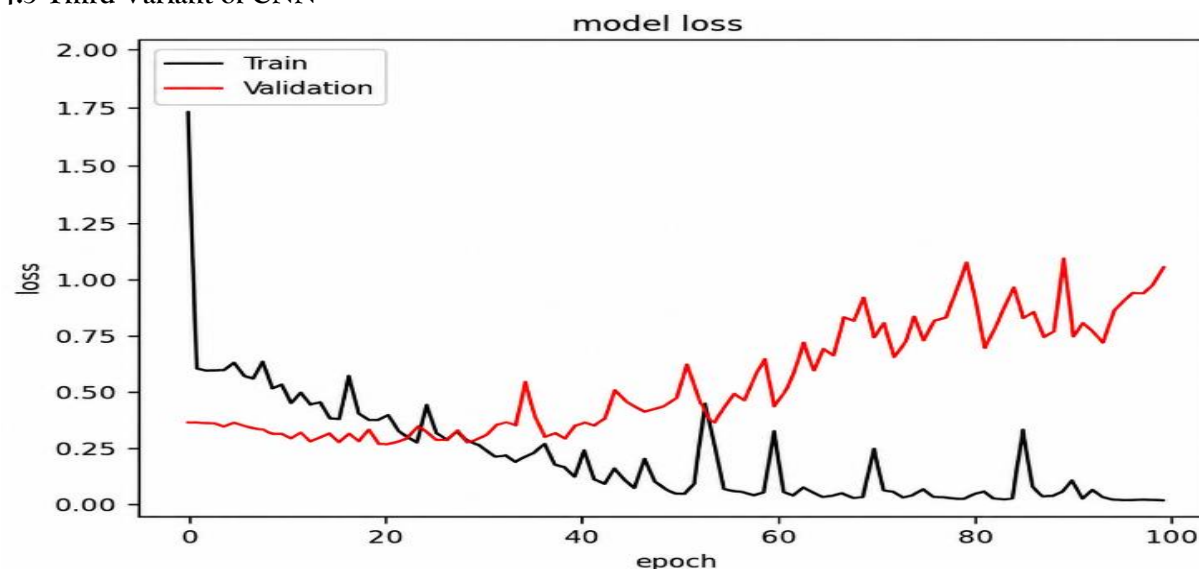


Figure 4.3: Third Variant of CNN

It is different, with size $\gg 32 \times 64 \times 128$, input $\gg$ shape $100 \times 100 \times 3$, and max pool size $\gg 2 \times 2$. Learn

to have a dropout rate of 0.5 across the network to make it more generalizable, avoid over fitting.

4.4 Fourth Variant of CNN

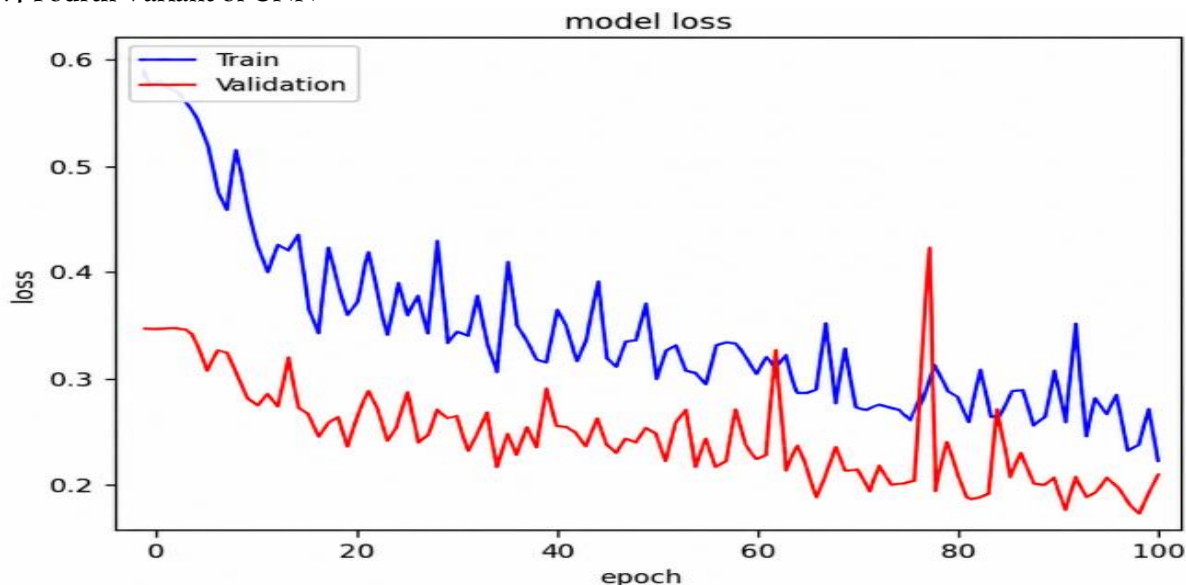


Figure 4.4 (A): Fourth Variant of CNN with Sigmoid model loss

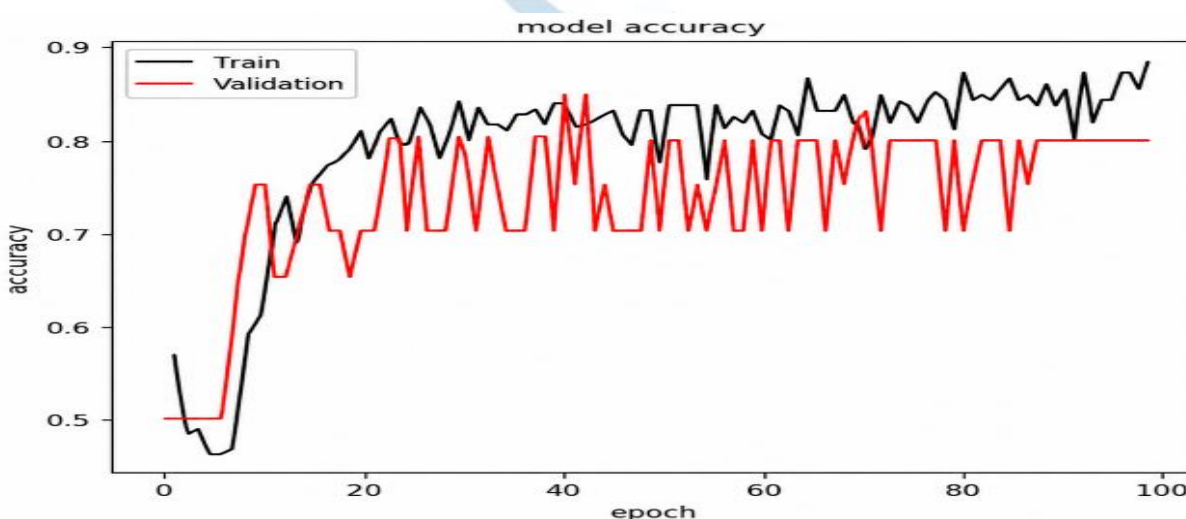


Figure 4.4 (B): CNN model accuracy in sigmoid activation with fourth variant.

The Input Shape set to $100 \times 100 \times 3$, while a 2×2 Configuration is used for max pooling. The consists of layers of CNN with sizes $\Rightarrow 16, 32, 64, 64$, & 128 , & the final layer uses the Sigmoid activation function. At the 0.1, 0.2, 0.3 points, testing is to be performed for training and

testing for drop out. The development of the model loss in the visualization is presented in the figure 4.4 (A), and the dynamics of the model's accuracy in the process of training and testing is shown in the figure 4.4 (B).

4.5 Fifth Variant of CNN

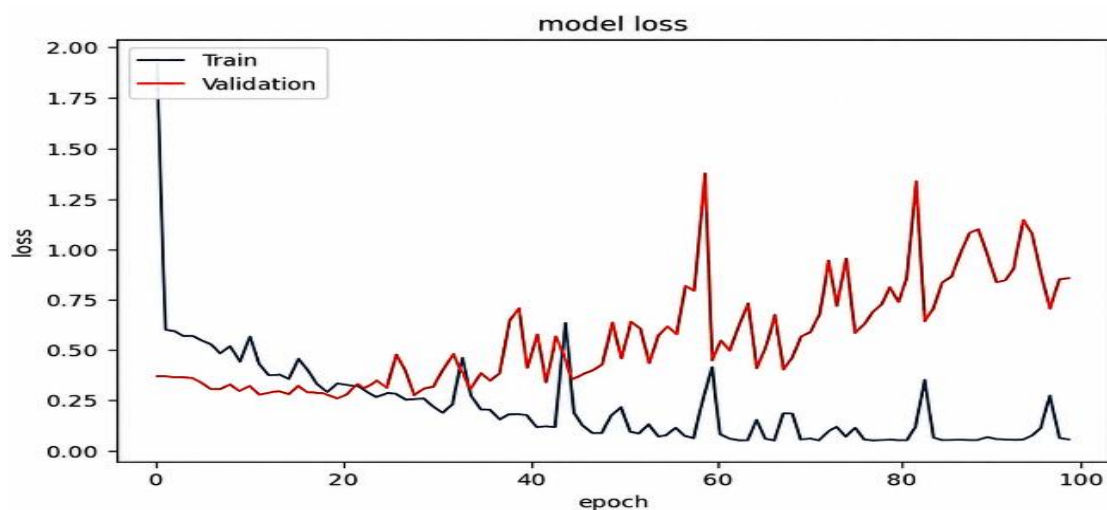


Figure 4.5 (A): Fifth Variant of CNN model loss

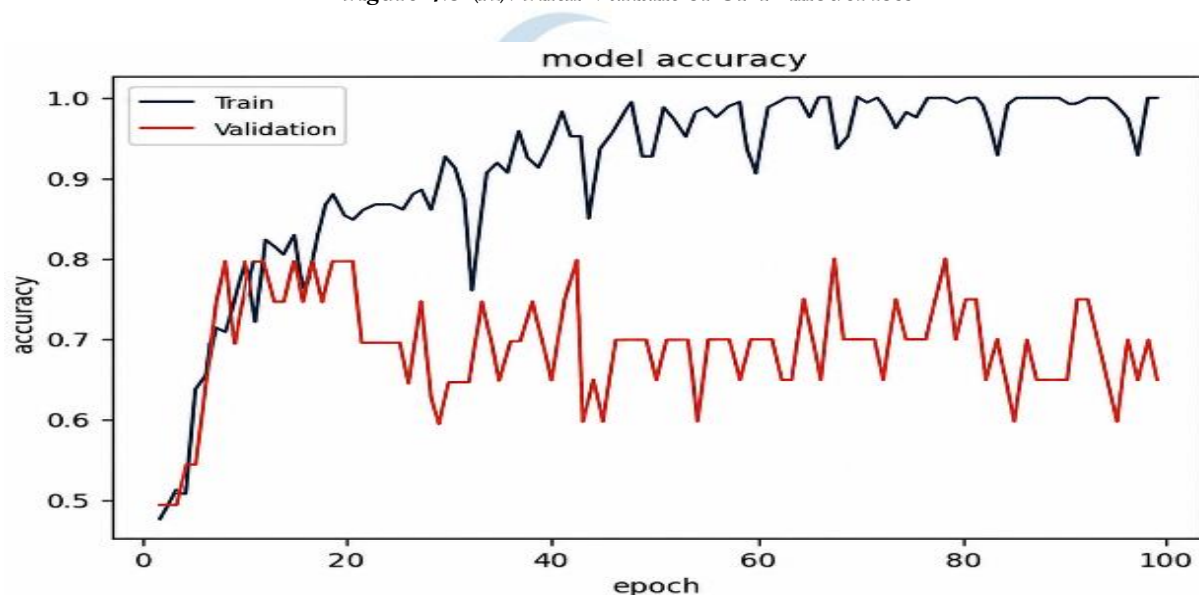


Figure 4.5 (B): Fifth Variant of CNN model accuracy

The fifth variant consists of 32, 64 and 128 dense layers with activation = sigmoid at the output layer, the input shape is = (100, 100, 3) and at max pooling 2x2. The dense layers are subjected to dropout of 0.5 to determine the impact of flatten

layer. Figure 4.5 (A) visualizes the model's loss progression, whilst Figure 4.5 (B) illustrates its accuracy development during the course of training and evaluation.

4.6 Sixth Variant of CNN

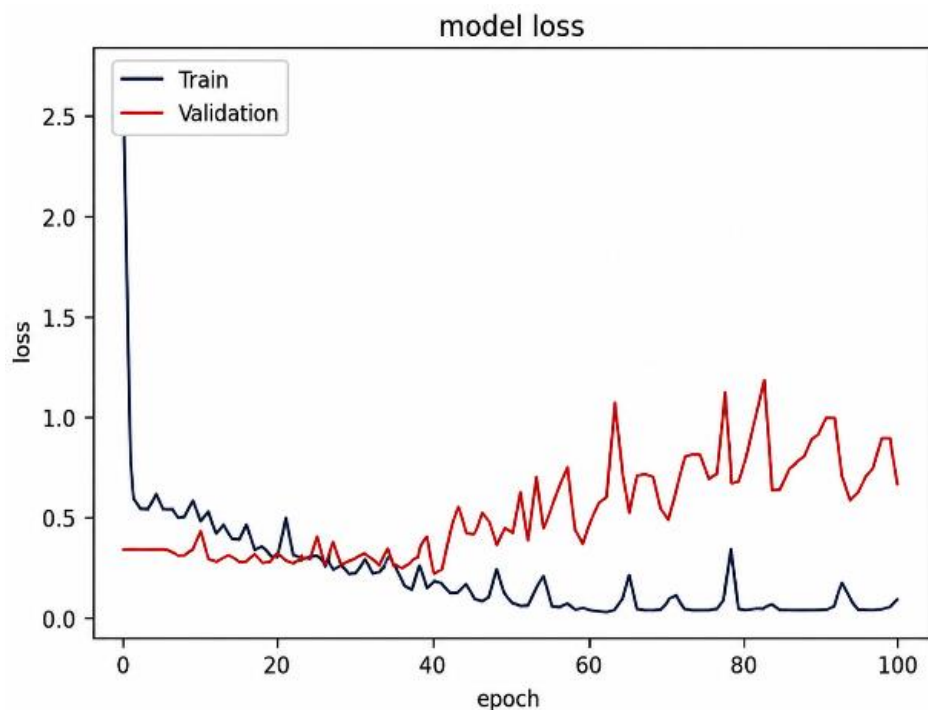


Figure 4.6 (A): Sixth Variant of CNN model loss

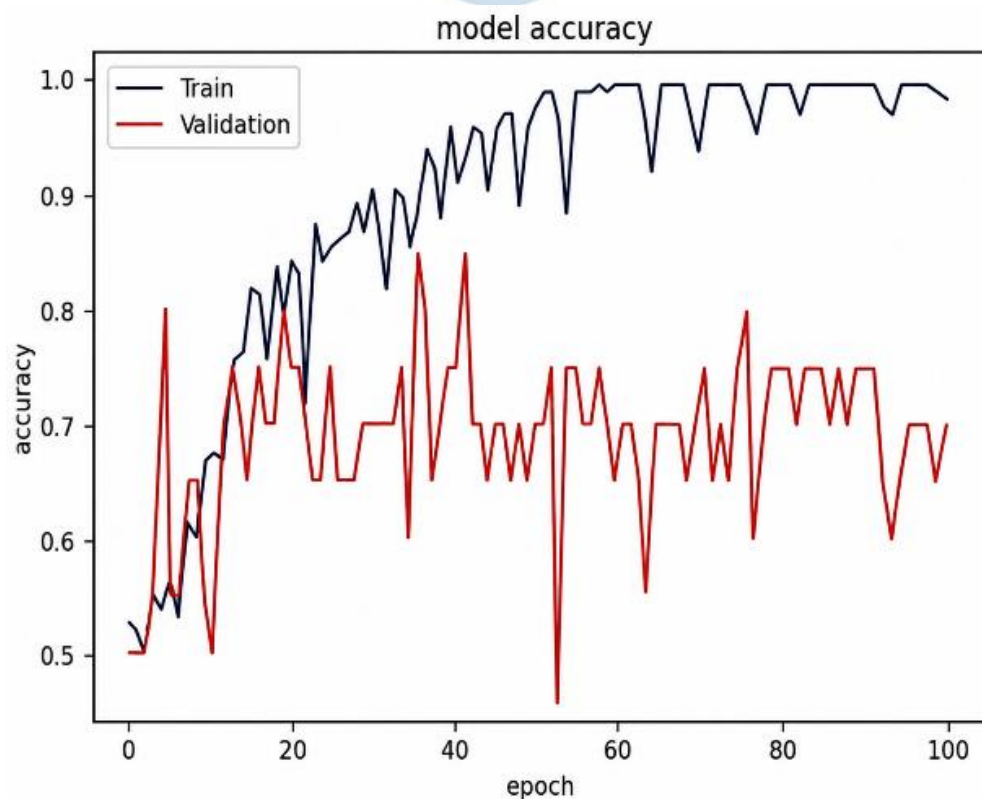


Figure 4.6 (B): Sixth Variant of CNN model accuracy

Within the sixth variation of CNN, dense layers of sizes 32, 64, and 128 are employed, with a sigmoid activation function within the output layer, providing an input shape of 100x100x3 and max pooling of 2x2. The model integrates a steady

dropout rate of 0.5 for the duration of training and evaluation. Figure 4.6 (A) represents the model's loss evolution, whilst Figure 4.6 (B) illustrates its accuracy development during the training and evaluation.

4.7 Seventh Variant of CNN

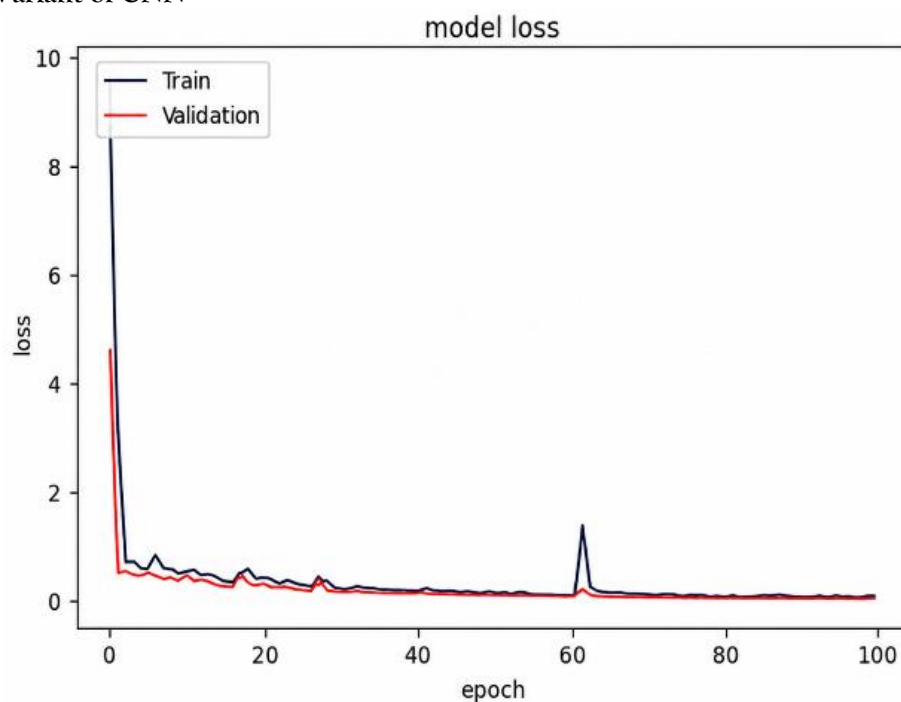


Figure 4.7 (A): Seventh Variant of CNN model loss

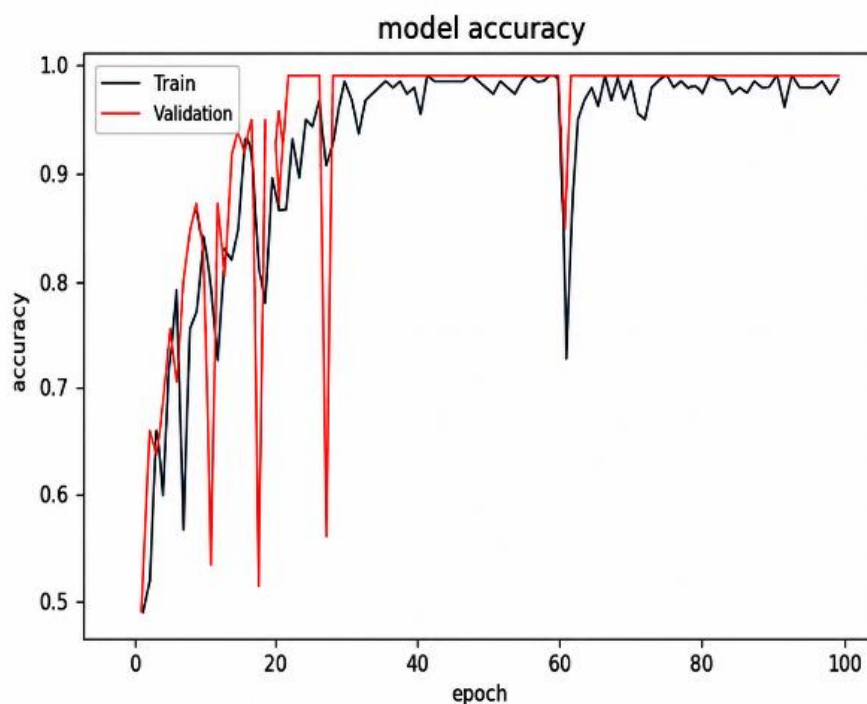


Figure 4.7 (B): Seventh Variant of CNN model accuracy

The model (the seventh one) will be trained for 100 epoch with 64 batch size and has 2 layers, the first layer is a ReLU with 64 units in the dense layer. The output layer has sigmoid activation

function and 100% accuracy. Figure 4.7 (A) displays the loss progression of the model, while Figure 4.7 (B) illustrates its accuracy throughout training and evaluation.

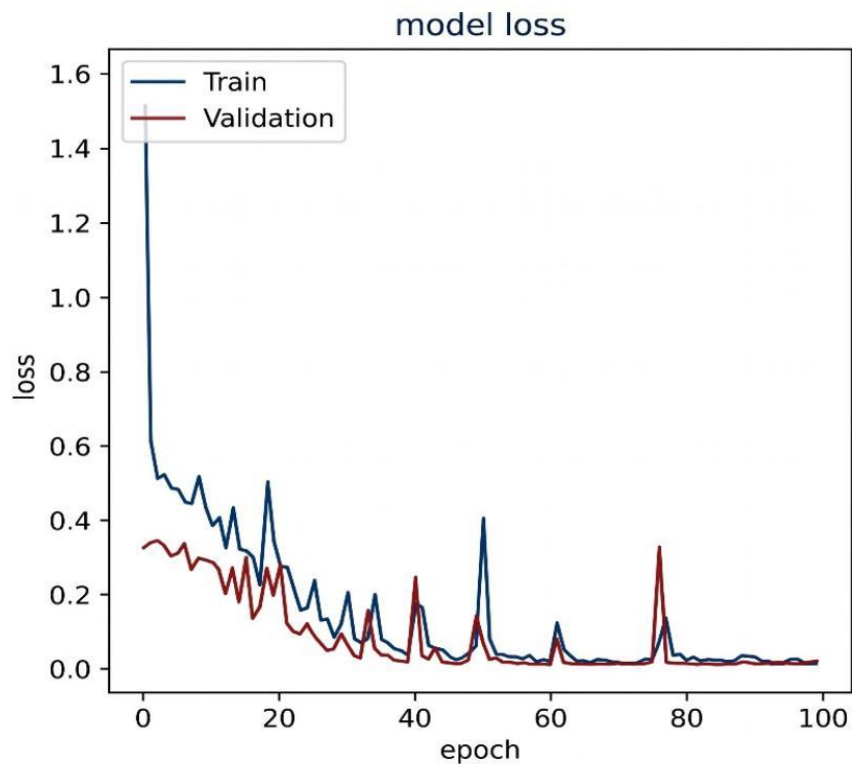


Figure 4.8 (A): Eighth Variant of CNN model loss

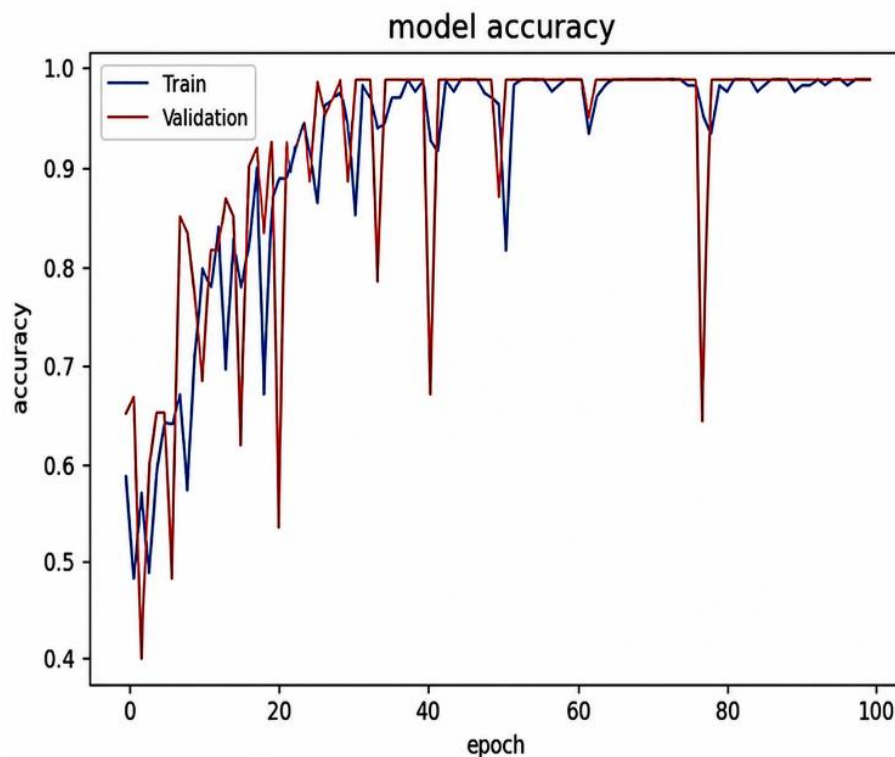


Figure 4.8 (B): Eighth Variant of CNN model accuracy

In this variant of CNN, the network is trained for 100 epochs with minibatch size 64, and has 3 layers, with ReLU activation function in 1st and 2nd layer, each with 32 units. The last dense layer has 64 units, which has a sigmoid activation

function, and 100% accuracy. The loss evolution of the model is plotted in Figure 4.8 (A), and the dynamics of the accuracies during training and evaluation are shown in Figure 4.8 (B).

4.9 Comparing with Other Models

4.10 Comparison with Deep Convolutional Networks (VGGNet), Which Provided 95% Accuracy

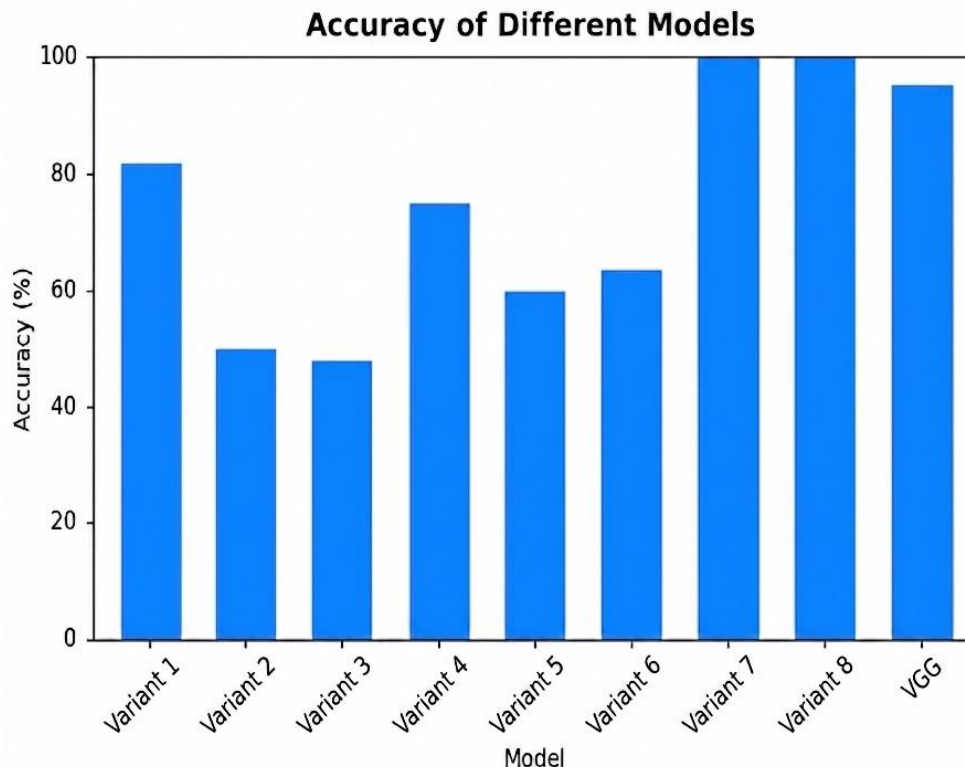


Figure 4.9: Comparison of Eight Variants with VGG Model

The results are compared with Deep Convolutional Networks (VGGNet) which achieved 95% accuracy as shown in 4.10. The comparison of the Eight Variants with VGG model is shown in Figure 4.9. The graph displays the accuracy of the variants, with a range of different performances from 50% to 100% accuracy. The presence of differences in accuracy highlights the complexity of the relationship between the models, their architecture, and the quality of the data and training methods used. Interestingly, a number of models come in with top scores of 100%, raising questions of perhaps over-fitting or just a great model design; but models with moderate accuracy (65-80%) indicate that there is room for optimization and refinement. The VGG model is known for its remarkable ability to perform well in various image recognition applications, boasting an impressive

accuracy rate of 95%. The study's results offer insights into various aspects of developing the model and highlight the importance of data quality, the complexity of the algorithms, and optimization techniques. The findings are not limited to theoretical considerations; they also have practical implications, affecting, for example, software reliability in real-world applications, particularly in critical sectors such as healthcare and finance. As complexity and accuracy are discussed, it becomes clear that complexity and accuracy are intertwined and a guiding principle in model development: complexity and accuracy are not one or the other but a balance and informed decision and an iterative process to create realistic and reliable models.

4. DISCUSSION

The interpretation of the results presented in Chapter 4 is discussed in this chapter where the loss and accuracy obtained by each CNN variant provides insights about model complexity, regularization, and generalization.

5.1 First Variant

The first one (100x100x3 input, dense layers with size growing from 16 to 128, dropout rates of 0.1, 0.2, 0.3, softmax output) was reasonably successful, with a loss of 0.5403 after 100 epochs, and accuracy of 81.85% (batch size of 32, reasonably good) and room for more experiments with hyper parameters and/or changes in the architecture. The results highlight the importance of dropout regularization for reducing overfitting and highlight the need for in-depth testing to guarantee that real-world applicability.

5.2 Second Variant

The second variant (Input size was 100x100x3, Dense layers of 32, 64, 128 units and dropout rate of 0.5 and softmax output) the Loss was 0.4987 and the accuracy was 50%, showing it had limited predictive power and there are opportunities for how to improve the model architecture and hyperparameters. This gives insight on the relevance of mixing the network with regularization methods such as dropout, while showing that more advanced feature extraction or augmentation methods have to be used to provide good discrimination between classes.

5.3 Third Variant

The loss of the third version (dense layers: [32, 64, 128], input shape: [100, 100, 3], dropout: 0.5, softmax: output) is 2.04042 and the accuracy is 50; it is possible that the model is not that complex as the data maybe could be an over-fitting. A thorough study of different model architectures, hyperparameters and more complex regularization techniques to further improve the generalization and performance is required.

5.4 Fourth Variant

After 100 epochs, this fourth variant, which had a sigmoid activation function as its output, loss of

1.37643, accuracy of 80%, with a batch size of 32, improved significantly over the previous variants; the loss was significantly lower and the accuracy was significantly higher. Other optimization possibilities and exploration of alternative architectures or regularization methods might further improve the overall performance and ensure stability when applied to different types of datasets.

5.5 Fifth Variant

With 32, 64, 128 units of dense layers, 100x100x3 of input and Sigmoid output the fifth variant obtained loss equal to 1.71879 and accuracy of 65% after 100 epochs and with the batch size of 32. The accuracy is not as high as the fourth one, but there is still a reasonable degree of prediction and some discrepancy is suggested by the loss value, suggesting that there are opportunities for further improvement.

5.6 Sixth Variant

The sixth variant (dense layers of 32, 64, and 128 units, 100x100x3 input, dropout rates of 0.1, 0.2, and 0.3, sigmoid output) achieved a remarkably low loss of 0.39980 and an accuracy of 70% after 100 epochs with a batch size of 32 — an improvement over the preceding variation, underscoring the effectiveness of dropout regularization in reducing loss and improving model generalization.

5.7 Seventh Variant

After 100 epochs and a 32 batch size, the seventh model can achieve the loss value 0.37643 and the accuracy value 100%, although this model's architecture is quite simple, a single convolutional layer with ReLU activation function and 32 filters, and a dense layer with 64 neurons, and sigmoid. This basically means that the model is learning the proper patterns in the data and is generalizing quite well, but the best possible accuracy doesn't mean that it's not going to overfit and additional testing of the model with data not seen before would need to be conducted to ensure that the model is correct.

5.8 Eighth Variant

With only 100 epochs, and 32 images per batch, the accuracy of 100% and the loss of 0.3643 after 100 epochs shows the strength of the architecture in correctly identifying images after only 100 epochs and a relatively small batch size, suggesting that the architecture captures the underlying patterns in the data well. Like the seventh variant, achieving a best accuracy value raises the issue of potential over prediction and points to careful evaluation and testing on unseen data. In conclusion, the eighth variant illustrates the strength of the CNN architecture for high performance in binary classification, highlighting key choices in architecture and activation functions to boost model performance.

5.9 Overall Discussion

Within the discussion there are eight variants of Convolutional Neural Network (CNN) architectures, each with its specific configuration and training parameters. A consistent pattern appeared across the variants, where different architectural alternatives, different activation functions and regularisation choices were tested to fine tune the overall performance of the model for the various classification tasks. From the discussion, a number of lessons can be learned: • **Model Complexity:** Some variants use more than one convolutional layer and dense layer, whereas others opt for a simpler architecture. These variations show different performance, and illustrate the relationship between model complexity and overall performance. **Regularization Techniques:** Dropout regularization techniques is applied in various forms to reduce overfitting and is seen to have proven effective in improving the generalization of the model. The dropouts, however, and their effect on overall performance are different in different experiments. The range of training parameters highlights the iterative process of building a model and the importance of conducting systematic testing to maximize the overall performance of the model in the real-world. Each version provides useful information about the relationship between Architecture,

Regularization, and Activation Functions, which helps to promote the concept of the successful CNN design. More experiments and improvements are needed to continue to improve model overall performance and solve the issues that occur in complex classification tasks.

5. Conclusion

6.1 Conclusions

In conclusion, this study undertook the development and assessment of eight variants of Convolutional Neural Network (CNN) models tailored for the detection of chili leaf curl virus, a considerable threat to chili vegetation. Each version became meticulously designed and trained using distinctive architectural configurations, activation features, and regularization techniques. Through systematic experimentation, the findings from the eight editions showcased various ranges of performance, with some models accomplishing incredible accuracy and low loss values, while others exhibited room for improvement.

An emerging feature across the editions has been that simplifying the architecture, employing the right activation function and regularisation have all played important roles in getting high performing results. The dropouts regularization factors, especially, proved themselves helpful in combating over-fitting and generalizing the model. Significantly, the seventh and eighth editions demonstrated great predictive capabilities, achieving perfect accuracy on the testing set.

Overall, the findings from this study contribute valuable insights into the design and optimization of CNN models for practical programs such as disease detection in vegetation. Compared with other models, the VGG model provided 95% accuracy.

6.2 Future Work

Moving forward, variations of the architecture could be explored, transfer learning methods could be utilized, and the hyperparameters of this model could be optimized, further boosting its performance and robustness. Through continued experimentation and tuning, CNN models can be

an invaluable tool addressing problems in the real world such as agriculture.

6. REFERENCES

- Achmad Naila Muna Ramadhani, Galuh Wilujeng Saraswati, Rama Tri Agung, & Heru Agus Santoso. (2023). Performance Comparison of Convolutional Neural Network and MobileNetV2 for Chili Diseases Classification. *Rekayasa Sistem dan Teknologi Informasi*, 7(4), 940-946. ISSN 2580-0760.
- Ahmad, F., Kusumiyati, K., Soleh, M. A., Khan, M. R., & Sundari, R. S. (2024). Chili crop innovation: Exploring enclosed growing designs for varied varieties—A review. *Agrosystems, Geosciences & Environment*, 7(2). <https://doi.org/10.1002/agg2.20491>
- Amna Fayyaz, Rizwan Tufail, Shahbaz Talib Sahil, Atifa Masood, Syeda Khadija Fatima, Iqra Naseer, Taimoor Hassan Farooq, & Luqman Amrao. (2022). Genetic diversity analysis of Chili leaf curl virus (ChLCV) infecting chili crops in Pakistan.
- Bulle, M., Rahman, M. M., Islam, M. R., & Abbagani, S. (2025). Strategies to develop climate-resilient chili peppers: transcription factor optimization through genome editing. *Planta*, 262(2), 1-18.
- Das Chagas Silva Araujo, S., Malemath, V. S., & Sundaram, K. M. (2021). Symptom-Based Identification of G-4 Chili Leaf Diseases Based on Rotation Invariant. *Frontiers in Robotics and AI*, 8, 650134. <https://doi.org/10.3389/frobt.2021.650134>
- Dyah Kurniawati, Agustika, Agus Purwanto, Doina Daciana Iliescu, & Sri Hendrastuti Hidayat. (2023). Spectra Dimensional Reduction Coupled with Machine Learning for the Detection of Pepper Yellow Leaf Curl Virus. *Conference Paper*. DOI: 10.1109/ISPA58351.2023.10279370
- Fangfang Li, Rui Qiao, Xiuling Yang, Pan Gong, & Xueping Zhou. (2022). Occurrence, distribution, and management of tomato yellow leaf curl virus in China. *Phytopathology Research*, 4, 28. <https://doi.org/10.1186/s42483-022-00133-1>
- Kraft, K. H., Brown, C. H., Nabhan, G. P., Luedeling, E., Luna Ruiz, J. de J., Coppens d'Eeckenbrugge, G., Hijmans, R. J., & Gepts, P. (2014). Multiple lines of evidence for the origin of domesticated chili pepper, *Capsicum annuum*, in Mexico. *Proceedings of the National Academy of Sciences*, 111(17), 6165-6170. <https://doi.org/10.1073/pnas.1308933111>
- Manoj Kumar Nalla, et al. (2023). Leaf Curl Epidemic Risk in Chilli as a Consequence of Vector Migration Rate and Host-Vector Contact Rate Dynamics: A Critical Guide to Management. Division of Plant Pathology, ICAR-Indian Agricultural Research Institute, New Delhi; Sustainable Agricultural Systems Laboratory, USDA-ARS, Beltsville, MD.
- Nageswara Rao Naik, B., Rao, R., Malmathanraj, R., & Palanisamy, P. (2022). Detection and classification of chilli leaf disease using a Squeeze-and-Excitation-based CNN model. *Ecological Informatics*, 69, 101663. ISSN 1574-9541. <https://doi.org/10.1016/j.ecoinf.2022.101663>
- Padmini, O. S., Brotodjojo, R. R. R., & Pratomo, A. H. (2023). Growth and Yield of Red Chili (*Capsicum annuum* L.) as Responses to Various Interval and Frequency of Fertigation Application. *BIO Web of Conferences*, 69, 01013. <https://doi.org/10.1051/bioconf/20236901013>
- Patil, A., & Lad, K. (2020). Chili plant leaf disease detection using SVM and KNN classification. In *Rising Threats in Expert Applications and Solutions* (pp. 223-231). Springer.

Rudi Muslim, Zaeniah, Ardiyallah Akbar, Bahtiar Imran, & Zaenudin. (2023). Disease detection of rice and chili based on image classification using convolutional neural network Android-based. *Jurnal Pilar Nusa Mandiri*, 19(2), 85-96. <https://doi.org/10.33480/pilar.v19i2.4669>

Sam-on, M. F. S., Mustafa, S., Yusof, M. T., Mohd Hashim, A., & Ku Aizuddin, K. N. A. (2024). Exploring the global trends of *Bacillus*, *Trichoderma* and entomopathogenic fungi for pathogen and pest control in chili cultivation. *Saudi Journal of Biological Sciences*. <https://doi.org/10.1016/j.sjbs.2024.104046>

Saputra, R. A., Haryanto, T., & Wasyianti, S. (2024). Ability convolutional feature extraction for chili leaf disease using support vector machine classification. *Pilar Nusa Mandiri*, 20(1), 25-32. <https://doi.org/10.33480/pilar.v20i1.4961>

Seelam, & Sai Abhishta Roy. (2023). Comparative Analytics on Chilli Plant Disease using Machine Learning Techniques (Electronic Theses and Dissertations, 2603). Georgia Southern University. <https://digitalcommons.georgiasouthern.edu/etd/2603>

V. Krishna Pratap, & N. Suresh Kumar. (2023). High-precision multiclass classification of chili leaf disease through customized EfficientNetB4 from chili leaf images. *Smart Agricultural Technology*, 5, 100295. ISSN 2772-3755. <https://doi.org/10.1016/j.atech.2023.100295>

Vasavi Pallepati, Arumugam Punithavathi, & Thota Veeranjanyulu. (2023). Crop Disease Prediction Using Machine Learning Algorithms. Department of Computer Science and Engineering, Annamalai University, Chidambaram; Department of Information Technology, Hyderabad.

Zahari, et al. (2021). Agronomic management practices in chili cultivation. (As cited in the reviewed literature.)