Disease Recognition in crops using Convolutional Neural Networks

A Major Project Report

Submitted in partial fulfillment of the

Requirements of VIII-Semester for the degree

Of

Bachelor of Technology

in

COMPUTER SCIENCE & ENGINEERING

by

Ashutosh S. Tripathi (17115015)

. Under the guidance of

Dr. Naresh Kumar Nagwani

Associate Professor

NIT-Raipur

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY

RAIPUR, CG (INDIA)

APRIL, 2020

DECLARATION

We hereby declare that the work described in this thesis, entitled **"Disease Recognition in crops using Convolutional Neural Networks"** which is being submitted by us in partial fulfillment for the VIII-Semester of the degree of Bachelor of Technology in the Department of **Computer Science and Engineering** to the National Institute of Technology Raipur is the result of investigations carried out by us under the guidance of **Dr. Naresh Kumar Nagwani (Associate Professor).**

The work is original and has not been submitted for any Degree/Diploma of this or any other Institute/university.

Name of Candidate and Roll No:

Ashutosh, 16115002

Place: Raipur

Date: 15.04.2021

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled "**Commodity Inflation Prediction using Fuzzy Logic On Economic Parameters**" that is being submitted by Abhisek Nayak (Roll no. 16115002), Aditya (Roll no. 16115003), Abhishek Dutta (Roll no. 16115901) in partial fulfillment for VIII-semester of the degree of Bachelor of Technology in Computer Science & Engineering to National Institute of Technology Raipur is a record of bonafide work carried out by them under my guidance and supervision.

The matter presented in this project document has not been submitted by them for the award of any other degree elsewhere.

Signature of Supervisor

Dr. Naresh Kumar Nagwani

Associate Professor

Department of Computer Science & Engineering

National Institute of Technology, Raipur (C.G)

Project Coordinator	H.O.D
Dr. K. Jairam Naik	**Dr. Pradeep Singh**
Department of Computer Science & Engineering	**Department of**
C.S.E National Institute of Technology, Raipur (C.G)	**NIT Raipur (C.G)**

ACKNOWLEDGEMENT

I would like to acknowledge my college **National Institute of Technology, Raipur** for providing a holistic environment that nurtures creativity and research- based activities. I express our sincere thanks to **Dr. Naresh Kumar Nagwani, Associate Professor CSE Department NIT Raipur,** the guide of the project for guiding and correcting throughout the process with attention and care. He has frequently suggested us creative ideas and guided through major hurdles that occurred during the duration of the project.

I would also thank **Dr. Pradeep Singh, Head of the Department** and all the faculty members without whom this project would be a distant reality. I also extend my heartfelt thanks to my family and friends who supported me.

Thank You!!

Ashutosh S. Tripathi, 17115015

ABSTRACT

Over seventy percent of the population in the Indian villages relies on farming, more than eighty percent of whom are either small or marginal in their scope of production. In the last year alone, globally, plant diseases caused more than 220 billion dollars worth of damage.

Thus, it is paramount for the farmers to quickly find out the dysfunctional and diseased crops, not by looking at the yield but by observing earlier factors and aid in developing plant resistance. The methods used thus far require a good experience in the field and are not completely reliable as the number of factors involved in predicting such a complex variable is too massive for an armchair guess.

In this project, the goal is to delegate this task to the computers, or specifically, to convolutional neural networks. I use approximately 87 thousand images to train a deep learning model in defining diseased and healthy crops. The software used here is Jupyter; it is one of the most robust environments that provide a versatile library for machine learning. By the end, the algorithm received over 97 percent accuracy over the training set and 91 percent accuracy on the previously unforeseen test set.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Overview:

In the dawning age of farming, men would find the crops grown in nature and feed the cattle which would lead their small tribe to relative prosperity in comparison to the hunter-gatherer counterparts. In the distant past, the people relied on their own lands to cultivate their food and feed their local community.
Even now, India is an agrarian economy and so agricultural practices in India have been absolutely crucial to the country's growth. With the rise of novel innovations, we see an increase in the gross output but an overall decrease in farmer satisfaction
Even with the growth in modern processes. they have become inaccessible and abstruse for a common farmer to use. Along with these techniques and the erratic nature of the climate, bountiful crop production has become quite uncertain.

In theory, the possibility of an economic explosion of growth is tangible, yet without a proper direction, the probability seems dubious at best. Needless to say, there are myriad of possible ways to increase crop productivity, one of the most promising of them is data mining and image recognition. Essentially, data mining is the process of creating valuable, practical information out of seemingly random, chaotic, and noisy data. In the purest of terms, it turns data into information. Generally speaking, these softwares allow us to find the underlying pattern between a dozen or so vaguely related variables, each with hundreds or more values. Thus various links between each variable, patterns, and interlinks can be found with ease. This information is then used to infer the knowledge required to map out the historical repetitions, present trends, and future predictions.

Giving a practical example, should the algorithm recognize a novel plant disease, farmers now have an option to cut their losses and preemptive preparation for climate based The scope of the application of such software is huge. The sheer number of livelihoods that can be tangibly improved with this is astounding. Before this, the only variable available to predict the yield was the farmer's anecdotes and his inherently limited knowledge about plant disease. An algorithm like this can also decrease the risk of causing damage to a crop by using a wrong pesticide. With the aforementioned precision, the farmer will be liable and prepared for creating crop-failure avoidance tactics. Therefore, in this project, the notion for creating an algorithm that helps in plant disease recognition is a stepping stone for improved crop quality.bjectives and Importance of the Project

The main objectives of the project are:

- To facilitate the use and awareness of technology for crop diseases.
- To alert the farmers about previously unrecognized plant diseases.
- To reduce the risk of crop failure.
- To reduce the chances of immunity building by inefficient use of pesticides
- To forecast the annual yield with the help of regression models.
- To track the impact of specific plant disease.
- To provide a correlation between climate change and plant disease.
- To empower farmers to find the crop failures by themselves.
- To help farmers to get reasonable prices for their produce.
- Increase the general productivity of farmers

Scope

As mentioned earlier, this project is largely focused on improving the general methodology of farming and making it more efficient by detecting . Thus the brunt of the benefits of this project will be enjoyed by the farmers.

With this project, it will be very easy to predict the seasonal output produced by the farmers of any particular crop at any particular season even though the data used here is specifically considering the rice crop in the Kharif season. Thus, the possibility of a monumental increase in output lies just ahead. However, the possibility of this algorithm being useful in other areas is not to be discounted. With enough data at our disposal, the random forest algorithm is useful at virtually any field that could benefit from its predictive capabilities.

Thus, in conclusion, this application can be of great value to people who've a vested interest in bountiful crop production.

Motivation

According to a recent survey, over 180 million people in India still suffer from malnutrition. This is an appallingly huge number that can directly be brought down by increasing the overall crop production.

Curing hunger in the country is a truly behemoth task that would require years of hard work. I believe this algorithm creates the foundation for this task. In this changing environment, appropriate and timely disease identification including early prevention has never been more important. There are several ways to detect plant pathologies. Some diseases do not have any visible symptoms, or the effect becomes noticeable too late to act, and in those situations, a sophisticated analysis is obligatory. However, most diseases generate some kind of manifestation in the visible spectrum, so the naked eye examination of a trained professional is the prime technique adopted in practice for plant disease detection. In order to achieve accurate plant disease diagnostics a plant pathologist should possess good observation skills so that one can identify characteristic symptoms. Variations in symptoms indicated by diseased plants may lead to an improper diagnosis since amateur gardeners and hobbyists could have more difficulties determining it than a professional plant pathologist. An automated system designed to help identify plant diseases by the plant's appearance and visual symptoms could be of great help to amateurs in the gardening process and also trained professionals as a verification system in disease diagnostics.

The secondary motivation of this project is to create a better integration between machine learning and farming so that more algorithms of all sorts can be developed.

Overview of the Project

In the first chapter, an introduction to the method and its possible applications is provided. The succeeding chapter involves a brief overview of relevant studies. The third hapter discussed the methodology whereas the fourth discusses the procedure of creating the program in Rstudio. The subsequent chapters involve the results, future applications and the conclusion drawn from this project respectively.

2. LITERATURE REVIEW

2.1 Existing Work

Artificial neural networks (ANN) is an essential part of the project and a lot of research has been done in this field. Some types of ANNs are:

- Shallow learning neural networks.g
- Deep learning neural networks.
- Convolutional neural networks.
- Recurrent neural networks.

In “A review of advanced techniques for detecting plant diseases” by Sindhuja Sankaran et al, compares various methods in real-time tracking of plant health [1]. Technologies like molecular polyamarase chain reactions and ground based sensor techniques are discussed.

“Digital image processing techniques for detecting, quantifying and classifying plant diseases” by Jayme Garcia Arnal Barbedo in 2013 presents a survey of methods used to classify plant diseases through digital image processing [2]. In spite of the fact that illness manifestations can show in any piece of the plant, just the methodologies that investigate noticeable side effects in leaves and stems were thought of. This was accomplished for two principle reasons: to restrict the length of the paper and in light of the fact that strategies managing roots, seeds and natural products have a few idiosyncrasies that would warrant further study. The chosen recommendations are partitioned into three classes as indicated by their goal: location, seriousness measurement, and arrangement. Every one of those classes, thus, are partitioned by the principle specialized arrangement utilized in the calculation. This paper is required to be helpful to specialists working both on plant pathology and example acknowledgment, giving a complete and open outline of this significant field of examination.

“Plant Disease Detection Technique Tool- A Theoretical Approach” by Prajwala Reddy et al aims for automatic detection in plant disease, and thus automatically detect the symptoms of diseases. This paper presents a method for identify plant disease based on color, edge detection and histogram matching. First phase concerns with training of healthy sample and diseased sample. Second phase concerns with the training of test sample and generates result based on the edge detection and histogram matching [3]

Various methods are right now being used for plant infection discovery applying PC vision. One of them is sickness identification by removing shading highlight as creators in [4] have introduced. In this paper YcbCr, HSI, and CIELB shading models were utilized in the investigation; accordingly, illness spots were effectively recognized and stayed unaffected by the clamor from various sources, for example, camera streak.

Also, plant sickness discovery could be accomplished by separating shape highlights strategy. Patil and Bodhe applied this procedure for illness recognition in sugarcane leaves where they have utilized limit division to decide leaf territory and triangle edge for lesioning region, getting the normal precision of 98.60% at the last tests[5].

Moreover, separating surface component could be utilized in distinguishing plant infections. Patil and Kumar proposed a model for plant sickness discovery utilizing surface highlights like inactivity, homogeneity, and relationship got by ascertaining the dark level cooccurrence framework on picture [6]. Joined with shading extraction, they investigated distinguishing infections on maize leaves.

Table 1: Table for existing work

S no.	Paper Title	Authors	Year	Method used	Limitation
1.	A review of advanced techniques for detecting plant diseases [1]	Sindhuja Sankarana , Ashish Mishraa, Reza Ehsania, Cristina Davisbe	2016	Spectroscopic and imaging-based method analysis	Small dataset results in inaccurate predictions
2.	Digital image processing techniques for detecting, quantifying and classifying plant diseases”[2]	Jayme Garcia Arnal Barbedo	2019	ANN	Convolutional Neural networks were not used in the project, leading to a shallower learning network
3.	Plant Disease Detection Technique Tool- A Theoretical Approach[3]	Prajwala R Reddy, Divya S N	2015	CBIR (content based image retrieval)	A considerably small diseased crop database.
4.	Color Transform Based Approach for Disease Spot Detection on Plant Leaf[4]	Piyush Chaudhary, Anand K Chaudhari, Sharda Godara	2012	CIELB shading model	Use of Macroscopic parameters and finding are limited to a handful of crops.

5.	A Robust Clustering Technique Based upon Density [5]	Kaur Prabhjot, Lamba I. M. S, and Gosain Anjana	2012	Noise Clustering Technique	Problem of overfitting.
6.	Feature extraction of diseased leaf images[6]	J. K. Patil and R. Kumar	2012	Approximate, rather than exact, modes of reasoning	Basic implementation and application determination.

Table 1: Related works

2.2 Summary

Thus, the various methods from varied fields are discussed and their advantages and disadvantages debated.

3. PROPOSED MODEL

3.1 About Proposed System

The proposed system has the following important components:

- The machine learning model.
- The Artificial Neural Network.

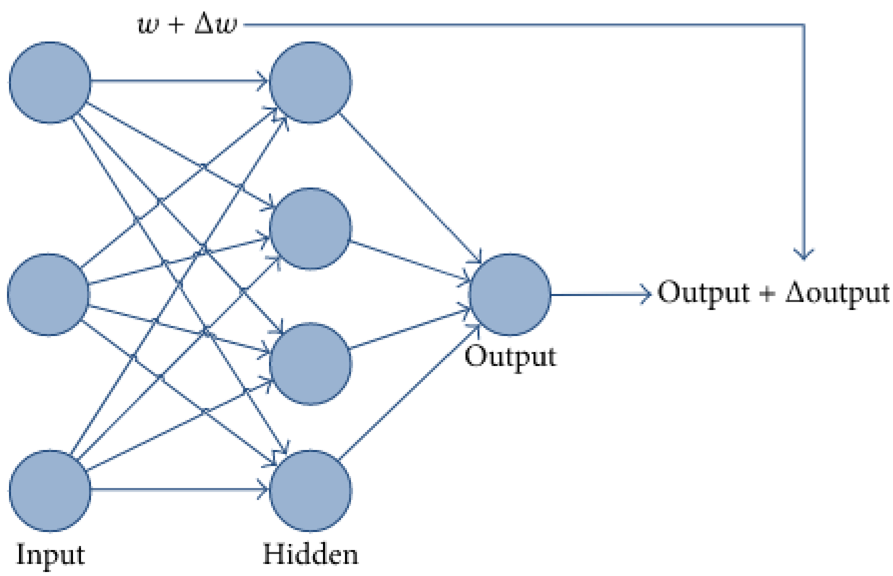


Fig 1: Flowchart showing working of a simplistic neural network

The machine learning model has been developed using 3 different Libraries aside from the OS library:

- Digital Image processing libraries
- Data mining libraries
- Machine learning libraries

```
In [32]: import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         import tensorflow as tf
         from tensorflow.keras import layers
         from keras.preprocessing.image import ImageDataGenerator
         import os

In [33]: tf.__version__

Out[33]: '2.3.0'
```

Fig 2: A code snippet of libraries installed.

3.2 Dataset

Appropriate datasets are required at all stages of object recognition research, starting from training phase to evaluating the performance of recognition algorithms. All the images collected for the dataset were downloaded from the Internet, searched by disease and plant name on various sources in different languages. Open source dataset site Kaggle was used to procure the majority of the images.

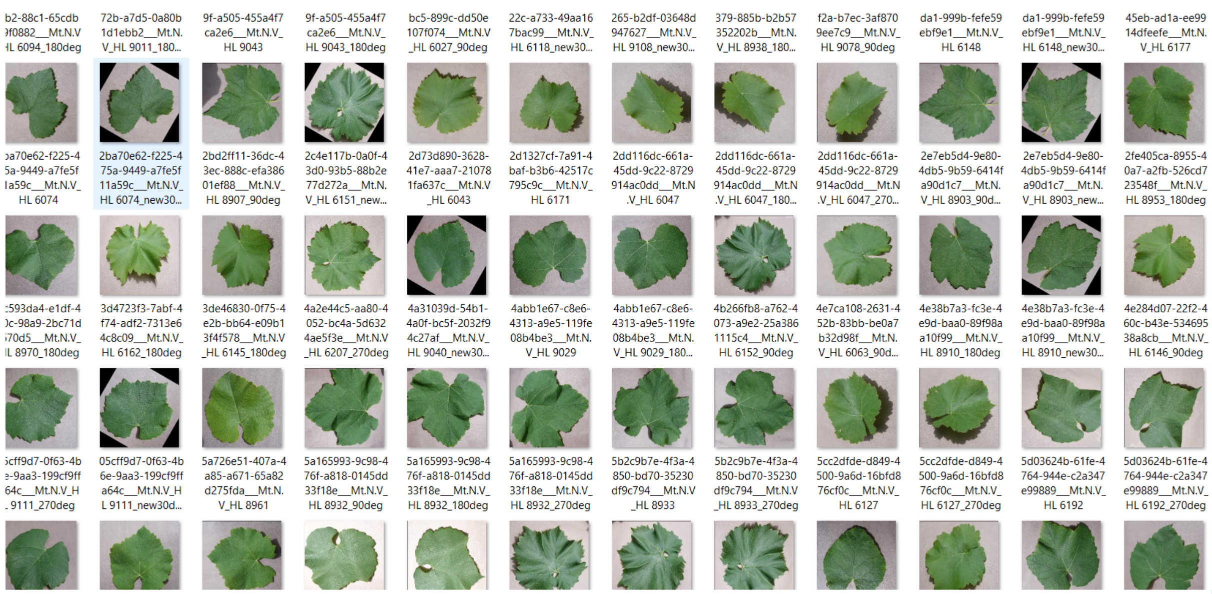


Figure 3: Sample of the dataset

In this stage, all copied pictures taken from various sources were eliminated by created python content applying the contrasting system. The content eliminated the copies by looking at the pictures' metadata: name, size, and the date. After the mechanized expulsion, pictures were evaluated by human specialists in much emphasis.

Next step was to enrich the dataset with augmented images. The main goal of the presented study is to train the network to learn the features that distinguish one class from the others. Therefore, when using more augmented images, the chance for the network to learn the appropriate features has been increased. Finally, a database containing 70295 images for training and 17572 images for validation has been created.

Type of crop (no. of subclasses)	No. of images in Training set	No. of images in Validation set
Apple (4)	7771	1943
Blueberry (1)	1816	454
Cherry (2)	3509	877
Corn (4)	7316	1829
Grape (4)	7222	1805
Orange (1)	2010	503
Peach (2)	3566	891
Pepper (2)	3901	975
Potato (3)	5702	1426
Soyabean and pepper (2)	2022	505
Squash (1)	1736	434
Strawberry (2)	3598	805
Tomato (8)	18,349	4585

Table 2: Crops and their image distribution

Table 1 shows all supported CROPS together with the number of original images for every class used as training and validation dataset for the disease classification model along with the no. of subclass.

3.3. Image Preprocessing and Labelling

Pictures downloaded from the Internet were in different arrangements alongside various dimensions and quality. To improve include extraction, last pictures proposed to be utilized as dataset for convolutional neural network classifier were preprocessed to acquire consistency.

In [44]:
```
batch_size = 100
img_height = 250
img_width = 250
training_ds = tf.keras.preprocessing.image_dataset_from_directory(
    'dataset/train',
    seed=42,
    image_size= (img_height, img_width),
    batch_size=batch_size

)
```
```
Found 70295 files belonging to 38 classes.
```

Figure 4: Image preprocessing phase code snippet.

Moreover, methodology of picture preprocessing included editing of the relative dimensions of pictures physically, making the square around the leaves, to feature the area of interest (plant leaves). During the period of gathering the pictures for the dataset, pictures with more modest goal and measurement under 500 px were not considered as legitimate pictures for the dataset. Furthermore, just the pictures where the area of interest was in higher goal were set apart as qualified possibility for the dataset. Around there, it was guaranteed that pictures contain all the required data for highlight learning. Pictures utilized for the dataset were picture resized to diminish the hour of preparing. The standard measurements were 250x250. The entire training dataset was divided into 703 batches, each of which had 100 images.

As the dataset involved plenty of randomised images, they didn't require further image manipulation to avoid overfitting.

Figure 5: A post-processed image

3.4 Training the neural network

Training the deep convolutional neural network for making an image classification model from a dataset described in Section 3.1 was proposed. There are several well-known state-of-the-art deep learning frameworks, such as Python library Keras and machine learning library that extends tenserflow is used. Data mining libraries like panda and matplotlib etc. are required. Finally the time library is required to judge the performance of the code.

Convolutional neural networks essentially are useful for image feature recognition. The way they function is by converting the image into the list of features that are being matched by the desired image.

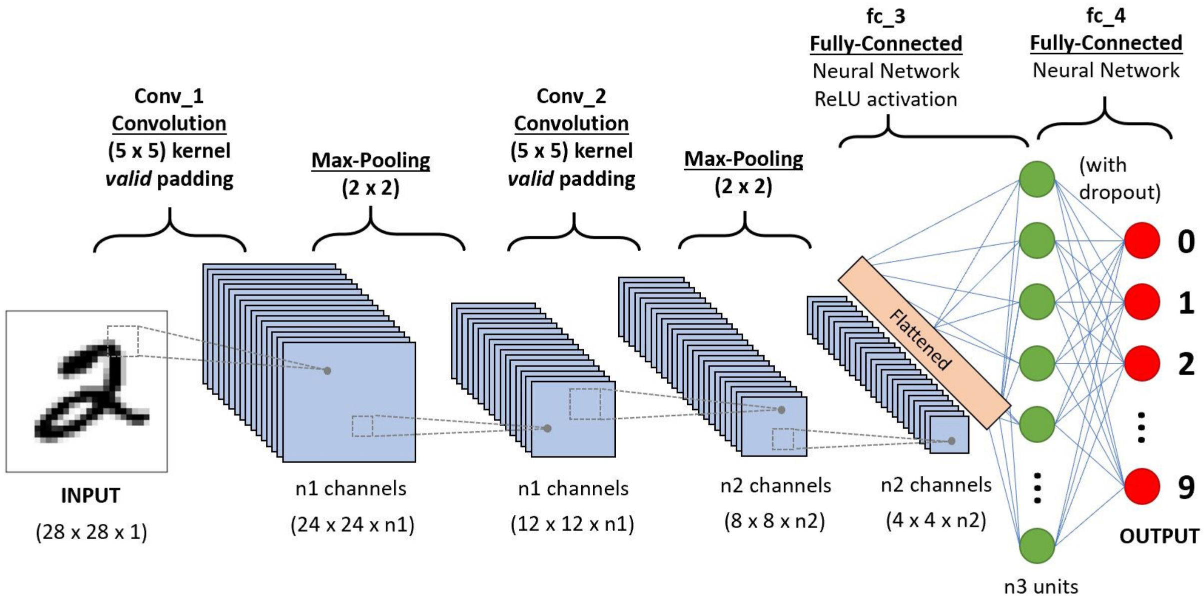


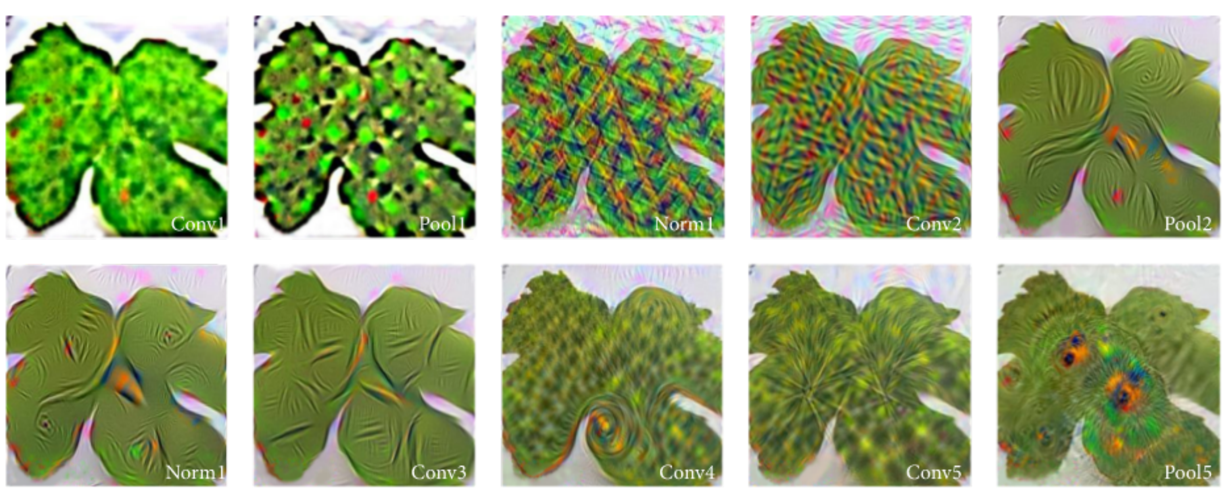
Figure 6: A pictorial representation of working of CNN

The CNN has layers that perform 4 broad functions and they are listed below in that order:

Step 1: Convolution
Step 2: Max-pooling
Step 3: Flattening
Step 4: Full connection

Step 1: Convolution:

In mathematics (in particular, functional analysis), convolution is a mathematical operation on two functions (f and g) that produces a third function ( ) that expresses how the shape of one is modified by the other. The term convolution refers to both the result function and to the process of computing it.



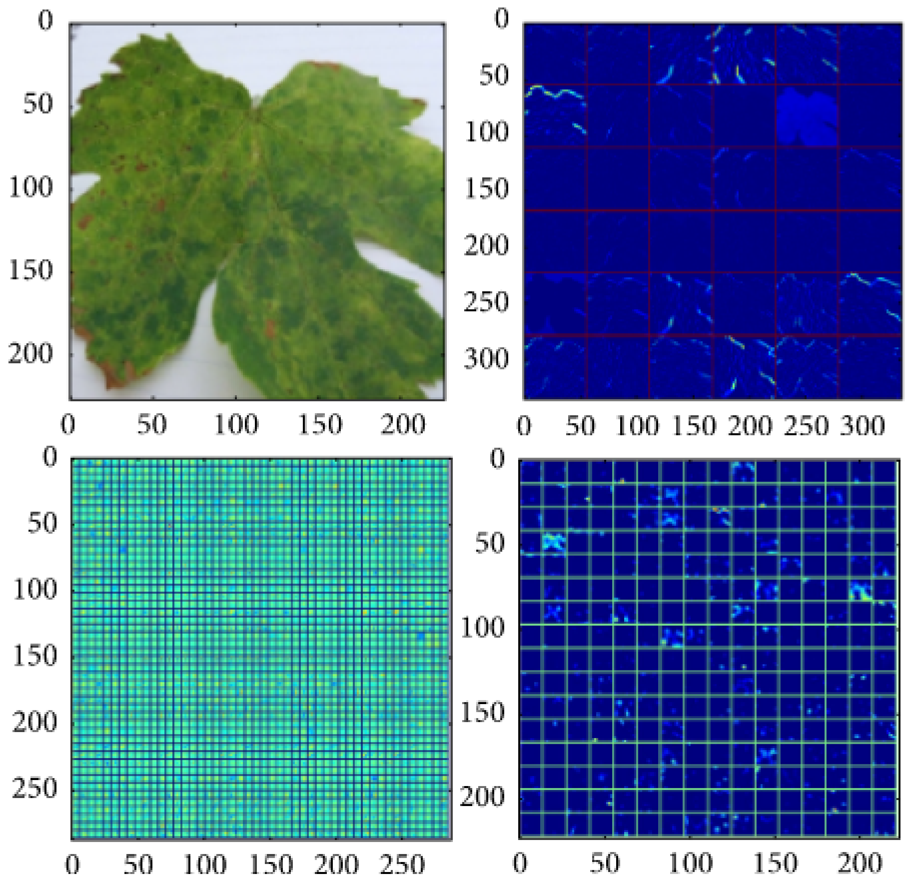


Figure 7: Pictorial representations of CNN abstractions.

$$(f^*g)(t) \triangleq \int_{-\infty}^{\infty} f(\tau)g(t-\tau)\tau'$$

$(f^*g)(t)$ = functions that are being convoluted

t = real number variable of functions f and g

$g(\tau)$ = convolution of the function f(t)

τ' = first derivative of g(tau) function

Figure 8: Convolution

In neural networks, a nxn feature detector matrix goes through the image and finds all the features that are being shared in order to provide a convolved image, reducing its size to make the code faster. This process happens multiple times through multiple images.

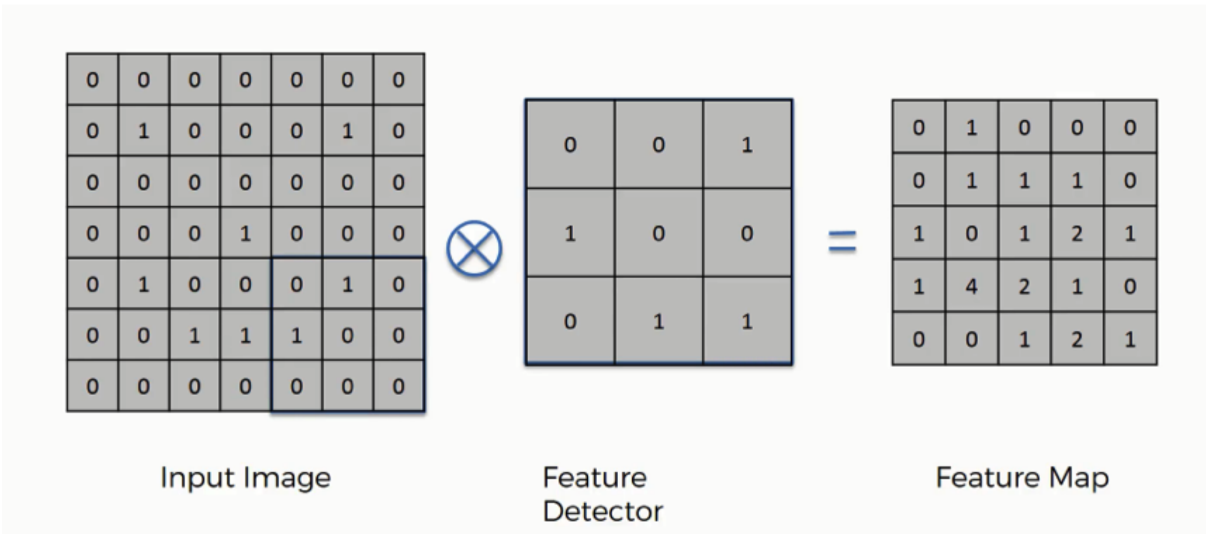


Figure 9: Feature detection

Finally, then, the image is filtered to fit into the featured version.

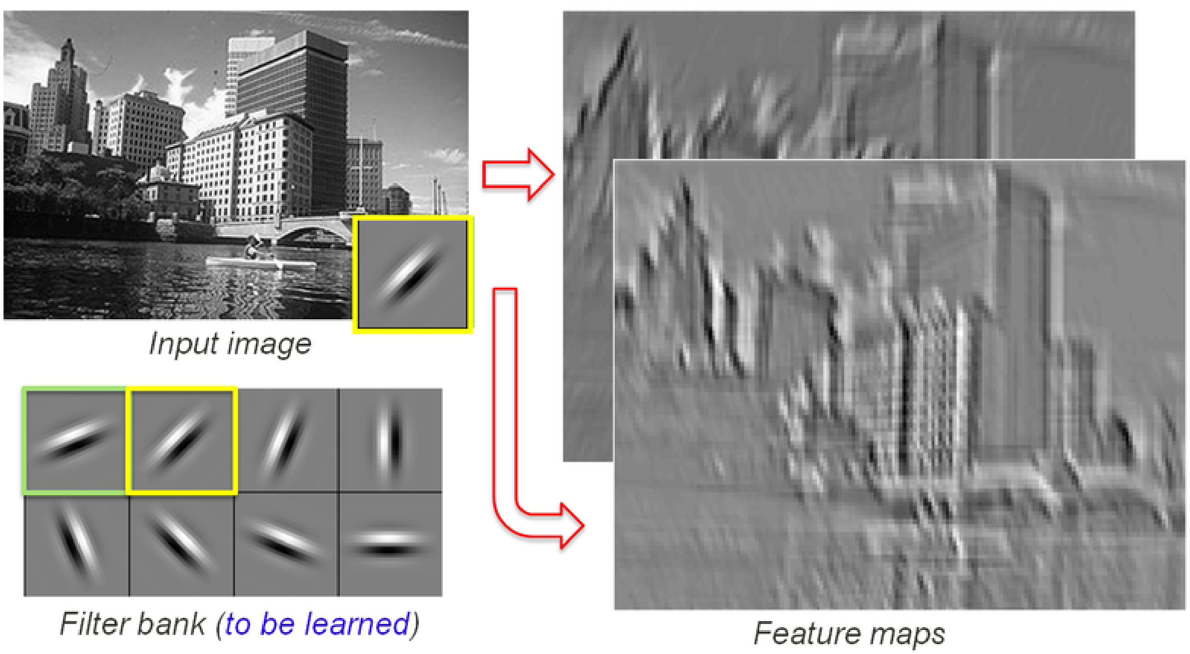


Figure 10: Feature maps

Various types of filter maps are sharpen, blur, edge detect, emboss, etc.

Since, the image recognition is a non-linear process, the CNN must reflect that. Deep CNN with ReLUs thus, introduced. It trains quicker. This strategy is applied to the yield of each convolutional and completely connected layer. Regardless of the yield, the information standardization isn't needed; it is applied after ReLU nonlinearity after the first and second convolutional layer since it diminishes top-1 and top-5 error rates.

Step 2: Max pooling

Another important layer of CNNs is the pooling layer, which is a form of nonlinear downsampling. It is required to achieve spatial invariance and higher flexibility. Pooling operation gives the form of translation invariance; it operates independently on every depth slice of the input and resizes it spatially. Overlapping pooling is beneficially applied to lessen overfitting. Also in favour of reducing overfitting, a dropout layer is used in the first two fully connected layers. But the shortcoming of dropout is that it increases training time 2-3 times compared to a standard neural network of the exact architecture. Optimization experiments also proved that ReLUs and dropout have synergy effects, which means that it is advantageous when they are used together .

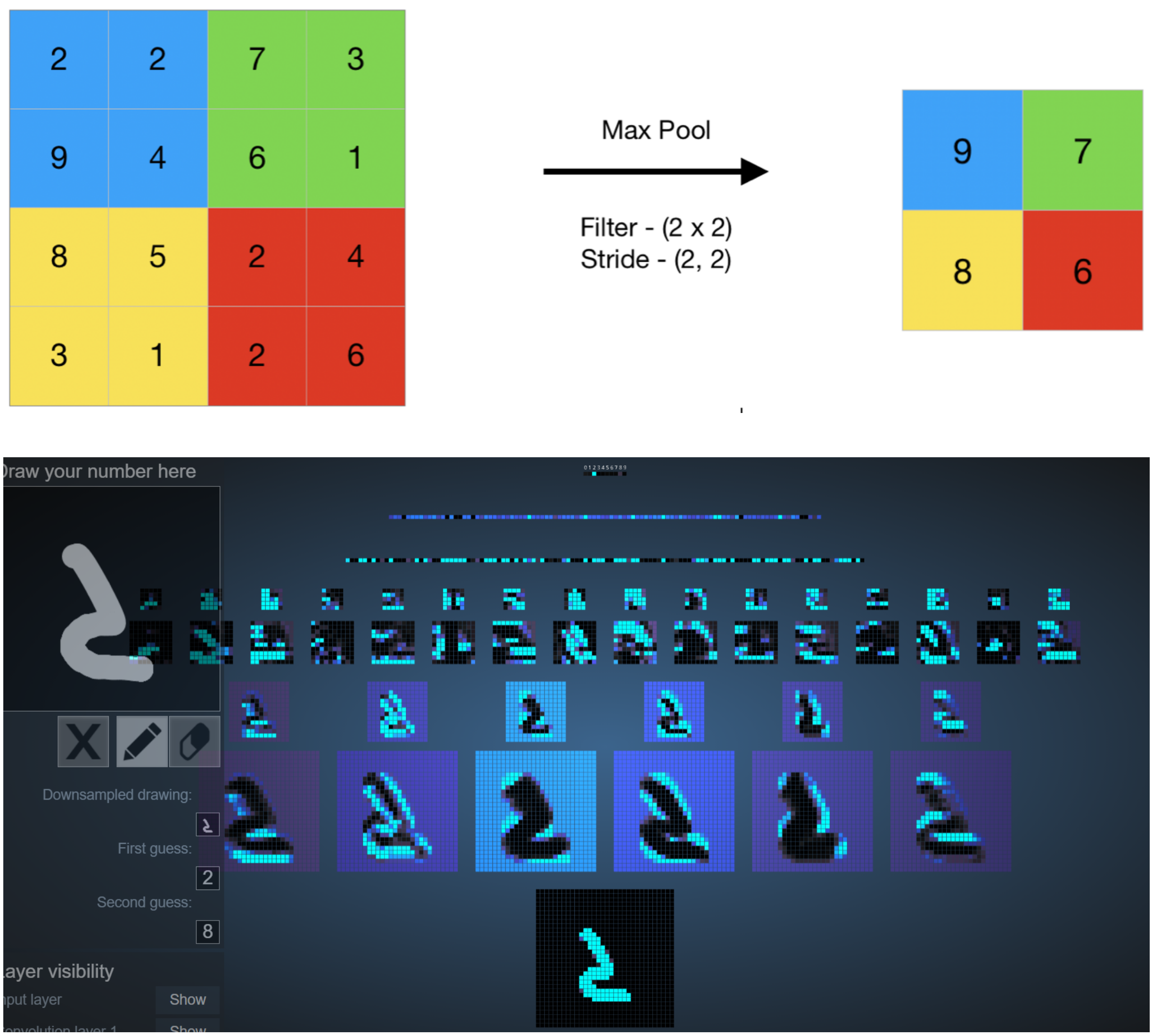


Figure 11: representation of CNN abstractions in pooling

Step 3: Flattening

Flattening is a simple step where the pooled feature map and put it into a unidimensional, if large, vector.

Step 4: Full connection:

This is the final step where the whole connection is put together to form a deep CNN. This CNN has initially randomized weights that, with time, adjust their weights through repeated backpropagation.

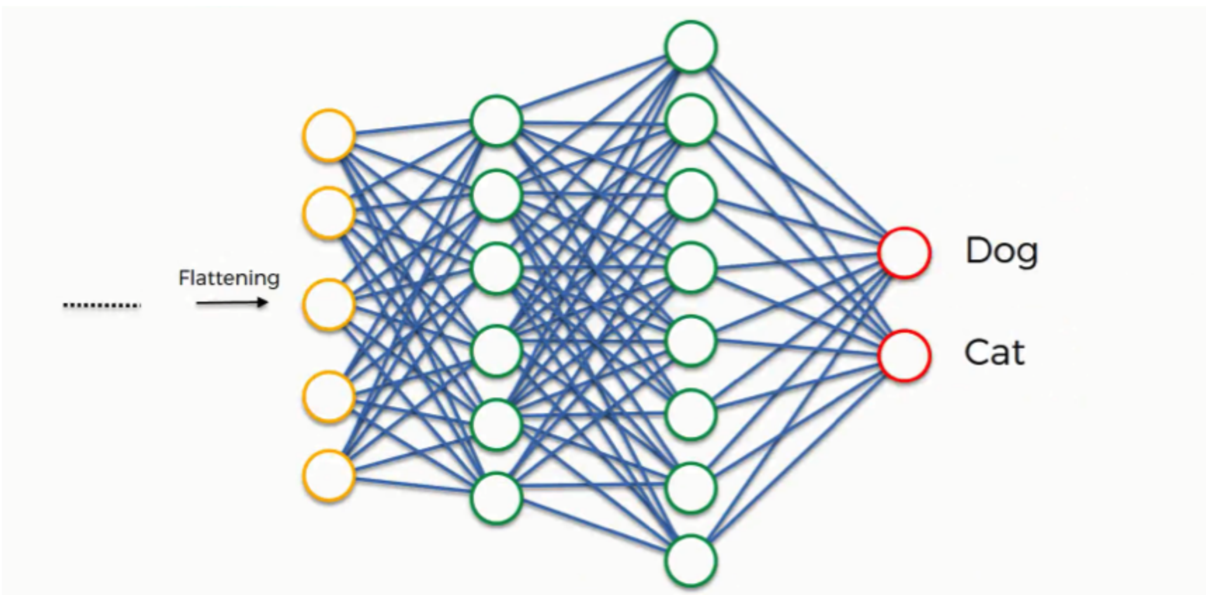


Figure 12: An example of a fully-connected CNN

Thus, in summation, the input image gets applied with multiple feature detectors to create feature maps. This comprises a convolution layer which gets applied to the Relu function to increase non-linearity in images. Then, the pooling layer gets applied to the convolutional layer. Then the pooling layer is there to avoid overfitting which is then flattened and fully-connected. Finally all of this is forward and back-propagated to adjust the weights through multiple epochs.

Part 2 - Building the CNN

```
In [55]:
Cnn = tf.keras.models.Sequential([
  layers.BatchNormalization(),
  layers.Conv2D(32, 3, activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(64, 3, activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(128, 3, activation='relu'),
  layers.MaxPooling2D(),
  layers.Flatten(),
  layers.Dense(256, activation='relu'),
  layers.Dense(38, activation= 'softmax')
])
```

```
In [56]: Cnn.compile(optimizer='adam',loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

Training the CNN

```
In [58]: retVal = Cnn.fit(training_ds,validation_data= validation_ds,epochs = 5)

Epoch 1/5
703/703 [==============================] - 4971s 7s/step - loss: 1.0789 - accuracy: 0.6931 - val_loss: 0.5585 - val_accuracy:
0.8272
Epoch 2/5
703/703 [==============================] - 4581s 7s/step - loss: 0.3720 - accuracy: 0.8821 - val_loss: 0.3083 - val_accuracy:
0.9013
Epoch 3/5
703/703 [==============================] - 3701s 5s/step - loss: 0.1979 - accuracy: 0.9357 - val_loss: 0.2715 - val_accuracy:
0.9186
Epoch 4/5
703/703 [==============================] - 3746s 5s/step - loss: 0.1157 - accuracy: 0.9607 - val_loss: 0.2737 - val_accuracy:
0.9225
Epoch 5/5
703/703 [==============================] - 3719s 5s/step - loss: 0.0882 - accuracy: 0.9705 - val_loss: 0.3291 - val_accuracy:
0.9109
```

Figure 13: Code snippet of creating a CNN

4. EXPERIMENTAL RESULTS

4.1 Environment Setup/Tools/Simulator

System Configuration:

The models were developed on system with the following configuration:

- CPU- i7 10^{th} generation.
- GPU – Nvidia RTX 2060 (mobile) with 6GB if VRAM
- RAM- 16GB
- ROM- 1 TB HDD
- OS- Windows 10 home 64-bit.

Environment Setup:

- A Tool called Anaconda is used to setup a virtual environment. This separated all other packages from other python libraries by creating a virtual Container.
- It is 64-bit distribution.

Libraries used:

- numpy
- pandas
- matplotlib.pyplot
- tensorflow
- keras.preprocessing.image
- os
- time

Tools used:

- Editors used – Jupyter notebook
- Git for version Control.
- Github for keeping a remote copy of the project.
- lucidchat.com for drawing flow-charts.
- Chrome Developer Tools for checking Network requests.
- Npm – it stands for node package manager used to keep track of files.

4.2 Experiment / Test Run:

The outcomes introduced in this segment are identified with preparing with the entire information base containing both unique and increased pictures. As it is realized that convolutional networks can learn highlights when prepared on bigger datasets, results accomplished when prepared with just original pictures won't be explored

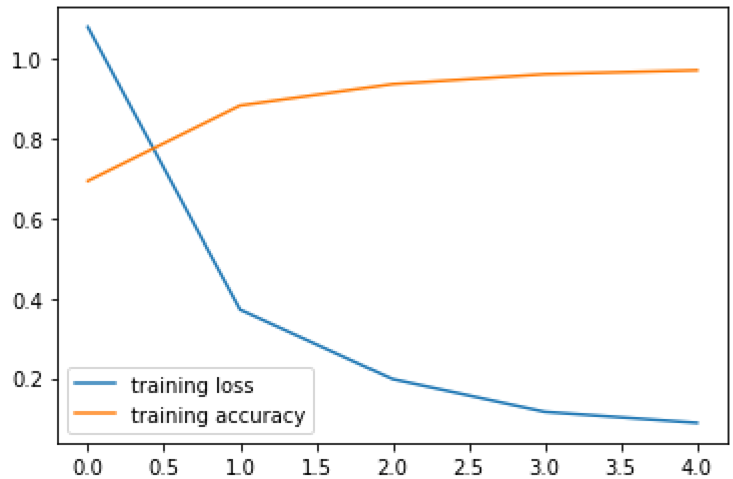
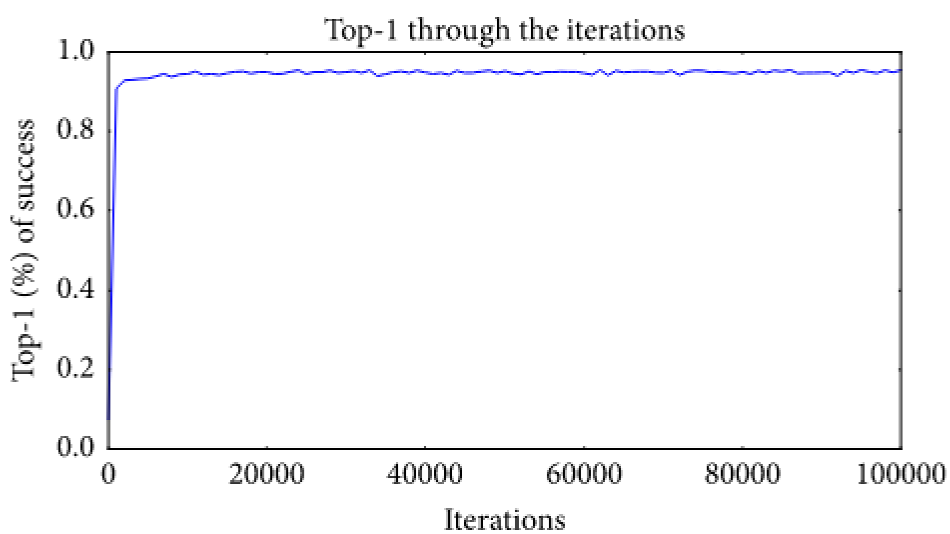


Figure 14: Baseline prediction



Top-1 accuracy success

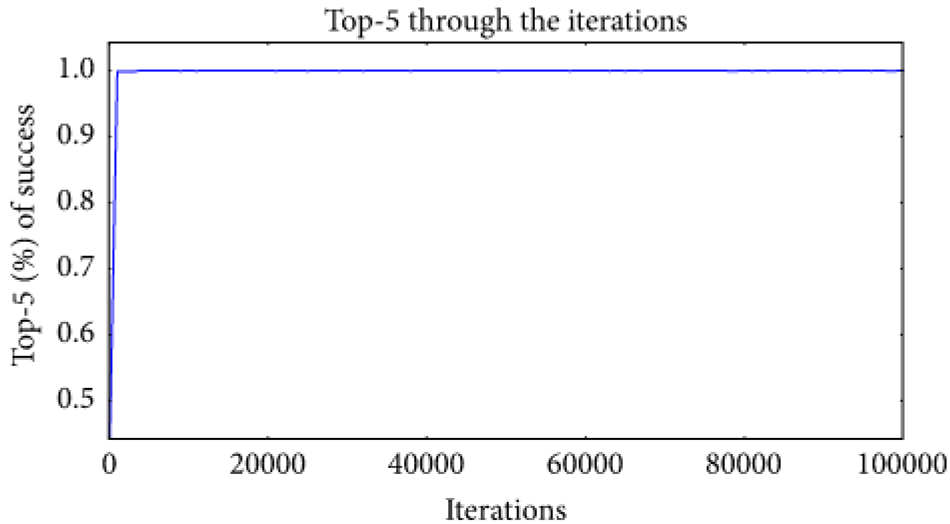


Figure 15: Estimated Top-1 and top-5 percent of success

Model summary	
Model size(MB)	316.6161422729492
Time on Validation data (sec)	268.1705623000016
Accuracy on validation data	0.9108809232711792

Table 3: Performance

5. CONCLUSION AND FUTURE WORK

There are many methods in automated or computer vision plant disease detection and classification process, but still, this research field is lacking. In addition, there are still no commercial solutions on the market, except those dealing with plant species recognition based on the leaves images.

In this paper, a new approach of using deep learning method was explored in order to automatically classify and detect plant diseases from leaf images. The developed model was able to detect leaf presence and distinguish between healthy leaves and 27 different diseases, which can be visually diagnosed. The complete procedure was described, respectively, from collecting the images used for training and validation to image preprocessing and augmentation and finally the procedure of training the deep CNN and fine-tuning. Different tests were performed in order to check the performance of newly created model.

New plant disease image database was created,\ Internet sources and extended to more than 87000 using appropriate transformations. The experimental results achieved precision of 91%, for separate class tests. The final overall accuracy of the trained model was 91.09 percent. Fine-tuning has not shown significant changes in the overall accuracy.

As the presented method has not been exploited, as far as we know, in the field of plant disease recognition, there was no comparison with related results, using the exact technique. In comparison with other techniques used and presented in Section 2, comparable or even better results were achieved, especially when taking into account the wider number of classes in the presented study.

An extension of this study will be on gathering images for enriching the database and improving accuracy of the model using different techniques of fine-tuning and augmentation.

The main goal for the future work will be developing a complete system consisting of server side components containing a trained model and an application for smart mobile devices with features such as displaying recognized diseases in fruits, vegetables, and other plants, based on leaf images captured by the mobile phone camera. This application will serve as an aid to farmers (regardless of the level of experience), enabling fast and efficient recognition of plant diseases and facilitating the decision-making process when it comes to the use of chemical pesticides.

Furthermore, future work will involve spreading the usage of the model by training it for plant disease recognition on wider land areas, combining aerial photos of orchards and vineyards captured by drones and convolution neural networks for object detection. By extending this research, the authors hope to achieve a valuable impact on sustainable development, affecting crop quality for future generations.

REFERENCES

[1] S. Sankaran, A. Mishra, R. Ehsani, and C. Davis, "A review of advanced techniques for detecting plant diseases," *Computers and Electronics in Agriculture*, vol. 72, no. 1, pp. 1–13, 2010.

[2] P. R. Reddy, S. N. Divya, and R. Vijayalakshmi, "Plant disease detection techniquetool—a theoretical approach," *International Journal of Innovative Technology and Research*, pp. 91–93, 2015.

[3] A.-K. Mahlein, E.-C. Oerke, U. Steiner, and H.-W. Dehne, "Recent advances in sensing plant diseases for precision crop protection," *European Journal of Plant Pathology*, vol. 133, no. 1, pp. 197–209, 2012.

[4] P. Chaudhary, A. K. Chaudhari, A. N. Cheeran, and S. Godara, "Color transform based approach for disease spot detection on plant leaf," *International Journal of Computer Science and Telecommunications*, vol. 3, no. 6, pp. 65–69, 2012.

[5] S. B. Patil and S. K. Bodhe, "Leaf disease severity measurement using image processing," *International Journal of Engineering and Technology*, vol. 3, no. 5, pp. 297–301, 2011.

[6] J. K. Patil and R. Kumar, "Feature extraction of diseased leaf images," *Journal of Signal & Image Processing*, vol. 3, no. 1, p. 60, 2012.