

Identification of Plant Diseases in Jordan Using Convolutional Neural Networks

Moy'awiah A. Al-Shannaq

moyawiah.s@yu.edu.jo

Yarmouk university

Shahed AL-Khateeb

Yarmouk university

Abed Al-Raouf k. Bsoul

Yarmouk university

Ahmad A. Saifan

Yarmouk university

Article

Keywords: Computer Vision, Convolutional Neural Nets, Learning (Artificial Intelligence), Diseases, Image Classification, Image Processing

Posted Date: October 16th, 2024

DOI: <https://doi.org/10.21203/rs.3.rs-4900445/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Abstract

In the realm of global food security, plants serve as the primary source of sustenance. However, plant diseases pose a significant threat to this security. The process of diagnosing these diseases forms the bedrock of disease control efforts. The precision and expediency of these diagnoses wield substantial influence over disease management and the consequent reduction of economic losses. Conversely, incorrect diagnoses can render interventions ineffective, leading to agricultural crop deterioration and compounding economic hardships for both farmers and their respective nations. This research endeavors to diagnose the prevalent crops in Jordan, as identified by the Jordanian Department of Statistics for the year 2019. These crops encompass four key agricultural varieties: cucumbers, tomatoes, lettuce, and cabbage. To facilitate this, a novel dataset known as "Jordan 22" was meticulously curated. Jordan 22 was painstakingly compiled through the collection of images featuring both diseased and healthy plants, captured within the confines of Jordanian farms. These images underwent meticulous classification by a panel of three agricultural specialists, well-versed in plant disease identification and prevention. The Jordan 22 dataset comprises a substantial size, amounting to 3210 images. Following the compilation of this dataset, a series of preprocessing steps were executed. These encompassed the standardization of image backgrounds and the uniformization of image dimensions. Furthermore, image augmentation techniques were applied to the dataset to expand its diversity. Subsequently, a deep learning model, the Convolutional Neural Network (CNN), was meticulously trained on the augmented dataset. The results yielded by the CNN were nothing short of remarkable, with a test accuracy rate reaching an impressive 0.9712. Optimal performance was observed when images were resized to 256x256 dimensions, and max pooling was employed in lieu of average pooling within the pooling layer. Furthermore, the initial convolutional layer was set at a size of 32, with subsequent convolutional layers standardized at 128 in size. In conclusion, this research represents a pivotal step towards enhancing plant disease diagnosis and, by extension, global food security. Through the creation of the Jordan 22 dataset and the meticulous training of a CNN model, we have achieved substantial accuracy in disease detection, paving the way for more effective disease management strategies in agriculture.

1. INTRODUCTION

Farmers and agricultural experts can discover plant diseases through the naked eye. The process of diagnosing plant diseases through the naked eye was characterized as inaccurate, and there is a need for an expert in the field of plant diseases; because of the different diseases and the similarity of the different symptoms of diseases from different plants. Plant leaves are the basis for recognizing plant disease; most disease symptoms may begin to appear on the leaves. Agriculturalists with less expertise may face difficulties assessing the infection that has infected the plant, affecting quality and yield [1]. In 1852, Ireland faced a great famine, often known as the Irish potato famine, which cost over a million Irish lives. Result of the spread of late blight on potato plants resulted in the loss of 70% of agricultural production. Despite the world's attempts to control pests with 11 pesticides, pathogens have led to a decline in agricultural production and a decrease in its quality [2].

Plant diseases are responsible for crop damage due to the rapid spread of diseases among agricultural crops, which require early and continuous diagnosis and appropriate treatment in a timely way. Misdiagnosis causes massive losses for farmers and decreased crop yields, which leads to higher pricing and economic instability. Accurate disease identification is required in order to provide disease information. Incorrect plant disease diagnosis leads to the use of inappropriate pesticides, the use of pesticides on agricultural crops leads to long-term resistance to plant diseases and the inability to manage disease spread among agricultural crops, resulting in economic losses and threatening global food security. The economy of developing countries depends primarily on agricultural productivity. Reduced agricultural production losses will help to accomplish one of the Sustainable Development Goals (SDG) of eradicating hunger and protecting water [3].

Effective detection technology will assist in decreasing agricultural economic losses related to incorrect diagnosis that leads to the destruction of crops due to the rapid spread of disease among plants. There are numerous efforts by the scientific research community in the agricultural field to use Artificial Intelligence (AI) in diagnosing plant diseases through Deep Learning (DL)[4] and Machine Learning (ML) [5] to identify diseases in various plants around the world. AI is used in different topics, and researchers around the world are becoming increasingly interested in neural networks (ANN), especially in the fields of recognition, prediction, and big data. Intelligent systems have been widely used in computer vision and many other applications and have a relatively competitive advantage.

The four core areas of AI are ANN, fuzzy inference systems, expert systems, and evolutionary computing. ANN can perceive human behavior [6]. Image classification is a difficult task. Multiple DL techniques, such as ANN, CNN, Recurrent Neural Network (RNN), Generative Adversarial Network (GAN), Long Short-Term Memory Networks (LSTMs), Radial Basis Function Networks (RBFNs), and Multilayer Perceptron (MLP), are used to identify various types of plant diseases. AI is a burgeoning revolution in agriculture. AI improves agricultural yield, production, and quality through real-time monitoring and harvesting [7].

There are many contributions to solving the problems of agriculture with AI, such as the use of AI in diagnosing plant diseases. Many researchers have used different models of AI to discover the diseases of plants, but there are some challenges, and a group of studies has suffered from a set of problems as some studies suffer from the use of a dataset of misclassified images, which leads to misinformation [8].

The common dataset used by researchers was called PlantVillage[9], an open-source database for researchers around the world that is used in the field of identifying plant diseases. The researchers used PlantDoc [7]and PlantVillage [9]. This work aimed to contribute to building a dataset containing images of healthy and diseased plant leaves of tomatoes, cucumbers, cabbage, and lettuce. A high-quality smartphone camera captured the images of plant leaves, and experts in the field of plant diseases classified the images. We used DL, a Convolutional Neural Network (CNN) that has proven helpful in the field of computer vision, to identify diseases that affect crops cultivated in Jordan. The dataset was

collected manually by capturing infected and healthy plants; agricultural experts classified these images. The next phase was preprocessing the dataset and, finally, training the CNN model on the dataset and getting results. The importance of the results of the study lies in obtaining the best CNN model that does not suffer from the problem of overfitting and has high accuracy and a low rate.

2. RELATED WORK

Some scientific studies in the field of plant disease classification using ML and DL algorithms were presented in this section. Previous research was classified into two categories: ML and DL.

2.1 Deep Learning in Plant Disease

The DL uses ANN designed to mimic the way humans think. DL aims to find features without the need for humans. DL needs more powerful hardware than ML. The one kind of hardware operated by DL is Graphical Processing Units (GPUs). In 2022, [10] For training the internal damage segmentation network (IDSNet), an attention-based generative adversarial network (AGAN) was created to produce artificial pictures. The F1-score and sensitivity provided by the IDSNet over a test set are both 0.941. In 2021, to support applications for precision plant protection and computer vision applications in agriculture [9], a CNN was used to discover 38 different plant diseases, and the CNN provided an accuracy of 88%. In the study [11], they proposed to classify citrus diseases using transfer learning. They used the following models: InceptionV3, ResNet50, VGG16, and VGG19, which are pre-trained models on a large dataset (ImageNet). Data augmentation was used to obtain more reasonable classification accuracy. The results indicate that VGG19 achieved the greatest level of training accuracy at 99.89%. This study[12] proposes to identify plant diseases by utilizing VGG16 and Resnet34. The model can identify 38 different kinds of plant diseases. The accuracy provided by Renet34 was 97.77, and VGG16 was 97.58. In this study,[8] proposed a CNN model to detect seven plant diseases, where the dataset contains 8685 leaf images from the PlantVillage dataset. The performance metrics that were used (F1 score, accuracy) are: accuracy is 996.2%, and the F1 score is greater than 95%. This study[13] suggested the accurate detection of pests or diseases that have attacked the Kenaf plant. Kenaf fiber was mostly utilized in alternative industrial goods made from forest wood. They used a VGGNet model on a Kenaf dataset that includes 838 images, providing an average accuracy of 73%. Table 1 shows a set of techniques that have used deep learning to identify plant diseases

Table 1
Previous studies in identifying plant diseases using different DL techniques.

References	Dataset	Classifier	Accuracy
[9]	PlantVillage	CNN	88%
[8]	PlantVillage	CNN	96.2%
[12]	Kaggle database	Resnet34	97.77%
[14]	Plant Pathology 2020 FGVC7 dataset	DLCNN	99.13%
[15]	Rice leaf diseases dataset GitHub	CNN	51.2%
[16]	PlantVillage	ResNet50	98.93%

2.2 Machine Learning in Plant Disease

Based on the data type, ML requires defining and hand-coding the applied features. ML can run on low-end devices with less computational power.

In 2021, [16]proposed the use of a decision tree model, and a current generic method for wheat disease classification has been developed. Domain specialists enhanced decision tree accuracy by 28.5%. The decision tree provides an accuracy of 94.7%. In this study [17], the images of the plant's leaf are in RGB format. To increase the quality of the image, preprocessing was used to eliminate any type of noise. The Gray Level Concurrence Matrix (GLCM) was used to extract characteristics that are utilized to determine the spatial gray level difference in an image. Contrast, correlation, energy, and homogeneity were some image properties retrieved using GLCM. The dataset used for training was obtained from the Internet and included images of plants infected with various diseases as well as healthy leaves. K-means clustering was used to cluster objects so that the objects inside the cluster have similar properties. The Euclidean distance was used to calculate the distance.

The classifier used for classifying SVM In 2020, [18]proposed the use of a random forest classifier for the detection of plant disease on PlantVillage, a publicly available dataset. The dataset was separated into a training set of 80% for model training and a test set of 20% for verification. To improve the degree of accuracy, a K-fold cross- validation approach was used. The RGB image was converted to grayscale before being smoothed with a Gaussian filter. After extracting the color characteristics from the image, textures such as contrast dissimilarity, homogeneity, energy, and correlation were retrieved from the GLCM. The random forest classifier has a 93% average accuracy and an F1 score of 0.93.

In 2019,[19] proposed to detect the three most common diseases that occur in the sugarcane plant in India: red rot, mosaic, and leaf burn. A combination of extracted features such as color, size, and shape. Clustering was performed using a k-means cluster algorithm. Based on this, the image was segmented into different regions, including diseased parts and non-diseased parts. GLCM was applied to the images to get features. The Multi-SVM classifier is used to classify images by type of disease. Table 2 below shows, a different study in the field of identifying different plant diseases using multiple ML models.

Table 2
Previous studies in identifying plant diseases using different ML techniques.

References	Dataset	Classifier	Accuracy
[16]	Surveys, interviews, and field experts' responses.	Decision Trees	94.7%
[5]	Data were from internet.	K-Means, SVM	Not specified
[20]	PlantVillage	Random Forest	93%
[21]	PlantVillage	Random Forest	95.2%

2.3 Discussing Previous Studies

The model's performance deteriorated when the dataset was imbalanced and the number of data samples was small. For the DL models on plant disease detection, increasing the number of data samples in the dataset was required to increase model performance. When the number of images in the dataset was not large enough for training, the data augmentation approach can be used to satisfy the requirements. In the ML algorithm, SVM was a better choice for disease detection [7], [19]. A feature extraction technique was required for all deployed ML algorithms. This feature extraction method was not required for DL models [22]. In DL models, in this study [13], the size of the dataset was 838 images. The data augmentation technique was not used to increase the size of the dataset. Here, VGGNet was used. The training accuracy reached 0.9375 and the validation accuracy reached 0.4889 at 400 epochs. We should point out that the performance was poor since the dataset was small and the data augmentation technique was not applied, even if the dropout layer exists. In this study[15] a CNN model was used to identify three diseases of rice. The average training accuracy of 91.41% and the average test accuracy reached 51.19%. Here, the dataset contains 2239 leaf images, the data augmentation technique was not used, and the dropout layer was not used. This study[11] suffered from the problem of overfitting where the accuracy of the validation reached 0.8268 and the accuracy of the training reached 0.9989 when using a VGG19 model on ImageNet dataset where the number of images was fewer than 1000 images. This study suffered from an overfitting issue. In this, study[23] despite the use of a large dataset for the classification of plant diseases, PlantVillage was used, and the data augmentation technique was applied to PlantVillage, which contains images of both healthy and diseased plants, demonstrates the importance of data augmentation techniques in the field of identifying plant diseases using DL techniques. This study[8] suffered from a very limited dataset. This study[13] suffered from the use of a dataset of incorrectly misclassified images, which caused a decrease in the accuracy of the model. The CNN was considered the latest technology in image recognition systems and was able to provide a fast and accurate diagnosis [7]. The DL models outperformed the ML models in classification accuracy, according to previous studies in the same area. In this research, we applied a CNN model to the newly collected dataset. The dataset utilized in previous studies was not used in this study. To prevent the errors that past studies have made, all images of plants and their diseases were collected manually with the support of specialists in the field of diagnosing plant disease, and it was important to

increase the amount of data by using data augmentation. To solve the problem of overfitting, we used a dropout layer in CNN and compared the performance with other studies.

3. RESEARCH METHODOLOGY

In this section, we demonstrate the dataset used and the technique used to identify plant diseases.

3.1 Overall Research Design

shows the main stages of the study. The first stage loads the dataset containing RGB images of plants; the second stage performs image preprocessing, then applies the | model for training and testing, and finally obtains the performance of the model. FIGURE 1 shows the main stages of the study. The first stage loads the dataset containing Red, Green and Blue (RGB) images of plants; the second stage performs image preprocessing, then applies the CNN model for training and testing, and finally obtains the performance of the model.

The first stage, as seen in FIGURE 1 is the collection and construction of a dataset including images of Jordan's most farmed plants for the year 2019, which include tomatoes, lettuce, cabbage, and cucumbers. The dataset includes images of both diseased and healthy plants. Three agricultural specialists with expertise in plant disease identification classified the images. The next stage was image preprocessing by standardizing the size of the images and increasing the size of the dataset by using data augmentation in the dataset. Finally, use the CNN model to train and test the dataset to evaluate the accuracy of the CNN model.

3.2 Research Phases

3.2.1 Data Collection

Our dataset was collected manually by capturing it with a smartphone camera. The gathered dataset was named Jordan22 [17]. It contains a set of images of unhealthy and healthy plants that were classified by experts in the field of plant diseases and prevention. The experts who reviewed and classified the dataset were the first expert¹ with experience in the field of plant diseases for more than 20 years, and the second and third experts² from the National Agricultural Research Center (NARC) specialized in the field of detecting plant diseases from the department of plant disease prevention at the NARC. These experts have classified the images of unhealthy and healthy plants from the Jordan22 [17]dataset, which was collected from different regions in Jordan, including Ramtha, Mafrqa, and Mansheya. The TABLE 3 shows the specifications of the image database that was created.

Table 3
Specifications of the images on the Jordan22 dataset[17].

The types of plants	Cucumber, tomato, cabbage, and lettuce
Size of dataset	2310 RGB
The stages of plants	The vegetative stage, the flowering stage, and the fruiting stage
Device name	iPhone 12 Pro Max
The agricultural areas	The agricultural areas
Camera specifications	12-megapixel resolution
Type of image format	Joint Photographic Group (JPG)
Disease type of cucumber	Downy mildew
Disease type of tomato	Early blight
Disease type of cabbage	Downy mildew
Disease type of lettuce	Leaf miners

In this research, images of diseased and healthy plants were collected manually. In FIGURE 2 the blight disease on a tomato crop in different datasets used by researchers in the field of identifying plant disease using AI, the right image is from the PlantVillage dataset [24], the mid image is from the Jordan22 dataset, and the left image is from the PlantDoc dataset [25]. As shown in the FIGURE 2, we can see tomato blight on different datasets.

Some images in the PlantDoc dataset were incorrectly classified due to a lack of in-depth expertise in plant diseases [25]. The dataset of images in the Jordan22 dataset is captured by a high- resolution smartphone camera on a group of farms in Jordan. According to the Department of Statistics (DOS) 2019, cucumbers are the most productive in Jordan, tomatoes, cabbage, and lettuce [26]. We used the CNN model to detect plant disease in the Jordan22 dataset. The dataset comprises 2310 in red, green, and blue values; they are known as RGB images of healthy and unhealthy plant leaves with 8 classes. There are 1278 images of diseased plants and 1032 images of healthy plants. Table 4, details the names of the plants, the type of disease that affected them, and the number of images taken of the diseased plants and healthy plants [17]. FIGURE 3 shows a group of images taken from the Jordan22 dataset [17]. The database was divided into 70% for training, 15% for validation, and 15% for testing as described in Table 5.

Table 4
Dataset Specifications [17].

Plants	Class Number	Category Name	Number of Image
Cucumber	C1	Healthy	264
	C2	Downy mildew	291
Tomato	C3	Healthy	385
	C4	Blight disease	516
Cabbage	C5	Healthy	282
	C6	Downy mildew	386
Lettuce	C7	Healthy	101
	C8	Leaf miners	85

Table 5
Test, train, and validate the number of datasets.

Number of train dataset	1618
Number of validation dataset	346
Number of test dataset	346
Total	2310

3.2.2 Preprocessing

• Unify the Backgrounds

The objective of unifying all image backgrounds is to minimize the noise in the dataset images. All image backgrounds are white; the CNN model can target plants without being distracted by other factors in the background. FIGURE 4 shows an image from the Jordan22 dataset where the background is white.

• Standard Image Size

The images were reconstructed in square ratios ($m \times n$), which were required for inclusion in the CNN model. The dimensions of this image are $m \times n \times 3$, where the width is m , the height is n , and the depth is three. Images resized to $m \times n$ pixels' resolution to reduce computational time. The images in Jordan22 dataset are in RGB format.

• Data Augmentation

Data augmentation is a critical stage in image classification in DL. Data augmentation applies various alterations to original images, resulting in many altered replicas of the same image. Each copy is distinct from the other in certain aspects relying on the augmentation techniques including, shifting, rotating, flipping, etc. Involving these little alterations in the original image has no effect on the target class, but rather feeds an unknown vision in real life. These data augmentation approaches can increase the quantity of the dataset in addition to incorporating a degree of interpretation, allowing the model to generalize better using unknown input. Furthermore, when trained on previously undiscovered, slightly changed images, the model becomes more robust. The ImageDataGenerator class guarantees that the model obtains the most recent image variations every epoch [27]. The training with ImageDataGenerator by applying random transformations to each image in the batch and replacing the original batch of images with a new randomly transformed batch [27]. Table 6 Shows the image augmentation techniques used for the Jordan22 dataset.

Table 6
ImageDataGenerator Args
and values that used.

Args	Value
shear_range	0.2
zoom_range	[0.5,1.0]
horizontal_flip	True
fill_mode	Nearest
rotation_range	25

The ImageDataGenerator class was designed to provide real-time data augmentation, while the model was still in the training phase. FIGURE 5 shows the use of data augmentation that adds multiple new images that did not previously exist in the dataset, consequently improving model performance.

3.2.3 Using Deep Learning Model

The phrase "deep learning" in a deep neural network refers to the number of layers in a neural network. In this section, we showed the CNN model structures.

- **Convolutional Neural Network (CNN)**

CNN is a feedforward neural network that is very effective in image processing. CNN includes three layers a convolutional layer, a pooling layer, and a fully connected layer. CNN is a popular place for scientific study in many domains, particularly pattern recognition. Because of its ability to input original images without requiring complex preprocessing.

FIGURE 6 demonstrates that CNN processes an image as an input and produces a distribution of class scores, which can be used to determine the most likely class for a given image.

The color images were transformed into three-dimensional value cubes with width, height, and depth. The number of colors used determined the depth. Most color images were represented by three colors: red, green, and blue values; these images are known as RGB images, and they have a depth of three. FIGURE 7 depicts Cucumber disease in RGB layers.

The first layer of the CNN network analyzes the incoming image. The convolutional layer is composed of a series of convolutional filters. Each filter extracts a different type of characteristic. The convolutional layers produce a series of feature maps (also known as activation maps), which are filtered representations of an original input image.

The filters are often smaller than the actual image. Each filter interacts with the image to generate an activation map. The filter was slid over the height and width of the image for convolution [28]. The weights of the filter help to extract meaningful information from the image the convolution includes a set of learnable filters in the form of a matrix (width, height, and depth). The convolutional layer is in charge of identifying patterns. A large number of filters are required to detect a pattern in an image in a complex dataset with many different object types, which indicates convolutional layer dimensionality can get large. Higher dimensionality necessitates the usage of additional parameters, which might result in overfitting. Therefore, we need a way to minimize this dimensionality. This is the role of CNN's pooling layers. As seen in FIGURE 8, the convolution layers input is on the left side. Feature extraction convolution filter can be seen in the middle sub-figure. The convoluted image result is shown on the right side of the figure.

The Max pooling layers examine regions in an input image and choose to maintain the highest pixel value in that area, therefore, resulting in a reduced-size area. Feature extraction aims to minimize the number of features. Feature extraction was made up of multiple pairs of convolutional and pooling layers. The pooling layer generates new features that summarize the new features from an initial set of features. The most common is max pooling, which computes the maximum value for each feature map patch. Average Pooling saves a lot of information about the "less essential" parts of a block or pool. Whereas max pooling simply discards them by selecting the highest value. FIGURE 9 depicts a basic max pooling layer and the average pooling layer utilizing a 2x2 sliding window. FIGURE 9 illustrates the difference between max pooling and average pooling, with max average taking the largest number, while average takes the average[29].

The features extracted on the convolutional layers and pooling layers were straightened and placed at the fully connected layer to classify results. The objective of a fully connected layer is to connect the input it sees to the correct type of output. This entails transforming an image feature matrix into a feature vector with dimensions of $1 \times C$, where C is the number of classes. The last fully connected output layer often uses the Softmax function as the activation function to aggregate all local features into global features and determine the score for each type. In our Jordan22 dataset, there are eight different

classes. The fully connected layer can turn feature maps into a single feature vector that has dimensions 1×8 . Then the Softmax function turns that vector into an 8-item long probability distribution in which each number in the resulting vector represents the probability that a given input image falls in class 1, class 2, class 3, ... class 8. This output was called the class scores and from these scores, we can extract the most likely class for the image. A complete CNN was built from convolutional, pooling, and fully connected layers, but there are extra layers that may be added to prevent overfitting. A dropout layer is one of the most often used layers to add to avoid overfitting. When the network trains, certain nodes may dominate others or make huge mistakes, hence dropout allows us to balance our network so that every node works equally towards the same objective, and if one makes a mistake, it does not dominate the behavior of our model. Dropout is a strategy for making a network robust; it ensures that all nodes operate effectively together by ensuring that no node is either weak or too powerful. Dropout layers probabilistically disable some nodes in a layer. Dropout guarantees that all nodes have an equal opportunity to attempt and classify different images during training, and it decreases the risk that a few, strongly weighted nodes would dominate the process.

4. Experimental Results

A number of modifications were made to the CNN structure. In the CNN model, we first investigated the effect of image size on learning. Secondly, we investigated the impact of max and average pooling on training and testing accuracy. Thirdly, we investigated the impact of changing the size of conv2D. Fourthly, we investigated the impact of adding padding to the Maxpooling layer. Finally, we examined the impact of the dropout layer in the CNN network. The data were divided into a ratio of 70:15:15 for training, validation, and testing respectively. dataset split into training, validation, and testing sets following the 70:15:15 ratio with 1618, 346, and 346 images, respectively.

4.1 Result of Changing the Resolution of the Images in Model Learning:

The CNN models No.1, No.2, No.3, No.4, and No.5 all have the same structure as described in Table 7, with the first conventional layer being 32, the second and third conventional layers being 64, and the remaining layers being 128. The dropout layers were used. The padding for the pooling layer was not used. The max pooling was used in the pooling layers; the only difference in those CNN models was the size of the images in the dataset.

Table 7
The structures of the first five CNN models.

CNN Structure	No.1, No.2, No.3, No.4, No.5
Conv2D	32
Padding for Conv2D	same
Max Pooling	(3,3)
Padding for Max Pooling	NA
Dropout	0.25
Conv2D	64
Conv2D	64
Max Pooling	(2,2)
Dropout	0.25
Conv2D	128
Conv2D	128
Max Pooling	(2,2)
Dropout	0.25
Dropout	0.5
Epoch	25

As shown in Table 8. We found that the CNN model No.3 with an image size of 256 pixels x 256 pixels had the greatest test accuracy of 90.65%.

Table 8
The result of changing the resolution of the images in the CNN model.

CNN Structure	No.1	No.2	No.3	No.4	No.5
Image size	(352,352)	(288,288)	(256, 256)	(224,224)	(128,128)
Train accuracy	0.9396	0.9538	0.9570	0.9448	0.9428
Train loss	0.1786	0.144	0.1352	0.1630	0.1633
Validation accuracy	0.7329	0.6318	0.9079	0.8484	0.8087
Validation loss	1.0281	1.7132	0.2285	0.7101	0.7974
Test accuracy	0.7194	0.5899	0.9065	0.8201	0.8345
Test loss	1.0712	1.6284	0.2956	0.7649	0.7589
Time (minutes)	58	41	34	24	9

In terms of test accuracy, model No.3 performed best when the picture size was 256 pixels × 256 pixels, while model No.2 performed worst when the image size was 288 pixels x 288 pixels, as shown in FIGURE 10.

4.2 The Effect of using the Average Pooling on CNN Learning:

The max pooling was used when studying the unique characteristics in the image and wanting the CNN model to learn a collection of unique properties in the image, while the average pooling is better when studying all of the features in the image [30]. Here, CNN model No.6 had the same structure as CNN model No.3, but in model No.6, we replaced all max pooling with average pooling to observe the effect of using average pooling instead of max pooling, as shown in Table 9 After using average pooling, the test's accuracy dropped to 0.7698 and the loss rate rose to 90.22%. This research aimed to investigate plant diseases that appear as distinct spots on plant leaves. In this case, it is better to use max pooling.

Table 9
The effect of using the average pooling on the CNN model.

CNN Structure	No.6
Conv2D	32
Padding for Conv2D	same
Average Pooling	(3,3)
Padding for Average Pooling	NA
Dropout	0.25
Conv2D	64
Conv2D	64
Average Pooling	(2,2)
Dropout	0.25
Conv2D	128
Conv2D	128
Average Pooling	(2,2)
Dropout	0.25
Dropout	0.5
Image size	(256, 256)
Epoch	25
Train accuracy	0.9416
Train loss	0.1678
validation accuracy	0.8303
validation loss	0.5665
Test accuracy	0.7698
Test loss	0.9022
Time (minutes)	30

4.3 The Effect of Changing the Size of Conv2d Layers on CNN Learning:

Here, we investigated the effect of the size of the convolutional layers on the CNN model's performance. As shown in Table 10. We found that the best performance was obtained in model No.10 when the first convolutional layer had a size of 32 and the remaining convolutional layers had a size of 128. The test's accuracy here was 94.24%, and the error rate was 21.70%.

Table 10
The effect of the size of the convolutional layers on the CNN model's performance.

CNN Structure	No.7	No.8	No.9	No.10
Conv2D	64	64	128	32
Padding for Conv2D	same	same	same	same
Max Pooling	(3,3)	(3,3)	(3,3)	(3,3)
Padding for Max Pooling	NA	NA	NA	NA
Dropout	0.25	0.25	0.25	0.25
Conv2D	64	128	128	128
Conv2D	64	128	128	128
Max Pooling	(2,2)	(2,2)	(2,2)	(2,2)
Dropout	0.25	0.25	0.25	0.25
Conv2D	128	512	128	128
Conv2D	128	512	128	128
Max Pooling	(2,2)	(2,2)	(2,2)	(2,2)
Dropout	0.25	0.25	0.25	0.25
Dropout	0.5	0.5	0.5	0.5
Image size Length *width	256	256	256	256
Epoch	25	25	25	25
Train accuracy	0.9512	0.9570	0.9615	0.9660
Train loss	0.1389	0.1123	0.0986	0.1004
validation accuracy	0.7924	0.8755	0.8755	0.9513
validation loss	0.9515	0.3045	0.3891	0.1728
Test accuracy	0.7914	0.8705	0.8561	0.9424

CNN Structure	No.7	No.8	No.9	No.10
Test loss	1.2720	0.3465	0.4874	0.2170
Time (minutes)	43	62	79	37

4.4 The Impact of Adding Padding to The Maxpooling Layer:

In models No11 and No12, the aim was to analyze the impact of adding padding to the max pooling layers on the performance of the CNN model. We discovered that the accuracy did not increase when adding the padding to the max pooling layers. As a result, we did not use the padding on the max pooling layers because it did not enhance the prior network outcomes. As seen in Table 11.

Table 11
The impact of adding padding to the Maxpooling layer.

CNN Structure	No.11	No.12
Conv2D	32	32
Padding for Conv2D	same	same
Maxpooling	(3,3)	(3,3)
Padding for Maxpooling	same	same
Dropout	0.25	0.25
Conv2D	64	128
Conv2D	64	128
Maxpooling	(2,2)	(2,2)
Dropout	0.25	0.25
Conv2D	128	128
Conv2D	128	128
Maxpooling	(2,2)	(2,2)
Dropout	0.25	0.25
Dropout	0.5	0.5
Image size	(256, 256)	(256, 256)
Epoch	25	25
Train accuracy	0.9383	0.9351
Train loss	0.1783	0.1665
validation accuracy	0.8267	0.8773
validation loss	0.6059	0.3300
Test accuracy	0.8561	0.8489
Test loss	0.4562	0.3198
Time (minutes)	29	35

4.5 The Impact of the Dropout Layer in the CNN Network:

In models, No.13, No.14, and No.15, the effect of the dropout layers on the CNN model was investigated as shown in Table 12. In model No.13 all dropout layers were set to 0.25, which produced satisfactory

results where the test accuracy was 90.65% and the test loss was 0.3599. Several dropout layers were removed in model No.14, resulting in a test accuracy of 0.8705 and a test loss of 0.7843.

When we compared model No.13 to model No.14, we discovered that deleting many dropout layers in the network resulted in a decrease in performance. All of the dropout layers were eliminated from the network in model No.15, and as a result, model No.15 has the lowest test accuracy rate of 82.01% and the lowest test loss of 0.6857 when compared to models No.13 and No.14. When applying the dropout layers with a value of 0.25 in model No.13, there was no difference in the training accuracy and test accuracy, which were 95.12% and 90.65% respectively.

However, in model No.15, where the dropout layer was not used, there was a difference in the results between the training accuracy and test accuracy, which were 96.66% and 82.01% respectively. We discovered that the training accuracy in model No.15 was higher than in models No.13 and No.14 and that the dropout layer is critical to avoid the problem of overfitting in the CNN network.

Table 12
The impact of the dropout layer in the CNN network.

CNN Structure	No.13	No.14	No.15
Conv2D	32	32	32
Padding for Conv2D	same	Same	same
Maxpooling	(3,3)	(3,3)	(3,3)
Padding for Maxpooling	NA	NA	NA
Dropout	0.25	NA	NA
Conv2D	128	128	128
Conv2D	128	128	128
Maxpooling	(2,2)	(2,2)	(2,2)
Dropout	0.25	NA	NA
Conv2D	128	128	128
Conv2D	128	128	128
Maxpooling	(2,2)	(2,2)	(2,2)
Dropout	0.25	NA	NA
Dropout	0.25	0.5	NA
Image size Length *width	256	256	256
Epoch	25	25	25
Train accuracy	0.9512	0.9608	0.9666
Train loss	0.1499	0.1144	0.1063
validation accuracy	0.8989	0.8574	0.8394
validation loss	0.3194	0.6463	0.6402
Test accuracy	0.9065	0.8705	0.8201
Test loss	0.3599	0.7843	0.6857
Time (minutes)	34	33	33

The CNN model No. 10 with the use of max pooling, multiple dropout layers, the first conventional layer size of 32, the remaining conventional layer size of 128, and the model trained at 25 epochs was the best CNN structure that provided the highest accuracy and lowest loss.

Here, we attempted to apply the early stop and increase the number of epochs for CNN model No.10 to 100. The model stopped at epoch 42 after employing the early stop. Table 12 Displays the model No. 10 result.

Table 13
The performance for model No.10.

Test accuracy	Test loss	Precision	Recall	F1- score	Epoch	Early Stop
0.9712	0.0783	0.97	0.96	0.96	100	42

We noticed the model improves after 20 epochs in, Fig. 11 which shows the accuracy of training and validation at 42 epochs for model No.10.

Figure 12, Shows the training and validation losses when the epoch number is 100 and the model was stopped at 42 using early stopping. Due to dropout layers, there is no problem with overfitting and no dispersion in the results.

4.6 Comparison With Other Approaches

The CNN model was used on the PlantVillage dataset in 2021, [9] to classify 38 types of healthy and unhealthy plants with an overall accuracy of 0.88. Our proposed CNN model No.10 was used to classify 38 classes from the PlantVillage, providing a training accuracy of 0.9488 and a validation accuracy of 92.54%. In 2020, this study [31] used CNN to classify 15 categories of healthy and unhealthy plants using 5032 images for training and 1220 images for validation from the PlantVillage dataset and the CNN model provides a training accuracy of 83.73%. When applying our proposed CNN to classify 15 classes from PlantVillage, the proposed CNN model had a training accuracy of 95.94% and the validation accuracy was 94.72%. This study [32] used a CNN model on the PlantVillage dataset to classify three maize diseases, and CNN provides a validation accuracy of 94.63%. When applying our proposed CNN to classify three classes in PlantVillage, the proposed CNN model has a validation accuracy of 94.72%. We found that using the proposed CNN model and resizing the image to 256 pixels x 256 pixels produces better results.

In Table 14, we compare our proposed CNN model with a set of researchers' results on the Planet Village dataset. According to the results, the proposed CNN model performs better than previous studies on the same database.

Table 14
Comparing the performance for model No.10 with previous research works in PlantVillage dataset.

Reference	Classes	Model	Train Accuracy	Validation Accuracy
[9]	38	CNN	NA	0.88
[31]	15	CNN	0.8373	0.8273
[32]	3	CNN	0.9964	0.9463
Proposed CNN	38	CNN	0.9488	0.9254
Proposed CNN	15	CNN	0.9594	0.9480
Proposed CNN	3	CNN	0.9650	0.9472

5. DISCUSSION

In this research, we used a CNN model to identify the images of diseased and healthy plants that were captured from the 12-megapixel resolution smartphone camera. The images were taken in the year 2022 from different farms in Jordan. The dataset named Jordan22 contains eight different categories of diseased and healthy plants of four agricultural crops, which are tomatoes, cucumbers, lettuce, and cabbage. The research contribution was to determine the ideal CNN structure by adjusting the layers such as the convolutional, pooling, and fully connected layers. We build a different CNN structure by changing the parameter values for the different layers.

Previous experiments yielded the following results:

- When the image size is 256 pixels × 256 pixels, the model performs better compared to other image sizes.
- In plant disease classification, the max-pooling layer outperforms average pooling.
- The best structure was obtained when the first convolutional layer had a size of 32 and the remaining convolutional layers had a size of 128.
- Adding padding to the max-pooling layer did not improve performance.
- Several dropout layers were required in order to produce a strong CNN model.
- Removing the dropout layer results in overfitting.

6. CONCLUSION

Plant diseases represent a major threat to global food security, causing farmers to incur financial losses as a result of agricultural pest damage, disease transmission, and misdiagnosis. It is critical to detect plant diseases early in order to maintain global food security.

The goal of this research was to use a CNN model to identify several plant diseases in a newly constructed dataset called "Jordan22." This dataset includes a diverse range of plant species from various agricultural crops in Jordan in 2022, such as tomatoes, cucumbers, cabbage, and lettuce. To address difficulties with misclassified data in the dataset, plant disease, and preventive experts performed a complete grouping of the Jordan22 dataset.

The study focuses on using CNNs to identify eight distinct plant species, using various data augmentation techniques such as shear range, zoom range, horizontal flipping, fill mode, and rotation range utilized to improve the training dataset. Convolutional layers were used for feature extraction, pooling layers for sample reduction, and dense layers using the SoftMax activation function for prediction. With an image resolution of 256 pixels x 256 pixels, the best results were obtained. Several CNN architectures were explored, and it was observed that incorporating dropout layers improved accuracy, while using max-pooling instead of average pooling enhanced performance. The CNN model labeled as No.10 achieved high accuracy, with a test accuracy of 0.9712 and a test loss of 0.0783 when trained for 42 epochs. To mitigate overfitting, a dropout layer and early stopping were employed in the CNN model.

7. FUTURE WORK

In the realm of plant disease detection using CNN models, several promising avenues for future research and development have been identified. These include expanding the dataset with more plant species and diseases, fine-tuning model architectures, exploring transfer learning, and real-time monitoring. Additionally, Extend the model's capabilities to not only classify diseases but also to localize and assess disease severity within plant images and integrate the model with a decision support system that provides actionable recommendations for disease management, including treatment options and preventive measures.

Declarations

Author Contribution

Author Contributions:1- Moy'awiah A. Al-Shannaq: conceived the research idea, wrote the paper, and contributed to the analysis and interpretation of the results. Additionally, he played a role in reviewing and revising the manuscript. 2- Shahed AL-Khateeb: performed the data analysis and played a key role in collecting the data. She contributed to the interpretation of the findings and provided essential inputs for shaping the research outcomes. 3-Abed Al-Raouf K. Bsoul : contributed to the proofreading and organization of the paper, ensuring clarity and coherence in the final manuscript. He also provided critical feedback and suggestions for improving the overall quality of the research.4- Ahmad A. Saifan: contributed to the interpretation of the results and provided valuable insights throughout the research process and ensured clarity and coherence in the final manuscript.

Data availability

The data that support the findings of this study are openly available in [Jordan22_Dataset] at [https://github.com/shahd1995913/Jordan22_Dataset], reference number [17].

[17] AL-Khateeb;Bsoul; Al-Shannaq, "Jordan22 Dataset used in Identification of Plant Diseases in Jordan Using Convolutional Neural Networks." Accessed: Nov. 21, 2023. [Online]. Available: https://github.com/shahd1995913/Jordan22_Dataset

References

1. L. Li, S. Zhang, and B. Wang, "Plant Disease Detection and Classification by Deep Learning - A Review," *IEEE Access*, vol. 9. Institute of Electrical and Electronics Engineers Inc., pp. 56683–56698, 2021. doi: 10.1109/ACCESS.2021.3069646.
2. R. Lewis, "'To Hell or Connaught:' How British Colonizers Both Caused and Benefitted from the Irish Potato Famine," 2020. [Online]. Available: <http://www.jstor.org/stable/650264>.
3. "Sustainable Development Goals | UNDP in the Arab States." Accessed: Jan. 11, 2022. [Online]. Available: <https://www.arabstates.undp.org/content/rbas/en/home/sustainable-development-goals.html>
4. A. Rakhmandasari, W. F. Mahmudy, and T. Yulianti, "Kenaf plant pest and disease detection using faster regional based convolutional neural network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 1, pp. 198–207, Oct. 2021, doi: 10.11591/ijeecs.v24.i1.pp198-207.
5. Deepa, N. Rashmi, and C. Shetty, "A Machine Learning Technique for Identification of Plant Diseases in Leaves," in *Proceedings of the 6th International Conference on Inventive Computation Technologies, ICICT 2021*, Institute of Electrical and Electronics Engineers Inc., Jan. 2021, pp. 481–484. doi: 10.1109/ICICT50816.2021.9358797.
6. T.-H. Meen, Institute of Electrical and Electronics Engineers, and International Institute of Knowledge Innovation and Invention, *Biomedical Engineering, Healthcare and Sustainability: 2nd IEEE Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability 2020: Tainan, Taiwan, May 29-31, 2020*.
7. G. Singh Sujawat and J. Singh Chouhan, "Application of Artificial Intelligence in detection of diseases in plants: A Survey," 2021.
8. K. Niha, S. Amutha, and A. Banu, "A Convolutional Neural Network based System to Detect Plant Disease," *Webology*, vol. 18, no. Special Issue, pp. 944–962, 2021, doi: 10.14704/WEB/V18SI04/WEB18175.
9. P. Srivastava, K. Mishra, V. Awasthi, V. Kumar Sahu, and P. Kumar Pal, "PLANT DISEASE DETECTION USING CONVOLUTIONAL NEURAL NETWORK," *Int J Adv Res (Indore)*, vol. 9, no. 01, pp. 691–698, Jan. 2021, doi: 10.21474/IJAR01/12346.

10. R. Ali and Y. J. Cha, "Attention-based generative adversarial network with internal damage segmentation using thermography," *Autom Constr*, vol. 141, p. 104412, Sep. 2022, doi: 10.1016/J.AUTCON.2022.104412.
11. A. K. Saini, R. Bhatnagar, and D. K. Srivastava, "Citrus Fruits Diseases Detection and Classification Using Transfer Learning," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Aug. 2021, pp. 277–283. doi: 10.1145/3484824.3484893.
12. H. M S, †, Niteesha Sharma, Y Sowjanya, Ch. Santoshini, R Sri Durga, and V. Akhila, "Plant disease prediction using convolutional neural network," *EMITTER International Journal of Engineering Technology*, vol. 9, no. 2, pp. 283–293, Dec. 2021, doi: 10.24003/emitter.v9i2.640.
13. D. Melse, N. Fajri, W. Firdaus Mahmudy, and T. Yulianti, "Detection of Disease and Pest of Kenaf Plant using Convolutional Neural Network," 2021. [Online]. Available: www.jitecs.ub.ac.id
14. M. Jammula, "An Artificial Intelligence Framework for Plant Leaf Disease Detection and Classification Using AMBF with GKFCM and GLCM," *Alinteri Journal of Agriculture Sciences*, vol. 36, no. 1, pp. 443–450, Jun. 2021, doi: 10.47059/alinteri/v36i1/ajas21065.
15. I. Y. Purbasari, B. Rahmat, and C. S. Putra PN, "Detection of Rice Plant Diseases using Convolutional Neural Network," *IOP Conf Ser Mater Sci Eng*, vol. 1125, no. 1, p. 012021, May 2021, doi: 10.1088/1757-899x/1125/1/012021.
16. W. Haider, A. U. Rehman, N. M. Durrani, and S. U. Rehman, "A Generic Approach for Wheat Disease Classification and Verification Using Expert Opinion for Knowledge-Based Decisions," *IEEE Access*, vol. 9, pp. 31104–31129, 2021, doi: 10.1109/ACCESS.2021.3058582.
17. AL-Khateeb;Bsoul; Al-Shannaq, "Jordan22 Dataset used in Identification of Plant Diseases in Jordan Using Convolutional Neural Networks." Accessed: Nov. 21, 2023. [Online]. Available: https://github.com/shahd1995913/Jordan22_Dataset
18. D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: A dataset for visual plant disease detection," in *PervasiveHealth: Pervasive Computing Technologies for Healthcare*, ICST, Jan. 2020, pp. 249–253. doi: 10.1145/3371158.3371196.
19. A. Kumar, "Detection of Sugarcane Disease and Classification using Image Processing," *Int J Res Appl Sci Eng Technol*, vol. 7, no. 5, pp. 2023–2030, May 2019, doi: 10.22214/ijraset.2019.5338.
20. P. Kulkarni, A. Karwande, T. Kolhe, S. Kamble, A. Joshi, and M. Wyawahare, "Plant Disease Detection Using Image Processing and Machine Learning."
21. M. Govardhan and M. B. Veena, "Diagnosis of Tomato Plant Diseases using Random Forest," *2019 Global Conference for Advancement in Technology, GCAT 2019*, Oct. 2019, doi: 10.1109/GCAT47503.2019.8978431.
22. M. V. Applalanaidu and G. Kumaravelan, "A review of machine learning approaches in plant leaf disease detection and classification," in *Proceedings of the 3rd International Conference on Intelligent Communication Technologies and Virtual Mobile Networks, ICICV 2021*, Institute of Electrical and Electronics Engineers Inc., Feb. 2021, pp. 716–724. doi: 10.1109/ICICV50876.2021.9388488.

23. M. S. Arshad, U. A. Rehman, and M. M. Fraz, "Plant Disease Identification Using Transfer Learning," in *2021 International Conference on Digital Futures and Transformative Technologies, ICoDT2 2021*, Institute of Electrical and Electronics Engineers Inc., May 2021. doi: 10.1109/ICoDT252288.2021.9441512.
24. M. A. Noyan, "Uncovering bias in the PlantVillage dataset," Jun. 2022, doi: 10.48550/arxiv.2206.04374.
25. D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: A dataset for visual plant disease detection," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Jan. 2020, pp. 249–253. doi: 10.1145/3371158.3371196.
26. Department of Statistics, "Crops Statistics – Department of Statistics," 2019. Accessed: Jun. 26, 2022. [Online]. Available: <http://dosweb.dos.gov.jo/agriculture/crops-statistics/>
27. "Image Augmentation Keras | Keras ImageDataGenerator." Accessed: Feb. 09, 2022. [Online]. Available: <https://www.analyticsvidhya.com/blog/2020/08/image-augmentation-on-the-fly-using-keras-imagedatagenerator/>
28. S. Mostafa and F. X. Wu, "Diagnosis of autism spectrum disorder with convolutional autoencoder and structural MRI images," *Neural Engineering Techniques for Autism Spectrum Disorder: Volume 1: Imaging and Signal Analysis*, pp. 23–38, Jan. 2021, doi: 10.1016/B978-0-12-822822-7.00003-X.
29. M. Yani, B. Irawan, and C. Setiningsih, "Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry's Nail," *J Phys Conf Ser*, vol. 1201, no. 1, May 2019, doi: 10.1088/1742-6596/1201/1/012052.
30. Y.-L. Boureau, J. Ponce, J. P. Fr, and Y. Lecun, "A Theoretical Analysis of Feature Pooling in Visual Recognition," 2010.
31. S. Mahale, K. Shah, and I. Kandivali, "Detection of Plant Diseases Using Convolutional Neural Network Architectures," *International Journal of Intelligent Communication, Computing and Networks Open Access Journal*, pp. 2582–7707, doi: 10.51735/ijiccn/001/19.
32. A. Hidayat and U. Darusalam, "DETECTION OF DISEASE ON CORN PLANTS USING CONVOLUTIONAL NEURAL NETWORK METHODS," 2019.

Footnotes

1. Muhammad Al-Shennaq is an agricultural expert with more than 20 years of experience in detecting plant diseases
2. Ziad Nasser and Amer Mahasneh work in plant disease control department at the NARC.

Figures

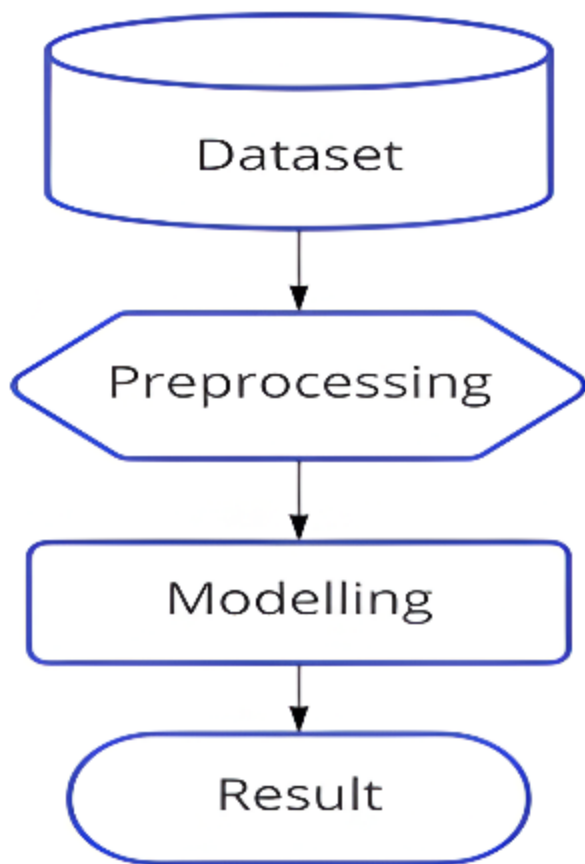


Figure 1

Overall research phases.



Figure 2

Different datasets show tomato blight.



Figure 3

Sample images in the dataset [17].



Figure 4

Sample of the image in the Jordan 22 dataset.

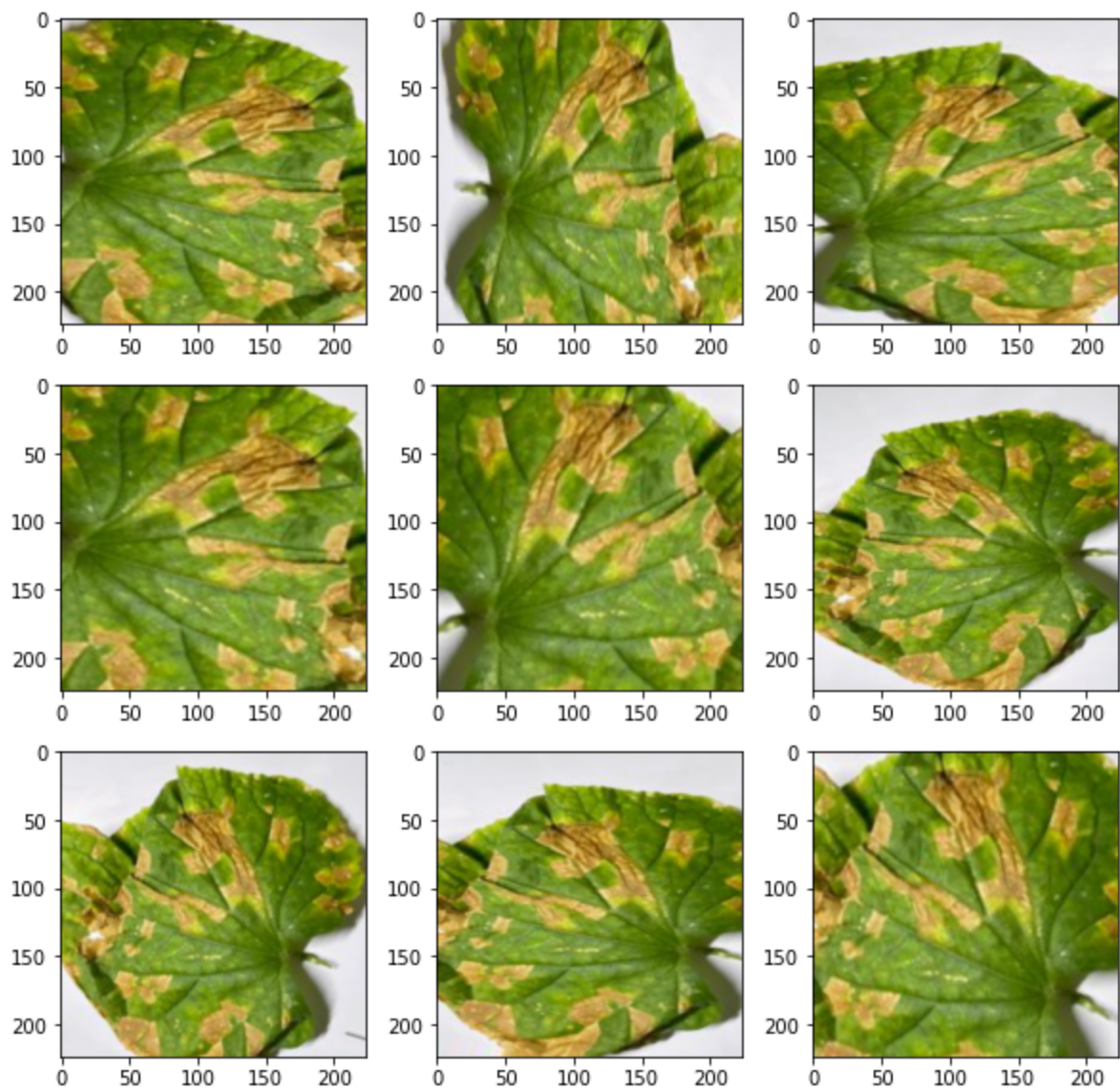


Figure 5

Image Transformations after executing ImageDataGenerator.

CNN

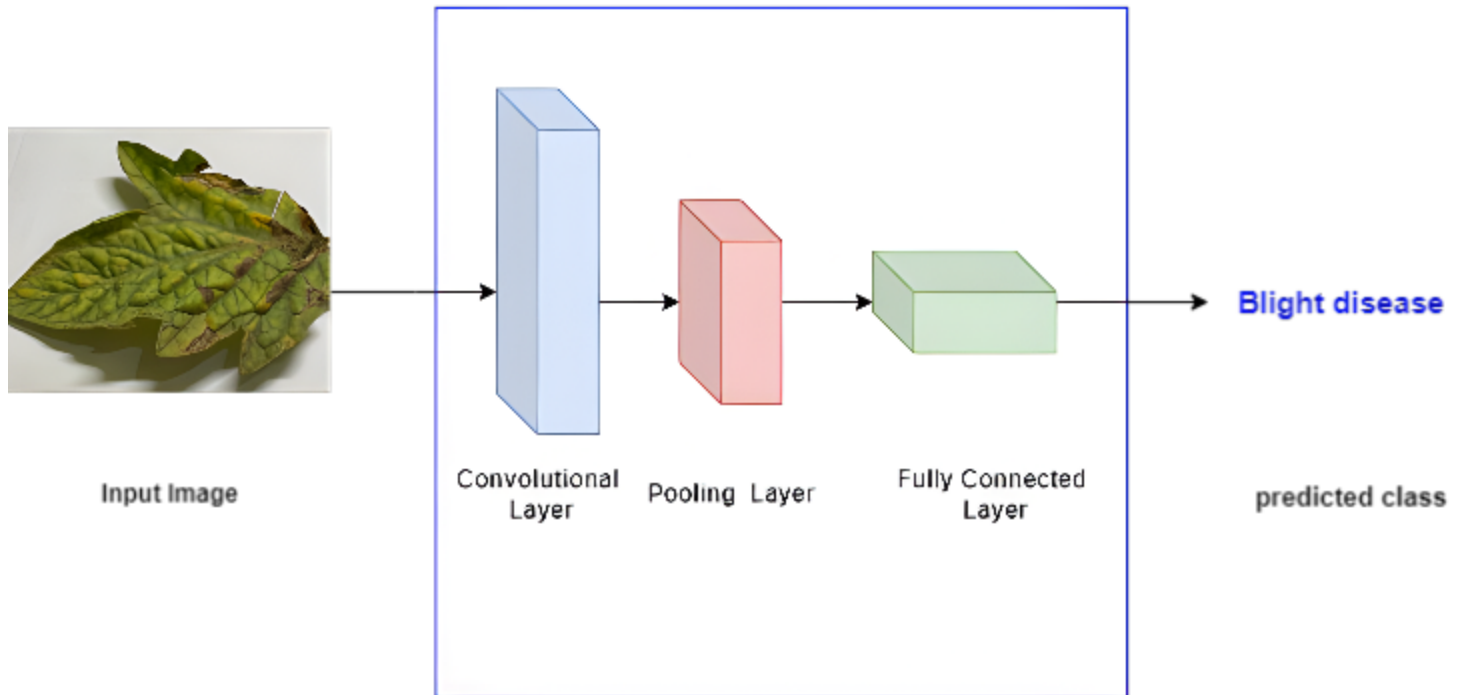


Figure 6

Layers in a CNN.

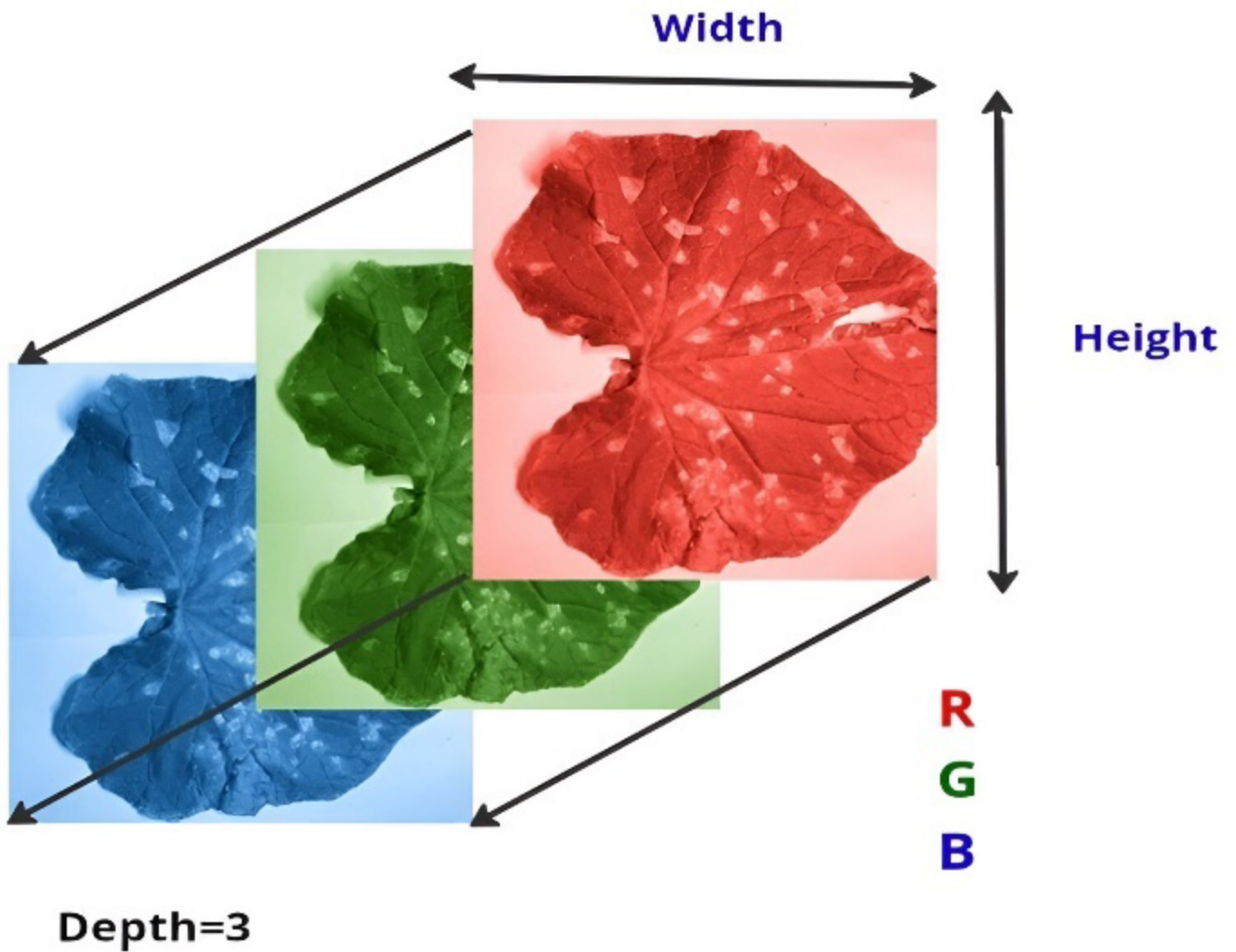


Figure 7

RGB layers of cucumber disease.

1	1	1	0	0
0	1	1	1	0
0	0	1*1	1*0	1*1
0	0	1*0	1*1	0*0
0	1	1*1	0*0	0*1

Input image

1	0	1
0	1	0
1	0	1

Filter size 3*3

4	3	4
2	4	3
2	3	4

Convolved feature

Figure 8

Convolution layer.

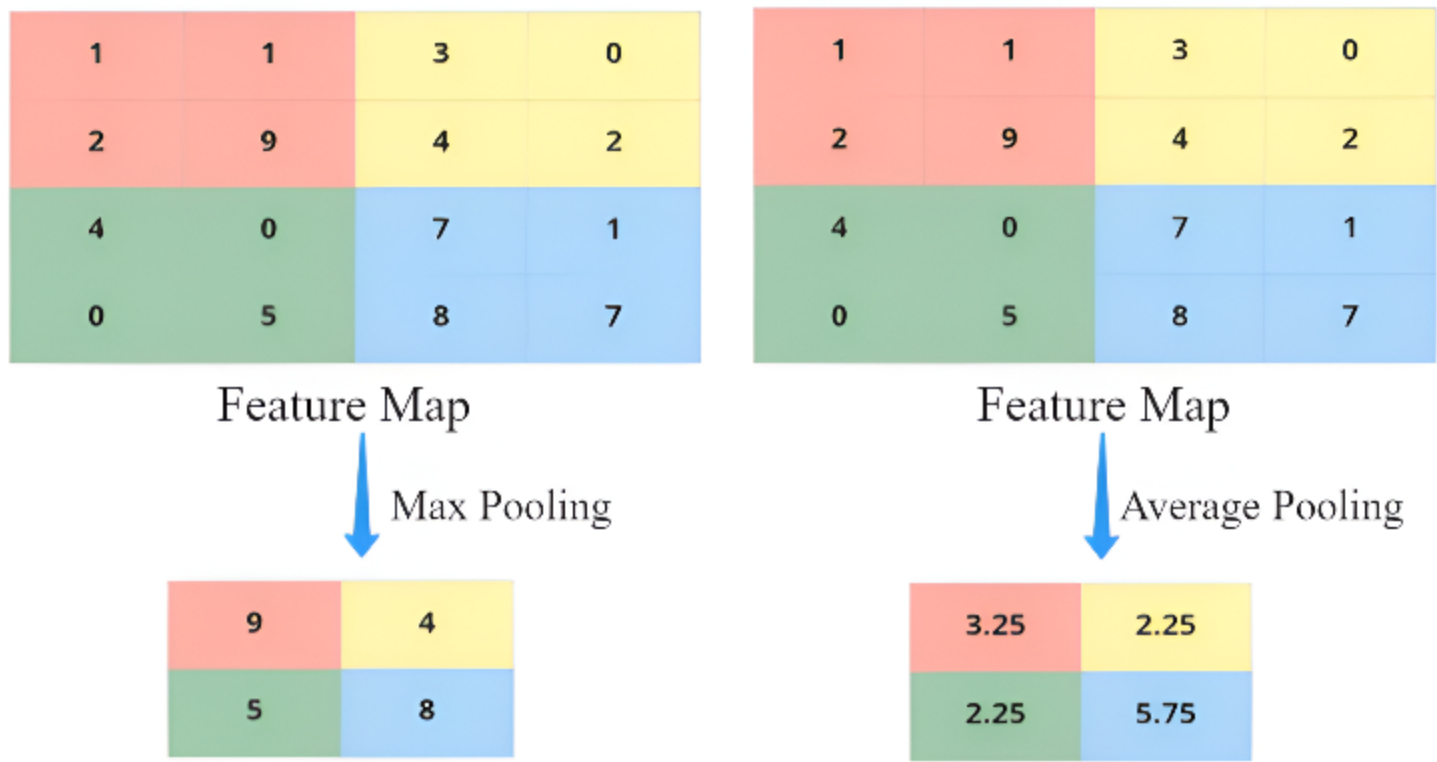


Figure 9

The max pooling layer vs. the average pooling layer[29].

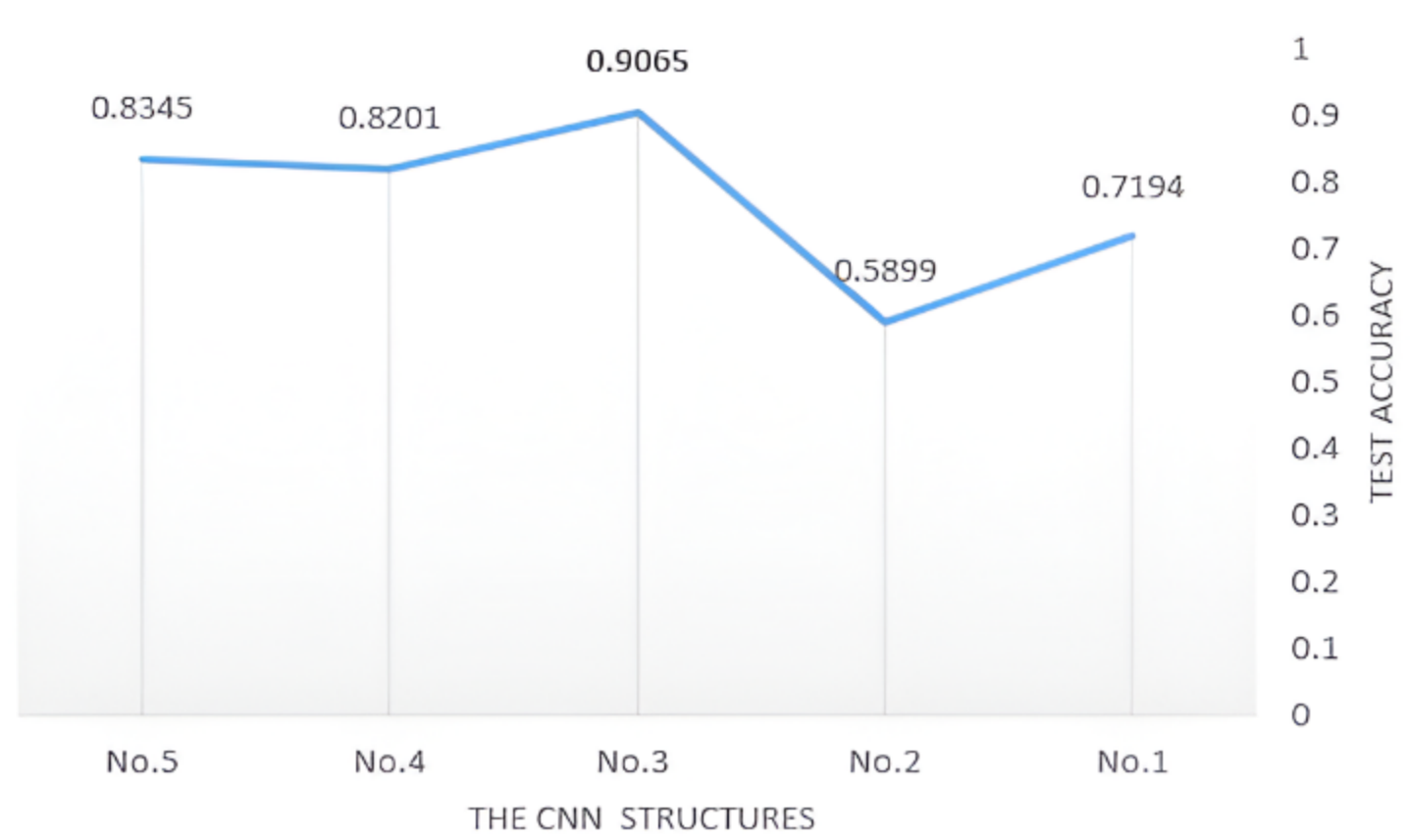


Figure 10

The effect of changing the size of the images in the CNN model.

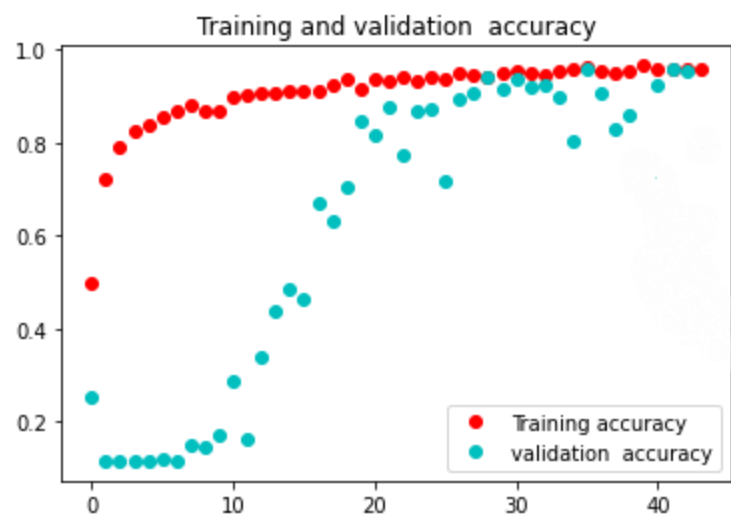


Figure 11

Accuracy of training and validation.

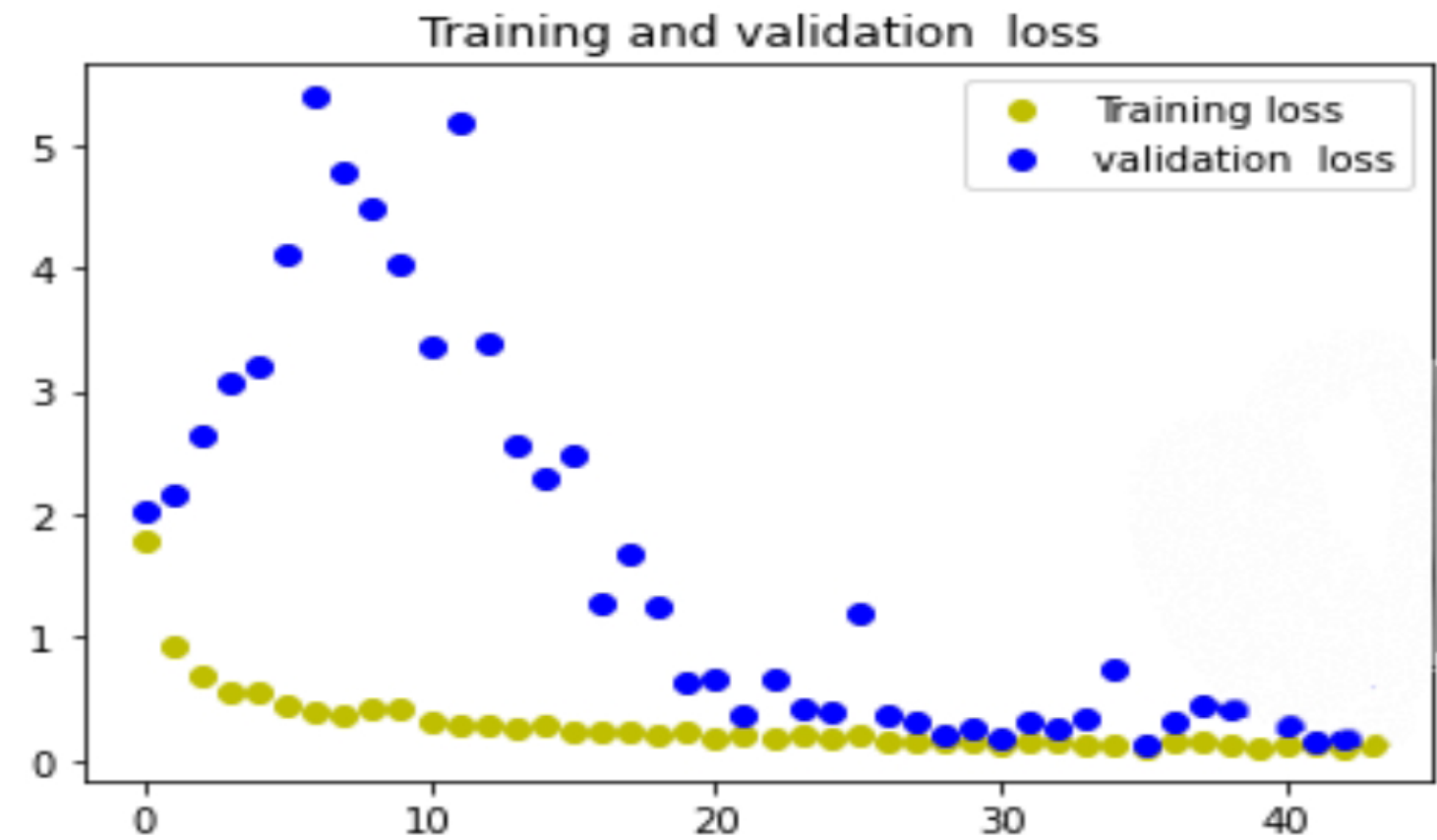


Figure 12

Training and validation losses.

