A PROJECT REPORT ON

Convolutional Neural Network and Serverless Cloud based Collaborative platform for Plant Disease Identification and Recommendation (PlantAI)

SUBMITTED TOWARDS THE
PARTIAL FULFILMENT OF THE REQUIREMENTS OF

BACHELOR OF ENGINEERING (Computer Engineering)

BY

Mahima Gaikwad

Durgesh Pachghare

Atharva Patil

Pranesh Dhake

Under The Guidance of

Prof. N. G. Sharma

Department of Computer Engineering
K. K. Wagh Institute of Engineering Education & Research
Hirabai Haridas Vidyanagari, Amrutdham, Panchavati, Nashik-422003
Savitribai Phule Pune University

A. Y. 2019-20 Sem I

K. K. Wagh Institute of Engineering Education and Research
Department of Computer Engineering

CERTIFICATE

This is to certify that the Project Titled

Convolutional Neural Network and Serverless Cloud based Collaborative platform for Plant Disease Identification and Recommendation (PlantAI)

Submitted by

Mahima Gaikwad

Durgesh Pachghare

Atharva Patil

Pranesh Dhake

is a bonafide work carried out by Students under the supervision of Prof. N. G. Sharma and it is submitted towards the partial fulfilment of the requirement of Bachelor of Engineering (Computer Engineering) Project during academic year 2019-20.

Prof. N. G. Sharma Internal Guide Department of Computer Engineering	Prof. Dr. S. S. Sane Head Department of Computer Engineering

Abstract

Plant diseases are a threat to farmers, consumers, the environment and the global economy. Almost 40% of the world's crop yield is damaged due to diseases and pest infestation. According to the survey in 2012, Maharashtra has the highest rate of farmer suicides and one of the major reasons for this is the failure of crops. The use of random pesticides and insecticides is also harmful to crops. These adverse effects on the yield can be avoided through early detection and proper expert's guidance. The proposed system is an integrated and collaborative platform with an automated image-based classification system for the identification of plant diseases and advising the preventive measures for the same. Since existing datasets focus across several countries and none focuses on Indian crops specifically, there is a need for establishing a local dataset to be of use to Indian farmers. It uses Generative Adversarial Networks (GANs) to augment the limited number of local images available. The classification is done by a Convolutional Neural Network (CNN). The AI model continuously learns from user-uploaded images suggestions given by experts to enhance its accuracy. For preventive measures, disease density maps along with spread forecasting are rendered from a Cloud-based repository of geo-tagged images and micro-climatic factors. Web Interface aims to allow experts to perform disease analytic with geographical visualizations. Some pesticides reduce soil fertility if applied frequently. So, with the help of input from the user about previous medication, appropriate recommendations are aimed to be provided to the user for proper medication with retaining maximum soil fertility

Acknowledgments

It gives us great pleasure in presenting the preliminary project report on ***'CONVOLUTIONAL NEURAL NETWORK AND SERVERLESS CLOUD BASED COLLABORATIVE PLATFORM FOR PLANT DISEASE IDENTIFICATION AND RECOMMENDATION (PlantAI)'****.*

We would like to take this opportunity to thank our internal guide ***Prof. N. G. Sharma*** *for giving us all the help and guidance we needed. We are really grateful to them for her kind support. Her valuable suggestions were very helpful.*

We are also grateful to ***Prof. S.S Sane****, Head of Computer Engineering Department,K.K.Wagh Institute Of Engineering Education And Research, Nashik for his indispensable support, suggestions.*

In the end our special thanks to ***Mr. Y. B. Nawale*** *for providing various resources such as laboratory with all needed software platforms, continuous Internet connection, for our Project.*

Mahima Gaikwad
Durgesh Pachghare
Atharva Patil
Pranesh Dhake
(B.E. Computer Engg.)

INDEX

1 Introduction **1**

1.1 Motivation of the project 2

1.2 Project Idea 2

1.3 Literature Survey 4

2 Problem Definition and scope **9**

2.1 Problem Statement 10

2.1.1 Goals and objectives 10

2.1.2 Statement of scope 10

2.2 Major Constraints 11

2.3 Methodologies of Problem solving 11

2.4 Scenario in which multi-core, Embedded and Distributed Computing used 13

2.5 Hardware Resources Required 13

2.6 Software Resources Required 14

3 Project Plan **15**

3.1 Project Estimates 16

3.1.1 Reconciled Estimates 16

3.1.2 Project Resources 18

3.2 Risk Management 18

3.2.1 Risk Identification 18

3.2.2 Risk Analysis 18

3.2.3 Risk Mitigation 20

3.3 Project Schedule 21
3.3.1 Project Timeline Chart 21
3.4 Team Organization 23
3.4.1 Team structure 23
3.4.2 Management reporting and communication 23

4 Software requirement specification **24**
4.1 Introduction 25
4.1.1 Purpose and Scope of Document 25
4.2 Usage Scenario 25
4.2.1 Use-cases 25
4.2.2 Use Case View 26
4.3 Data Model and Description 27
4.3.1 Datasets 27
4.3.2 Data Augmentation 29
4.4 Functional Model and Description 30
4.4.1 Data Flow Diagram 30
4.4.2 Activity Diagram 32
4.4.3 Sequence Diagram 33
4.4.4 User Interaction Diagram 34
4.4.5 Process Diagram 35

5 Detailed Design Document **36**
5.1 Architectural Design 37
5.1.1 System Architecture 37
5.1.2 Deployment Architecture 39
5.1.3 CNN Architecture 40

6 Summary and Conclusion **43**
6.1 Summary 44
6.2 Conclusion 44

7 Mathematical Model **48**

8 Plagiarism Report **52**

Annexure A Sponsorship detail (if any) **65**

List of Figures

1.1 Summary of statistical approaches for Plant Disease Identification and Classification 4

1.2 Summary of statistical approaches for Plant Disease Identification and Classification 5

1.3 Imaging Techniques successfully investigated for several crops . . . 6

1.4 Comparison Matrices of all five CNN architectures 7

1.5 No of Epoch with accuracy and validation accuracy w.r.t. 5,4,3 CNN layer 8

1.6 Overall Accuracy Comparison 8

3.1 Project Execution Plan 17

3.2 Project Outline 22

4.1 Use case diagram 26

4.2 PlantDisease Dataset: images and labeled objects per class (marked classes also exist in the PlantVillage dataset) 28

4.3 Data Flow Diagram 30

4.4 Activity Diagram 32

4.5 Sequence Diagram 33

4.6 User Interaction Diagram 34

4.7 Process Diagram 35

5.1 Block Diagram 38

5.2 Serverless Architecture 39

5.3 CNN Architecture 41

5.4 CNN block diagram 42

8.1 Plagiarism Report for Abstract 53
8.2 Plagiarism Report for Introduction 54
8.3 Plagiarism Report for Problem Definition and scope : Part 1 55
8.4 Plagiarism Report for Problem Definition and scope : Part 2 56
8.5 Plagiarism Report for Project Plan 57
8.6 Plagiarism Report for Software Requirements and Specification . . 58
8.7 Plagiarism Report for Software Requirements and Specification . . 59
8.8 Plagiarism Report for Software Requirements and Specification . . 60
8.9 Plagiarism Report for Detailed Design Document 61
8.10 Plagiarism Report for Detailed Design Document 62
8.11 Plagiarism Report for Detailed Design Document 63
8.12 Plagiarism Report for Summary and Conclusion 64

List of Tables

2.1 Hardware Requirements 13

3.1 Risk Table 19

3.2 Risk Probability definitions 19

3.3 Risk Impact definitions 19

4.1 Use Cases 25

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION OF THE PROJECT

- While India has progressed in a number of different sectors, the development of agriculture in India has not grown in parallel with the new available technologies.
- Crop diseases adversely affect the agricultural yield, it is even causing famine and socio-economic distress. India has the highest number of Farmer's suicidal cases.
- In developing countries, the majority of agricultural produce comes from small scale farmers and major loss is reported due to pests and diseases.
- Timely and accurate identification of crop condition is vital for implementing appropriate measures immediately, specifically where certain diseases have no treatment or spread rapidly
- Diagnosis through visual examination could be time consuming and error prone while existing lab based diagnosis methods could turn out too expensive.
- There is a need for a system where farmers can identify crop diseases at an early stage and perform proper medications with the help of computer algorithms. There should be a collaborative platform that connects experts and farmers for better crop yield.

1.2 PROJECT IDEA

- An Integrated and collaborative platform with an automated image-based classification system for the identification of plant diseases and advising the preventive measures. Farmers can use a mobile application to capture the image of an infected plant or leaf and receive user-friendly recommendations.
- For Preventive measures, disease density maps with spreading forecasting are rendered from a Cloud-based repository of geo-tagged images and micro-climatic factors.

- To improvise a system which classifies and detects disease of a plant from an image given by an mobile app or web portal and give proper user friendly assistance to the user.

- To create cloud based repository of geo-location tagged images uploaded by farmers with micro-climatic factors and update it with new entries which helps for analytics.

- To use a web scrapper to find plant images and create local database of plant images using Generative Adversarial Networks (GANs) to augment the limited number of local images available.

- To create a web interface allows experts to perform disease analytics with geographical visualizations with the help of disease density maps with spread forecasting which to be rendered from that Cloud based repository.

1.3 LITERATURE SURVEY

Publication and Year	Statistical Approach	Crop	Disease
Ferentinos (2018)	AlexNetOWTBn, VGG	25 Plant Species	58 different plant-disease pair
Wang, Sun, & Wang (2017)	Deep Learning (VGG16, VGG19, Inception-v3, ResNet50)	Apple	Apple Black Rot
Brahimi, Boukhalfa, & Moussaoui (2017)	Deep Learning: AlexNet, GoogleNet	Tomato	Spider Mites, Septoria Spot, Late blight, Early blight, Bacterial Spot, Tomato Yellow Leaf Curl Virus, Leaf Mold, Tomato Mosaic Virus, Target Spot.
Durmus, Günes, & Kirci (2017)	Deep Learning: AlexNet, SqueezeNet	Tomato	Spider Mites, Septoria Spot, Late blight, Early blight, Bacterial Spot, Tomato Yellow Leaf Curl Virus, Leaf Mold, Tomato Mosaic Virus, Target Spot.
Lu, Zhou, Gao, & Jiang (2017)	ROC curve analysis	Tomato	Yellow Leaf Curl Disease
Mohanty, Hughes, & Salathé (2016)	Deep Learning Models (GoogleNet and AlexNet)	14 crops	26 diseases
Sabrol & Kumar (2016)	Decision Trees	Tomato	Late Blight, Bacterial Leaf Spot, Septoria Leaf, Leaf Curl, Bacterial Canker
Zhao et al. (2016)	Partial Least Square Regression (PLSR)	Cucumber	Angular Leaf Spot (ALS) disease
Singh & Misra (2016)	SVM classifier + Minimum Distance Criterion+ K-means	Rose, beans leaf , lemon leaf, banana leaf	Banana leaf: Early scorch disease; Lemon leaf: Sun burn disease; Beans & Rose: Bacterial disease; Beans leaf: Fungal disease.
Chung et al. (2016)	SVM	Wheat	Bakanae disease
Sabrol & Kumar (2016)	Fuzzy Inference System (FIS), Adaptive neuro-fuzzy inference system, Multi-layer FFBPNN	Tomato	Late Blight, Bacterial Leaf Spot, Tomato Leaf Curl, Septoria Leaf Spot, Bacterial Canker.
Rupanagudi, Ranjani, Nagaraj, Bhat, & Thippeswamy (2015)	K-means Clustering and PCA	Tomato	Borer Insects

Figure 1.1: Summary of statistical approaches for Plant Disease Identification and Classification

Raza, Prince, Clarkson, & Rajpoot (2015)	SVM classifier with linear kernel	Tomato	Powdery Mildew
Xie, Shao, Li, & He (2015)	Extreme Learning Machine (ELM) classifier model	Tomato	Early blight, Late blight
Liu, Zhang, Shu, & Jin (2013)	RBF SVM	Wheat	Stripe rust, powdery mildew, leaf rust, leaf blight
Arivazhagan, Shebiah, Ananthi, & Varthini (2013)	MDC, SVM	Tomato	Bacterial leaf spot, Leaf spot
Sankaran, Mishra, Ehsani, & Davis (2011)	LCA, QDA, K-nearest neighbor, SIMCA	Citrus Orchards	Huanglongbing (HLB)
Al-Bashish, Braik, & Bani-Ahmad (2011)	K-means clustering and FFBPNN	General leaf images	Late Scorch, Tiny Whiteness, Cottony Mold, Ashen Mold, Early Scorch.
Rumpf et al. (2010)	Decision Trees, ANN, SVM+RBF	Sugar Beet	Powdery mildew, sugar Beet rust, cercospora leaf spot,
Singh, Jayasa, & Paliwala (2010)	Linear, Quadratic, Mahalanobis and a BPNN classifier	Wheat Kernels	Grain Borer, Rusty Grain Beetle, Rice Weevil, Red Flour Beetle.
Ghaffari et al. (2010)	**Clustering:** K-Means clustering, PCA and FCM **For classification:** MLP, LVQ and RBF based ANNs	Tomato	Spider mite infected plants and Powdery mildew (Oidium lycopersicum)
Camargo & Smith (2009)	SVM	Cotton	Southern Green Stink Bug, Bacterial Angular, Ascochyta Blight
Wang , Zhang , Zhu, & Geng (2008)	BPNN with gradient descent	Tomato	Late Blight
Kuo-Yi Huang (2007)	BPNN	Phalaenopsis	Phalaenopsis seedling disease
Xu, Zhu, Ying, & Jiang (2006)	DPLS, MD.	Tomato	Tomato mosaic virus

Figure 1.2: Summary of statistical approaches for Plant Disease Identification and Classification

Imaging Technique	Crop	Publication and Year
RGB Imaging	Apple, Tomato, PlantVillage Dataset (Blueberry, Potato, Raspberry, Strawberry, Cherry, Corn, Grape, Orange, Peach, Pepper, Soybean), Rose, beans leaf , lemon leaf, banana leaf, Wheat, Cotton, Phalaenopsis, Wheat Kernels, Watermelon, Pumpkin, Bell, Peach.	Ferentinos (2018), Wang, Sun, & Wang (2017), Brahimi, Boukhalfa, & Moussaoui (2017), Durmus, Günes, & Kirci (2017), Mohanty, Hughes, & Salathé (2016), Sabrol & Kumar (2016), Singh & Misra (2016), Chung et al. (2016), Raza, Prince, Clarkson, & Rajpoot (2015), Liu, Zhang, Shu, & Jin (2013), Al-Bashish, Braik, & Bani-Ahmad (2011), Camargo & Smith (2009), Kuo-Yi Huang (2007), Singh, Jayasa, & Paliwala (2010)
Hyperspectral Reflectance	Cucumber, Tomato, Sugar Beet, Wheat Kernels	Zhao et al. (2016), Xie, Shao, Li, & He (2015), Rumpf et al. (2010), Singh, Jayasa, & Paliwala (2010), Wang , Zhang , Zhu, & Geng (2008), Zhang, Qin, & Liu (2005), Zhang, Qin, Liu, & Ustin (2003)
Thermal & Stereo Visible Light Imaging	Tomato	Raza, Prince, Clarkson, & Rajpoot (2015)
Visible-near Infrared Spectroscopy	Citrus Orchards	Sankaran, Mishra, Ehsani, & Davis (2011), Xu, Ying, Fu, & Zhu (2007), Xu, Zhu, Ying, & Jiang (2006)

Figure 1.3: Imaging Techniques successfully investigated for several crops

Inception ResNet V2	Target_Spot	Late_blight	Tomato_mosaic_virus	Leaf_Mold	Bacterial_spot	Early_blight	Healthy
Target_Spot	**361**	0	1	2	4	6	5
Late_blight	12	**523**	0	9	3	13	1
Tomato_mosaic_virus	5	0	**81**	3	0	1	0
Leaf_Mold	14	2	2	**234**	2	3	0
Bacterial_spot	8	3	0	0	**598**	4	0
Early_blight	26	28	1	6	13	**230**	2
Healthy	15	1	0	0	0	1	**445**
Tomato_Yellow_Leaf_Curl_Virus	8	1	0	2	3	3	0
Two-spotted_spider_mite	26	2	0	2	1	2	5
Septoria_leaf_spot	25	5	5	11	20	3	0

ResNet-50	Target_Spot	Late_blight	Tomato_mosaic_virus	Leaf_Mold	Bacterial_spot	Early_blight	Healthy
Target_Spot	**377**	3	1	2	0	4	5
Late_blight	3	**549**	0	7	1	10	1
Tomato_mosaic_virus	1	0	**92**	1	0	0	0
Leaf_Mold	3	2	0	**250**	1	2	0
Bacterial_spot	1	3	0	0	**617**	1	0
Early_blight	13	25	2	2	7	**277**	0
Healthy	5	1	0	0	0	0	**457**
Tomato_Yellow_Leaf_Curl_Virus	3	2	1	1	4	0	0
Two-spotted_spider_mite	22	0	2	0	1	1	1
Septoria_leaf_spot	7	2	3	1	4	4	0

VGG-16	Target_Spot	Late_blight	Tomato_mosaic_virus	Leaf_Mold	Bacterial_spot	Early_blight	Healthy
Target_Spot	**376**	2	1	1	1	1	4
Late_blight	21	**521**	0	3	1	20	1
Tomato_mosaic_virus	4	0	**93**	0	0	0	0
Leaf_Mold	20	7	4	**212**	0	4	0
Bacterial_spot	14	5	0	0	**585**	8	0
Early_blight	31	31	0	0	9	**252**	0
Healthy	21	2	1	0	0	0	**438**
Tomato_Yellow_Leaf_Curl_Virus	13	0	1	1	15	4	0
Two-spotted_spider_mite	44	3	2	1	0	2	2
Septoria_leaf_spot	28	4	3	0	4	3	2

VGG-19	Target_Spot	Late_blight	Tomato_mosaic_virus	Leaf_Mold	Bacterial_spot	Early_blight	Healthy
Target_Spot	**346**	3	1	3	3	11	9
Late_blight	7	**535**	0	11	0	14	0
Tomato_mosaic_virus	1	0	**91**	3	0	0	1
Leaf_Mold	2	8	2	**234**	1	5	0
Bacterial_spot	1	8	0	0	**595**	7	0
Early_blight	9	31	1	6	12	**257**	0
Healthy	11	2	1	0	0	1	**445**
Tomato_Yellow_Leaf_Curl_Virus	0	4	3	4	8	3	0
Two-spotted_spider_mite	21	2	1	2	1	8	3
Septoria_leaf_spot	5	10	3	11	3	6	2

Xception	Target_Spot	Late_blight	Tomato_mosaic_virus	Leaf_Mold	Bacterial_spot	Early_blight	Healthy
Target_Spot	**367**	2	0	0	1	5	7
Late_blight	20	**518**	0	9	1	13	0
Tomato_mosaic_virus	7	0	**81**	4	0	1	0
Leaf_Mold	11	9	1	**224**	2	4	0
Bacterial_spot	7	5	0	0	**602**	3	0
Early_blight	30	35	0	4	10	**237**	2
Healthy	14	3	0	0	0	2	**442**
Tomato_Yellow_Leaf_Curl_Virus	6	4	0	1	7	4	0
Two-spotted_spider_mite	42	1	0	1	2	0	5
Septoria_leaf_spot	23	4	5	5	8	11	2

Figure 1.4: Comparison Matrices of all five CNN architectures

No of Epoch	Five layer CNN		Four Layer CNN		Three Layer CNN	
	Accuracy	*Validation Accuracy*	*Accuracy*	*Validation Accuracy*	*Accuracy*	*Validation Accuracy*
1	0.5613	0.5920	0.6633	0.6520	0.5064	0.5083
2	0.7365	0.7170	0.7337	0.7477	0.6543	0.6310
3	0.8167	0.8033	0.8291	0.8120	0.7077	0.7127
4	0.8155	0.7997	0.8370	0.8233	0.7342	0.7070
5	0.8646	0.8300	0.8509	0.8333	0.7712	0.7487
6	0.8782	0.8453	0.8720	0.8413	0.8032	0.7860
7	0.8754	0.8670	0.9081	0.8587	0.8092	0.7583
8	0.9129	0.8797	0.9072	0.8487	0.8315	0.7963
9	0.9238	0.8647	0.8879	0.8680	0.8205	0.8153
10	0.9117	0.8640	0.9214	0.8503	0.8269	0.8230
11	0.9245	0.8780	0.9271	0.8507	0.8410	0.8260
12	0.9341	0.8710	0.9116	0.8453	0.8581	0.8200
13	0.9257	0.8950	0.9348	0.8507	0.8499	0.8310
14	0.9417	0.8740	0.9160	0.8700	0.8735	0.8453
15	0.9375	0.8763	0.9241	0.8713	0.8753	0.8287
16	0.9341	0.8797	0.9373	0.8817	0.8624	0.8173
17	0.9387	0.8870	0.9279	0.8603	0.8824	0.8260
18	0.9492	0.8843	0.9464	0.8900	0.8978	0.8440
19	0.9524	0.8850	0.9531	0.8673	0.8940	0.8447
20	0.9402	0.8967	0.9372	0.8643	0.8862	0.8477

Figure 1.5: No of Epoch with accuracy and validation accuracy w.r.t. 5,4,3 CNN layer

	3-layers		4-layers		5-layers	
Total Epoch	*Accur Acy*	*Validation*	*Accuracy*	*Validation*	*Accuracy*	*Validation*
10	88.24%	83.23%	92.19%	87.17%	91.01%	86.93%
15	91.11%	82.97%	92.35%	86.77%	95.05%	86.30%
20	88.62%	84.77%	93.72%	86.43%	94.02%	89.67%

Figure 1.6: Overall Accuracy Comparison

CHAPTER 2

PROBLEM DEFINITION AND SCOPE

2.1 PROBLEM STATEMENT

- To implement Convolutional Neural Networks on serverless platform for improvising a system which identifies and detects the plant disease input image of crop and recommended and assist appropriate medications alongside giving all data in visualized form to experts for analysis.

2.1.1 Goals and objectives

- To use a web scrapper to find plant and crop images and similar datasets.
- To create a database of augmented plant images using local images available which focuses on Indian crops.
- To standardize all the datasets retrieved in one form and preprocess it. (Same extension, same resolution, remove inappropriate data)
- To deploy the CNN model efficiently on Serverless platform. Build APIs and Actions for the same.
- To reduce the cost of computations with the help of Serverless features.
- To securely retrieve images of infected plant or leaf to the cloud and call appropriate APIs for triggering the CNN.
- To accurately detect the disease of the plant from the input image and give user-friendly assistance regarding the disease.
- To achieve high accuracy and increase it with the help of experts recommendations.
- To provide disease density maps with spreading forecasting from cloud-based repository of geo-tagged images and climatic factors for analysis and prediction.

2.1.2 Statement of scope

- input image size between 20kB and 20 MB.

- input image must be in JPG, JPEG, PNG format
- minimum resolution of input image must be 200 x 200
- Though it takes recommendations and answers from experts to increase the accuracy of model, the system does not connect users with experts directly.
- The disease detection does not work with rare plants of which photographs and data are rare.

2.2 MAJOR CONSTRAINTS

- The end user needs to input image through mobile application only. User's access to mobile application or web portal is must.
- Disease detection is limited to the plants or crops which data is available for training the model. Recommendations about rare plants may not get generated.

2.3 METHODOLOGIES OF PROBLEM SOLVING

Different modules associated with the system are as follows:

1. Data Preparation
2. Feature extraction
3. Training the model
4. Evaluation
5. Retraining the model
6. Mobile Application
7. Server side Application

For completion of each module, particular method is used. Task wise description of methods is:

1. Data Preparation:

- Data preprocessing: All the images in dataset are resized to 100x100 pixel format.Data is divided into two parts 80% training set, 20% test set.
- Data augmentation: Augmentation process is applied of Training set to rotate, resize and adding some random noise to images in order to avoid over fitting.

2. Feature extraction:
   Features are extracted in starting layers of CNN architecture using convolutional operation.
3. Training the model:
   System use LeNet based architecture . Once architecture is developed system train the model with Training set features.
4. Evaluation:
   Accuracy of model is evaluated with the help of Test set.
5. Retraining the model:
   - Tuning: If results are not satisfactory tune the model by changing the parameters of architecture such as kernel size, Nodes in last fully connected layer.
   - Store the weights: final model which has trained save it in model_name.h5 configuration file so that it can be used for new data.
6. Mobile Application:
   Application is developed using java for android to upload images on server, call the APIs and display the results.
7. Server Side application:
   - This application responsible for preprocessing the image uploaded by user and classify it based on its features and give the results in the form of JSON objects.
   - Update the Cloud database and other information for further analytics.

2.4 SCENARIO IN WHICH MULTI-CORE, EMBEDDED AND DISTRIBUTED COMPUTING USED

The proposed system aims to use distributed computing, as the system is deployed on serverless platform. Functional modules are deployed as separate co-dependent functions. Combined with cloud storage, data parallelism aligns with the serverless architecture naturally, in which it can launch a number of stateless functions, each accessing a different batch of data, without managing and maintaining any servers. The amount of resources invested into a training job should be dynamically adjusted as the job progresses, in the best interest of improving the model quality. Not only the model quality improve at a variable rate during training, but the dependency of model quality improvement on the amount of resources invested is also non-linear and complex.

2.5 HARDWARE RESOURCES REQUIRED

Hardware Resources Required are needed before deploying to the server i.e. if model training is done on local computer at initial stage. Since the application is hosted on ESDS eNlight Serverless platform, it is provided by ESDS.
Thus, minimum hardware requirement to train a model is given below:

Sr. No.	Parameter	Minimum Requirement	Justification
1	CPU Speed	Core i5 2.6 GHz+	For training the model
2	RAM	4 GB	To load large dataset
3	GPU	GTX 1050 2GB	To train model with GPU parallelism

Table 2.1: Hardware Requirements

2.6 SOFTWARE RESOURCES REQUIRED

Platform :

1. Operating System: Windows or Linux
2. IDE: Jupyter Notebook, IDLE
3. Programming Language: Python, SQL

CHAPTER 3

PROJECT PLAN

3.1 PROJECT ESTIMATES

3.1.1 Reconciled Estimates

3.1.1.1 Cost Estimate

The model followed is the Constructive Cost Model (COCOMO) for estimating the efforts required in the completion of the porject. Like all estimation models, the COCOMO model requires sizing information. This information can be specified in the form of:

- Object Point
- Function Point(FP)
- Lines of Source Code(KLOC)

For our project, sizing information in the form of Lines of Source Code is used. The total lines of code,

KLOC = 750

Equations: The initial effort(Ei) in man-months is calculated using equations:

$$E = ax(KLOC)^b$$

where, a = 3.0, b = 1.12, for a semi-detached project

E = Efforts in person-hours

E = 4.5 PM

$$D = ax(E)^b$$

Where, a = 2.5,

b = 0.35, for a semi-detached project

D = Duration of Project in months

D = 4 Months

3.1.1.2 Time Estimates

$$C = D * Cp * hrs$$

Where, C = Cost of project

D = Duration in Hours

Cp = Cost incurred per person-hour

hrs = hours

Total of 4.5 person-months are required to complete the project successfully.

Duration of Project D = 6 Months

The approximate duration of the project is 4 months

The project timeline is as given:

Project Outline

Phase	Action Item	Description
Technology Introduction and Training	Introduction, Training: Module 1	1. Welcoming Students 2. Group / Group hierarchy finalization 3. Discussion on Code of conduct 4. Technology overview
	Training: Module 2-4	1. Introduction to eNlight Cloud Functions 2. Training on CLI / API of fn.enlight.dev
Project Planning	Research / Literature Survey	1. Literature survey 2. Market Research 3. Technology Research
	Project Idea discussion	1. Project Idea discussion and feasibility study
	Project Idea submission	1. Submission of Idea
Project Execution	Prototype Development	1. Project Documentation 2. Development of Project Prototype 3. Unit Testing
	Prototype Demo	1. Project Demo
Project Acceptance	Project Finalization & Acceptance	1. Project finalization 2. Sponsor letter handouts

Figure 3.1: Project Execution Plan

3.1.2 Project Resources

- ESDS eNlight Serverless Platform
- ESDS Mentors
- IEEE Access Provided by Institute

3.2 RISK MANAGEMENT

3.2.1 Risk Identification

1. Datasets that primarily focuses on Indian Plants are so less.
2. Overfitting of CNN algorithm
3. In case the Servers went down
4. Poor Implementation of Serverless functions
5. Developer team does not have much knowledge about implementation on Serverless Cloud.

3.2.2 Risk Analysis

1. Datasets that primarily focuses on Indian plants are less. Those who are available on internet does not contain enough images or have inappropriate data. This results in model not getting trained well enough to classify other images.
2. Overfitting is the big risk while using CNN with less data. Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data.
3. If servers went down, whole application may get unusable. This is the risk where the entire application becomes inaccessible to users.
4. This is the risk associated with the Cloud Provider. If Cloud Provider implemented the serverless functionality poorly, this may negatively impact the performance of the application.

5. Since the project is Sponsored by ESDS, the implementation of the application on the cloud is different for that cloud. The development team does not have entire knowledge about implementation of functions on Serverless Cloud.

ID	Risk Description	Probability	Impact		
			Schedule	Quality	Overall
1	Less Dataset available	High	Low	Medium	Medium
2	Overfitting of CNN	Medium	Medium	High	Medium
3	Server Down time	Low	Low	High	Low
4	Poor implementation of Serverless	Low	Low	Low	Low
5	Developer team does not have much knowledge about implementation on Serverless Cloud	Low	Low	Low	Low

Table 3.1: Risk Table

Probability	Value	Description
High	Probability of occurrence is	$> 75\%$
Medium	Probability of occurrence is	$26 - 75\%$
Low	Probability of occurrence is	$< 25\%$

Table 3.2: Risk Probability definitions

Impact	Value	Description
Very high	$> 10\%$	Schedule impact or Unacceptable quality
High	$5 - 10\%$	Schedule impact or Some parts of the project have low quality
Medium	$< 5\%$	Schedule impact or Barely noticeable degradation in quality Low Impact on schedule or Quality can be incorporated

Table 3.3: Risk Impact definitions

3.2.3 Risk Mitigation

Risk ID	1
Risk Description	Less Dataset available that focuses primariliy on Indian plants
Category	Dataset
Source	Data Collection
Probability	High
Impact	Medium
Response	Creating more data using Data Augmentation
Strategy	Data Augmentation for variety of data
Risk Status	Identified

Risk ID	2
Risk Description	Overfitting of CNN
Category	Software
Source	CNN Risks
Probability	Medium
Impact	High
Response	Image augmentation, and more data is provided while training to prevent overfitting
Strategy	More data results in less overfitting
Risk Status	Identified

Risk ID	3
Risk Description	Server Down Time
Category	Hardware / Implementation
Source	Implementation
Probability	Low
Impact	High
Response	Having backup server provider eliminates this issue
Strategy	Host the application at more than one location
Risk Status	Identified

Risk ID	4
Risk Description	Poor Implementation of Serverless
Category	Implementation
Source	Implementation
Probability	Low
Impact	Low
Response	Co-ordinating with ESDS, better application should be developed to reduce the risk
Strategy	aquire good knowledge and implement efficiently
Risk Status	Identified

3.3 PROJECT SCHEDULE

3.3.1 Project Timeline Chart

Risk ID	5
Risk Description	Developer team does nothave much knowledge aboutimplementation on Server-less Cloud
Category	Implementation
Source	Implementation
Probability	Low
Impact	Low
Response	Co-ordinating with ESDS, and mentors. Training is scheduled
Strategy	aquire good knowledge and implement efficiently and communicate with ESDS mentors regularly
Risk Status	Identified

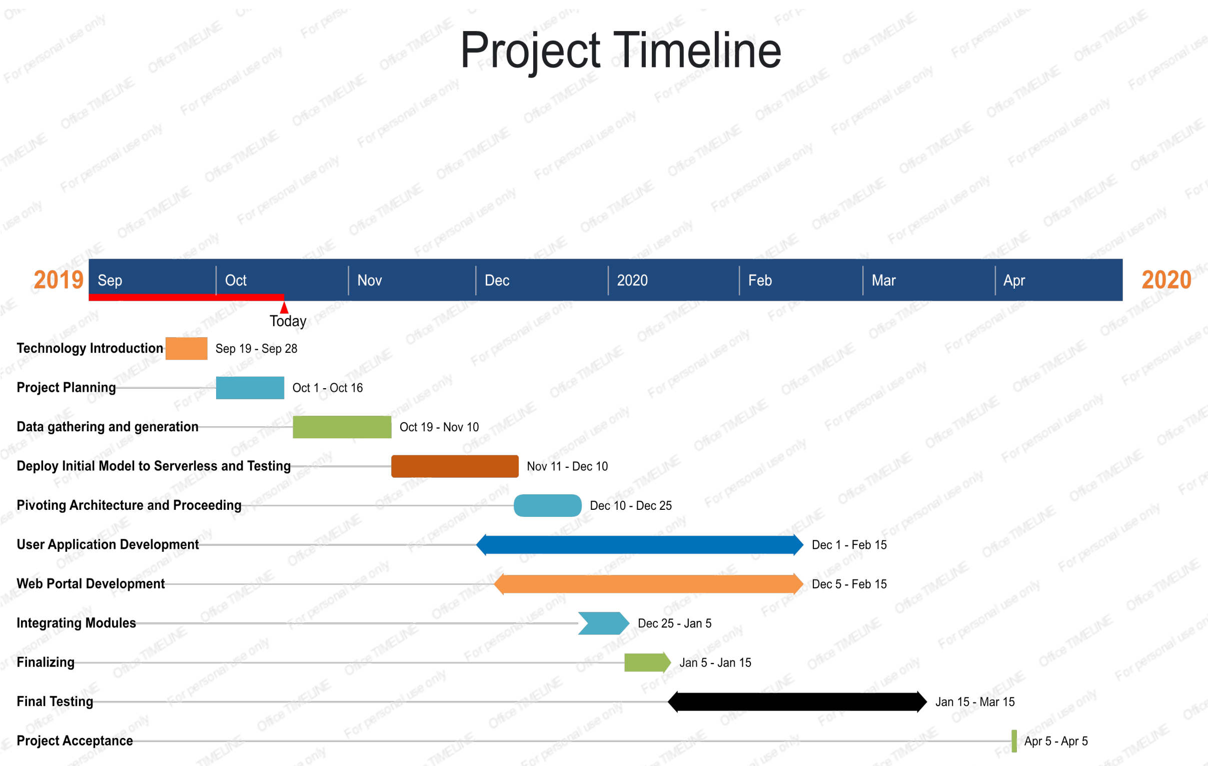

Figure 3.2: Project Outline

3.4 TEAM ORGANIZATION

- Team of 4 members
- 1 Project Guide
- 1 Project Co-ordinator

3.4.1 Team structure

A team of 4 members

3.4.2 Management reporting and communication

- Weekly Reporting to guide about the work.
- Weekly Reporting to ESDS Mentor (Sponsors)
- Project Idea Presentation (1st Sem)
- Progress Presentation (1st Sem)
- Project Execution and Demo (2nd Sem)

CHAPTER 4

SOFTWARE REQUIREMENT SPECIFICATION

4.1 INTRODUCTION

4.1.1 Purpose and Scope of Document

The purpose of the project to ease the assistance process of Plants diseases to the users with the help of CNN based classification. The system gives user friendly assistance about medication and care and it also gives visualized data about the crop diseases over large area for analytics.

4.2 USAGE SCENARIO

4.2.1 Use-cases

Sr No.	Use Case	Description	Actors	Assumptions
1	Input Image	User uploads image	User	Image is of minimum 200x200 resolution, uploaded completely in jpg/jpeg/png format
2	Disease Identification	Disease Identification on cloud	User	The photograph is clear, and model was trained enough to classify that disease
3	Getting Assistance	Providing Assistance to User	User	Proper Assistance is specified in program
4	Giving Recommendations	Recommendations about photographs classified with less accuracy	Expert	Enough Experts are available to give recommendations
5	Preventive Measures	By providing experts geographical visualizations about data	Expert	Data analyst are available

Table 4.1: Use Cases

4.2.2 Use Case View

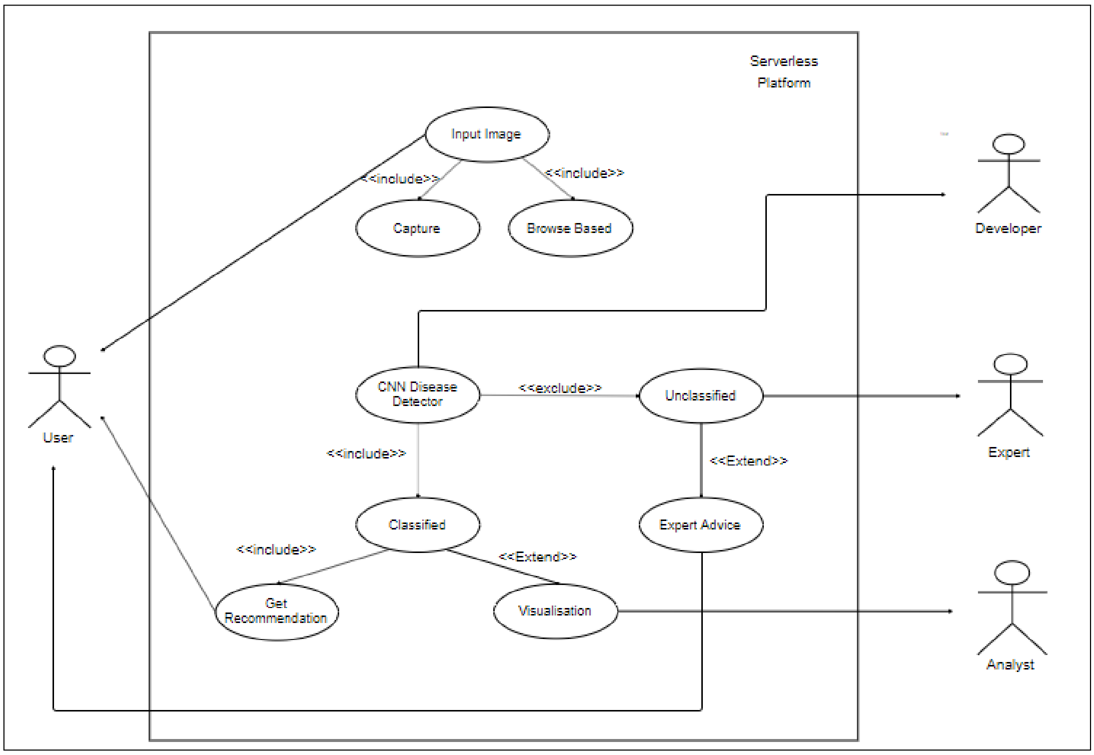

Figure 4.1: Use case diagram

4.3 DATA MODEL AND DESCRIPTION

The description about data models being used in the project is described in this section. NoSQL Database are being used to store the photographs and each record which is used for analysis purpose. Further details about datasets are and data models are given below:

4.3.1 Datasets

PlantVillage consists of 54,323 images divided into 38 classes of diseased and healthy plants based on 14 different crop species; it is available online from the crowd-sourced platform Kaggle(Supplementary Materials). All images are taken as a single leaf on a solid background which is labeled only by a class name. One of the crucial drawbacks of the present study in the area of plant disease detection is a major drop in the classification performance of the models on real images taken in fields compared to the images from a controlled environment. The reason for it is the lack of large public datasets of plant diseases and most of the present achievements are based on the PlantVillage dataset.

The Object detection algorithms were explored in the plant disease detection task. Due to this, the PlantVillage dataset was additionally labeled generating metadata holding two types of bounding boxes, one marking the entire leaf and the second one marking only the infected areas. All additional labeling was performed and verified by agricultural experts. In order to adapt the plant disease detection model for more practical usage, a new dataset was introduced, the Plant Disease dataset. New images of healthy and diseased leaves have been taken in fields under various weather conditions and at numerous angles during different daylight intensity. Alongside this, all images have an inconsistent background, meaning that sometimes a single leaf is in focus, and sometimes images hold many leaves of different crops whether they are healthy or infected with different diseases. This way, dataset consists of images that mimic practical situations where the model could be potentially used.

Class Number	Class Name	Images PlantDisease	Images (PlantVillage)	Objects
1.	*Venturia insequalis (Apple)*	1736	630	3423
2.	*Gymnosporangium juniperi-virginianae (Apple)*	2538	276	4296
3.	*Podosphaera leucotricha (Apple)*	1302		2478
4.	*Botryosphaeria obtuse (Apple)*	705	621	1207
5.	*Alternaria pomi (Apple)*	3084		5436
6.	*Malus domestica (Apple-healthy)*	2058	1645	3552
7.	*Xanthomonas euvesicatoria (Bell Pepper)*	2402		4044
8.	*Xanthomonas campestris (Bell Pepper)*	1344	997	2445
9.	*Capsicum (Bell Pepper-healthy)*	2414	1478	4443
10.	*Blumeriella jaapii (Cherry)*	1914		3480
11.	*Podosphaera* spp. *(Cherry)*	1064	1052	1851
12.	*Prunus cerasus (Cherry-healthy)*	1022	854	1863
13.	*Uncinula necator (Grape)*	1708		2811
14.	*Plasmopara viticola (Grape)*	2038		3372
15.	*Botrytis cinereal (Grape)*	2254		4560
16.	*Botryosphaeria obtuse (Grape)*	1790		3549
17.	*Pseudocercospor a vitis (Grape)*	1704	1076	3087
18.	*Guignardia bidwellii (Grape)*	1584	1180	2613
19.	*Phaeomoniella* spp. *(Grape)*	1692	1284	2739
20.	*Vitis vinifera (Grape-healthy)*	1898		3657
21.	*Peronospora destructor (Onion)*	2984		5199
22.	*Allium cepa (Onion-healthy)*	1361		2448
23.	*Claserosporium carpophilum (Peach)*	804		1401
24.	*Prunus persica (Peach-healthy)*	901	360	1831
25.	*Alternaria solani (Potato)*	2310	1000	3948
26.	*Solanum tuberosum (Potato-healthy)*	1718	152	3345
27.	*Polystigma rubrum (Plum)*	2482		4182
28.	*Plum Plox (Plum)*	1806		3474
29.	*Tranzschelia pruni-spinosae (Plum)*	1746		3228
30.	*Stigmina carpofilia (Plum)*	2192		4149
31.	*(Plum-healthy)*	2653		3423
32.	*Mycosphaerella fragariae (Strawberry)*	1242		2181
33.	*Fragaria (Strawberry-healthy)*	1686	456	2700
34.	*Cercospora beticola (Sugar beets)*	2748		4629
35.	*Beta vulgaris (Sugar beets-healthy)*	2953		4387
36.	*Phytophthora infestans (Tomato)*	2792	1910	4755
37.	*Septoria lycopersici (Tomato)*	2214	1771	3837
38.	*Solanum lycopersicum (Tomato-healthy)*	2826	1592	4683
39.	*Erysiphe graminis (Wheat)*	1566		2940
40.	*Puccinia* spp. *(Wheat)*	2036		3690
41.	*Septoria* spp. *(Wheat)*	1204		2028
42.	*Triticum sp. (Wheat-healthy)*	790		1647
	Total	**79,265**	**18,334**	**139,011**

Figure 4.2: PlantDisease Dataset: images and labeled objects per class (marked classes also exist in the PlantVillage dataset)

4.3.2 Data Augmentation

Since all the related global work has a scope with diluted focus on Indian crops and diseases, the aim here is to put local agriculture on priority. Even so, the availability of such a native image dataset for training the model is limited. A lot more (and more variable) data is required to train the model so that it achieves considerable accuracy. In this approach, Generative Adversarial Networks play an important role in achieving that. These networks are being used in augmenting the available limited data for local plants and diseases. GANs have shown promising results in generating images with significant detail.

4.4 FUNCTIONAL MODEL AND DESCRIPTION

4.4.1 Data Flow Diagram

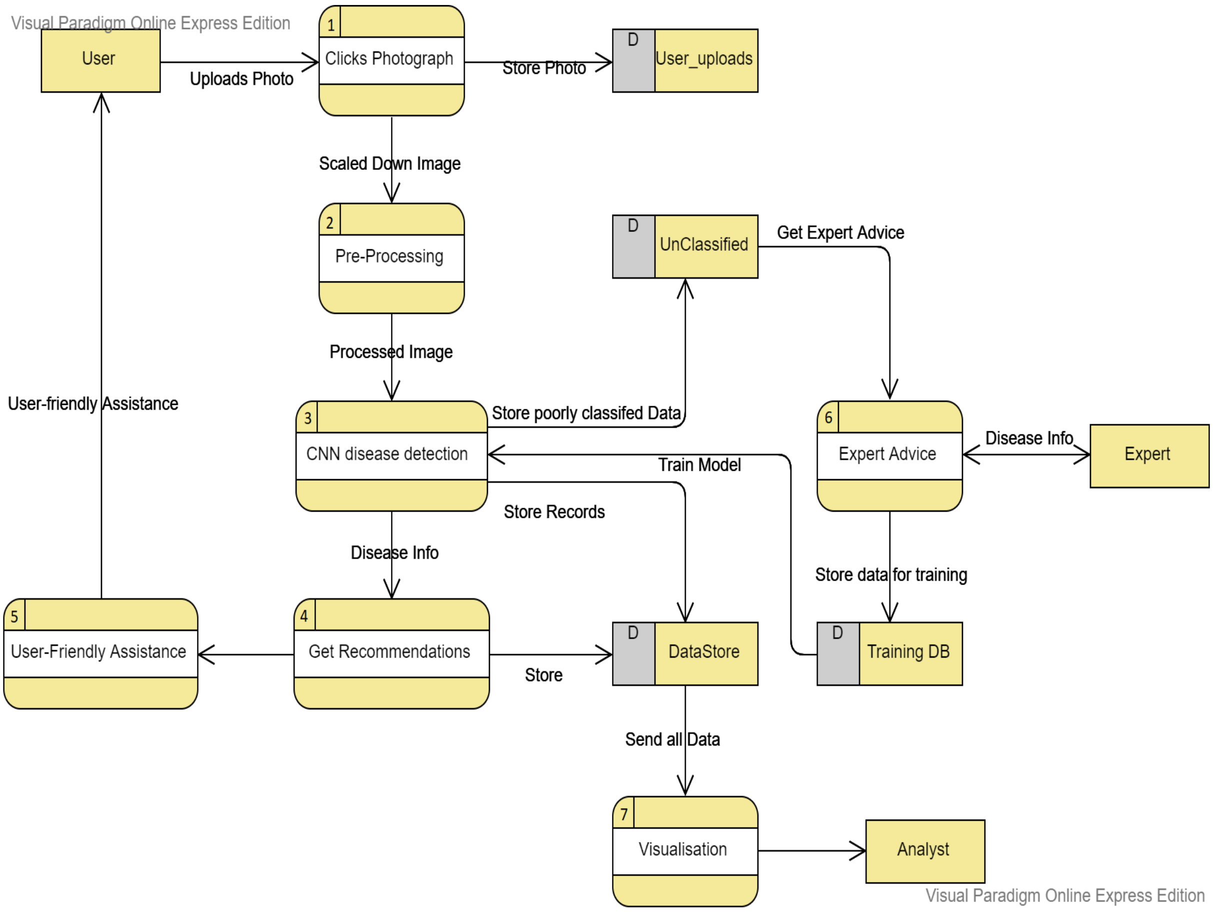

Figure 4.3: Data Flow Diagram

1. User Clicks photograph of infected plant and its leaf and through the mobile application and uploads. Then it calls APIs of Actions for further process. Data is stored on Data Storage User_uploads where all raw images that user uploads is stored.

2. The image uploaded is pre-processed, segmented for detection.

3. Then CNN disease classifier detects the plant and the disease. All the classifications less than threshold accuracy is stored in UnClassified Data Storage. All classifications greater than threshold accuracy is stored in another Data Store.

4. Recommendations are fetched about about disease detected.

5. Here, User-friendly assistance is generated. This includes translation to local language and speech for easy assistance. It then sent to User.

6. All the data from UnClassified is shown to expert for their advice. Those inputs from Experts are stored in TrainingDB which is used to tune the model again to increase the accuracy.

4.4.2 Activity Diagram

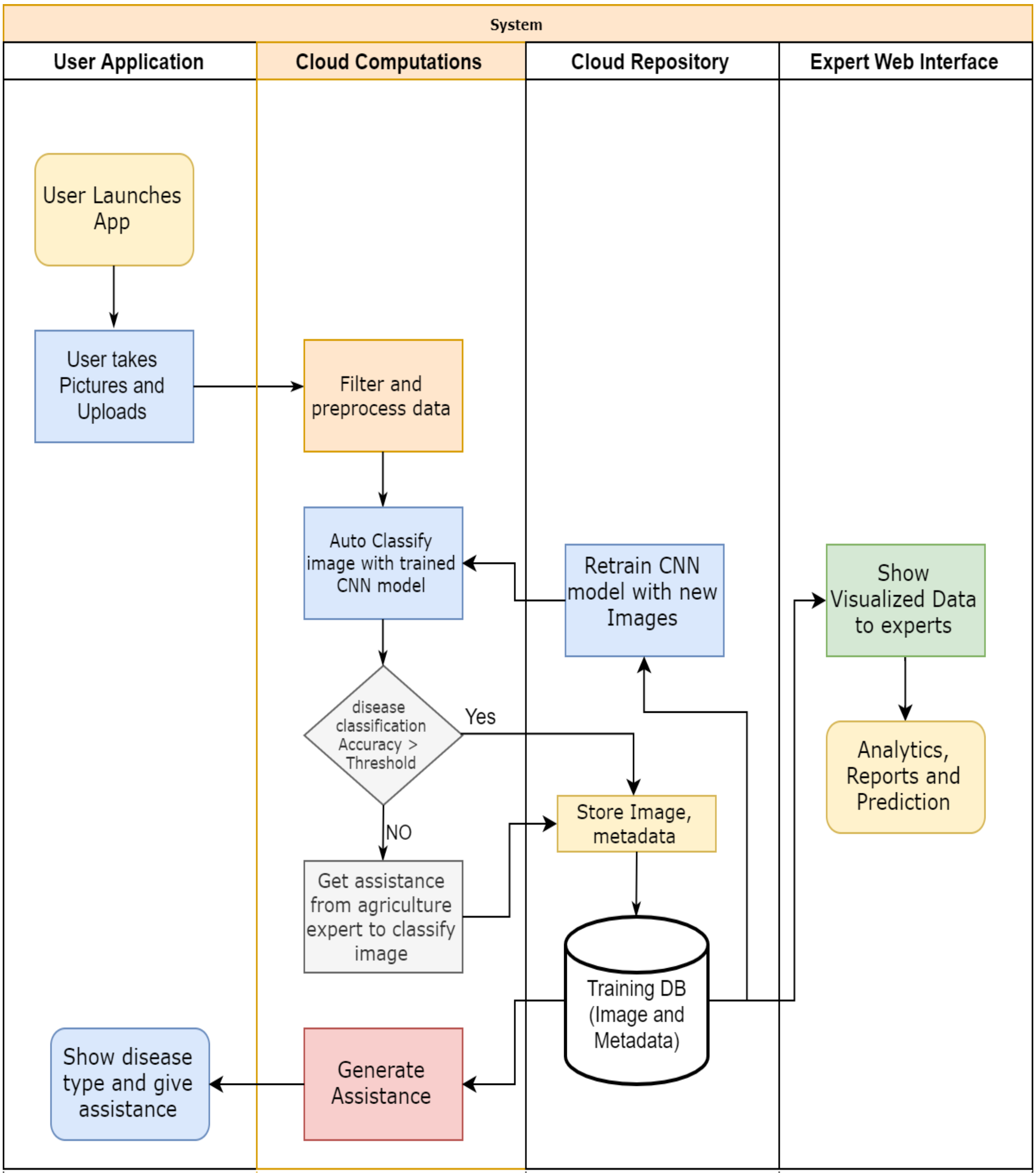

Figure 4.4: Activity Diagram

4.4.3 Sequence Diagram

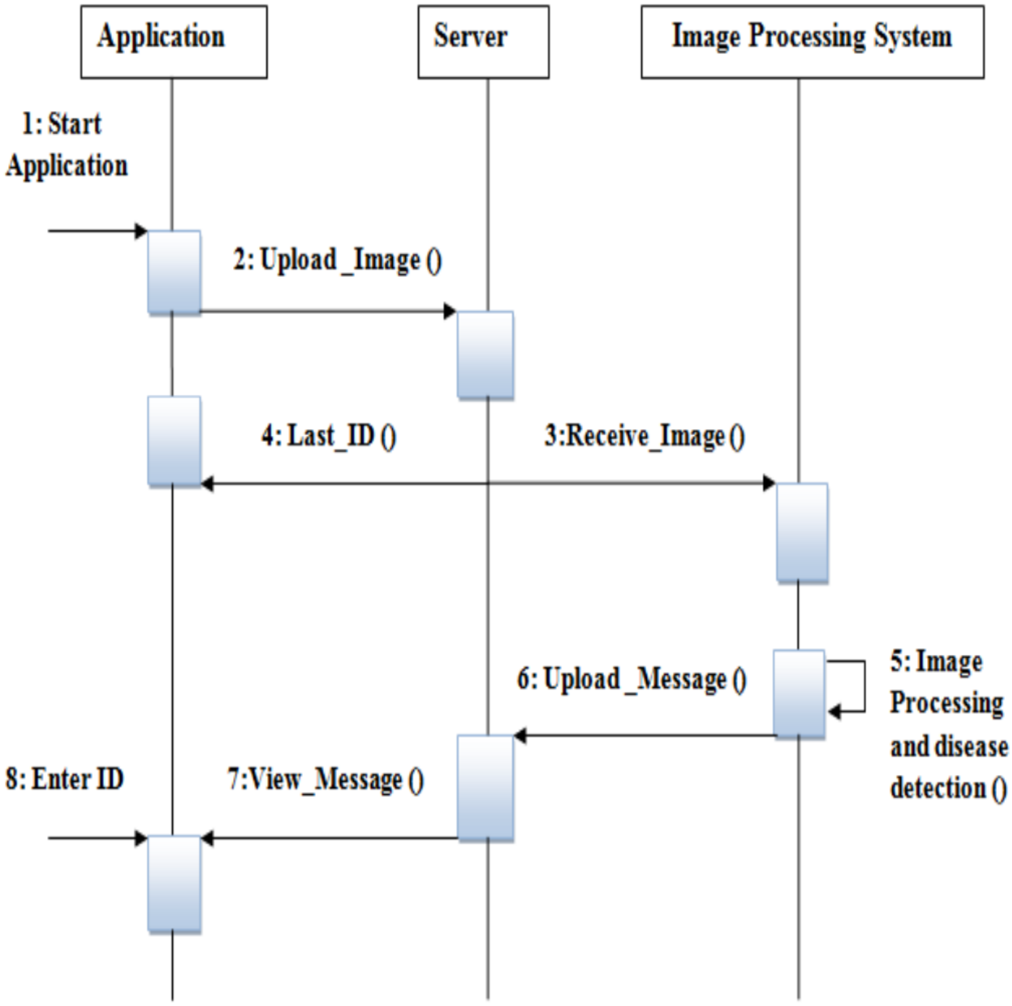

Figure 4.5: Sequence Diagram

4.4.4 User Interaction Diagram

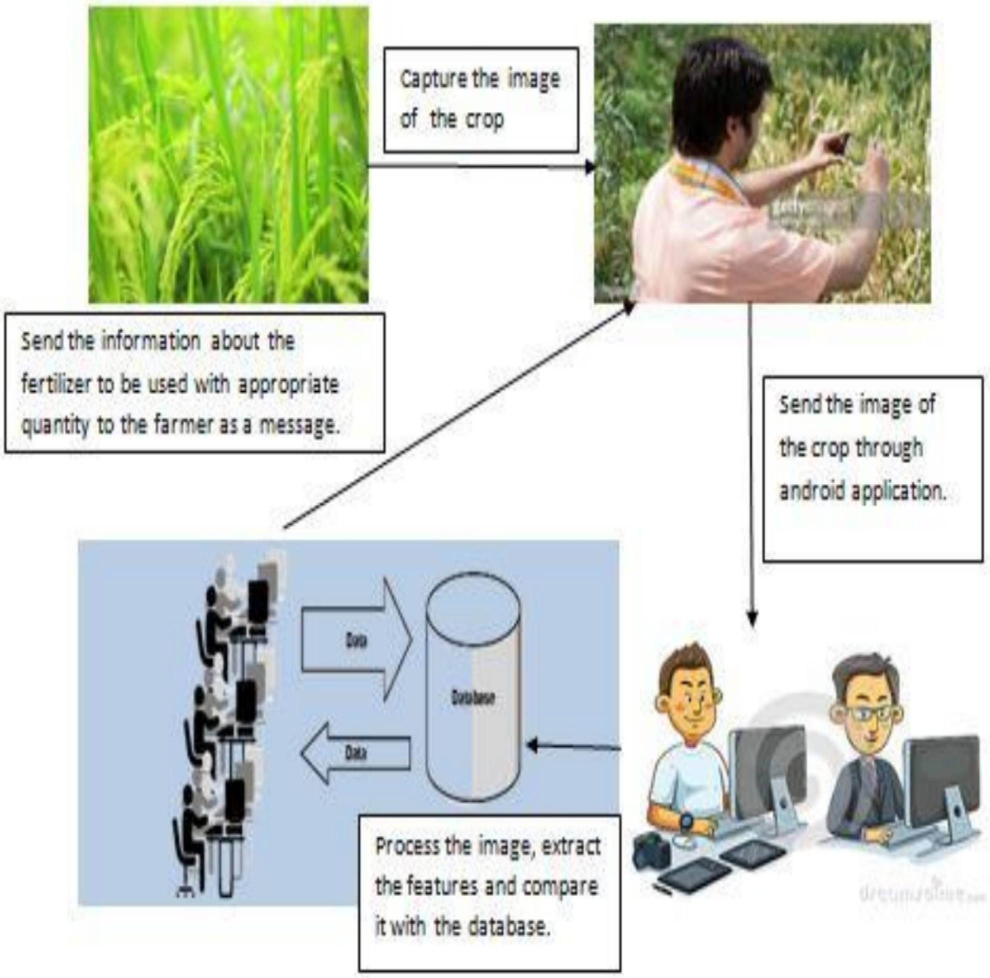

Figure 4.6: User Interaction Diagram

4.4.5 Process Diagram

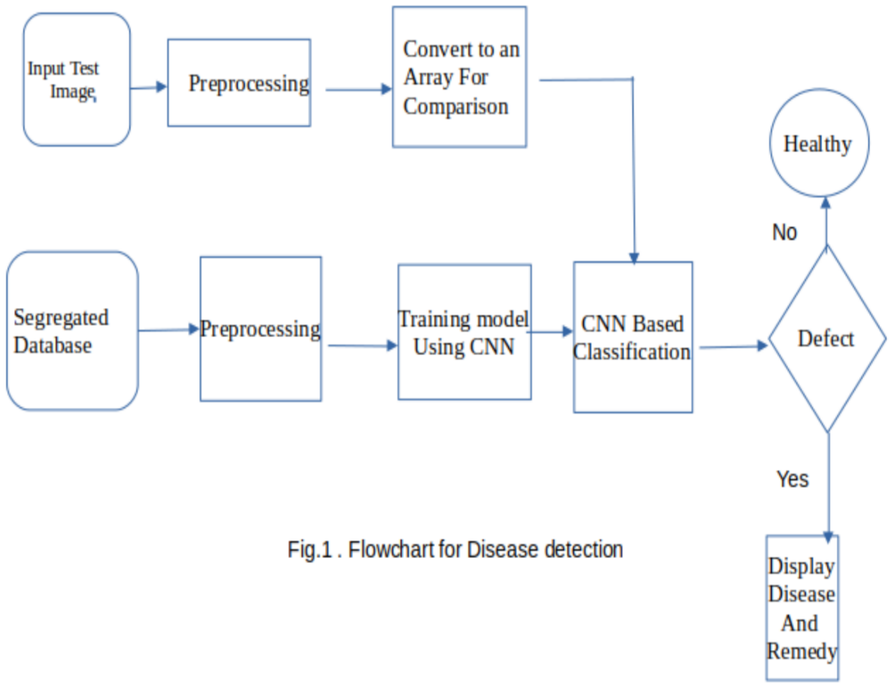

Figure 4.7: Process Diagram

CHAPTER 5

DETAILED DESIGN DOCUMENT

5.1 ARCHITECTURAL DESIGN

The proposed system is implemented using CNN deployed on serverless platform.

5.1.1 System Architecture

Input set includes 3 types of inputs i.e. image input from end user of infected crop, expert suggestion from data analyst and augmented dataset.
Processing is aimed to process inputs for required output as stated. The output acquired from system are disease detection and recommendation about medication for end user and efficiency evolution for developer.

architecture.png

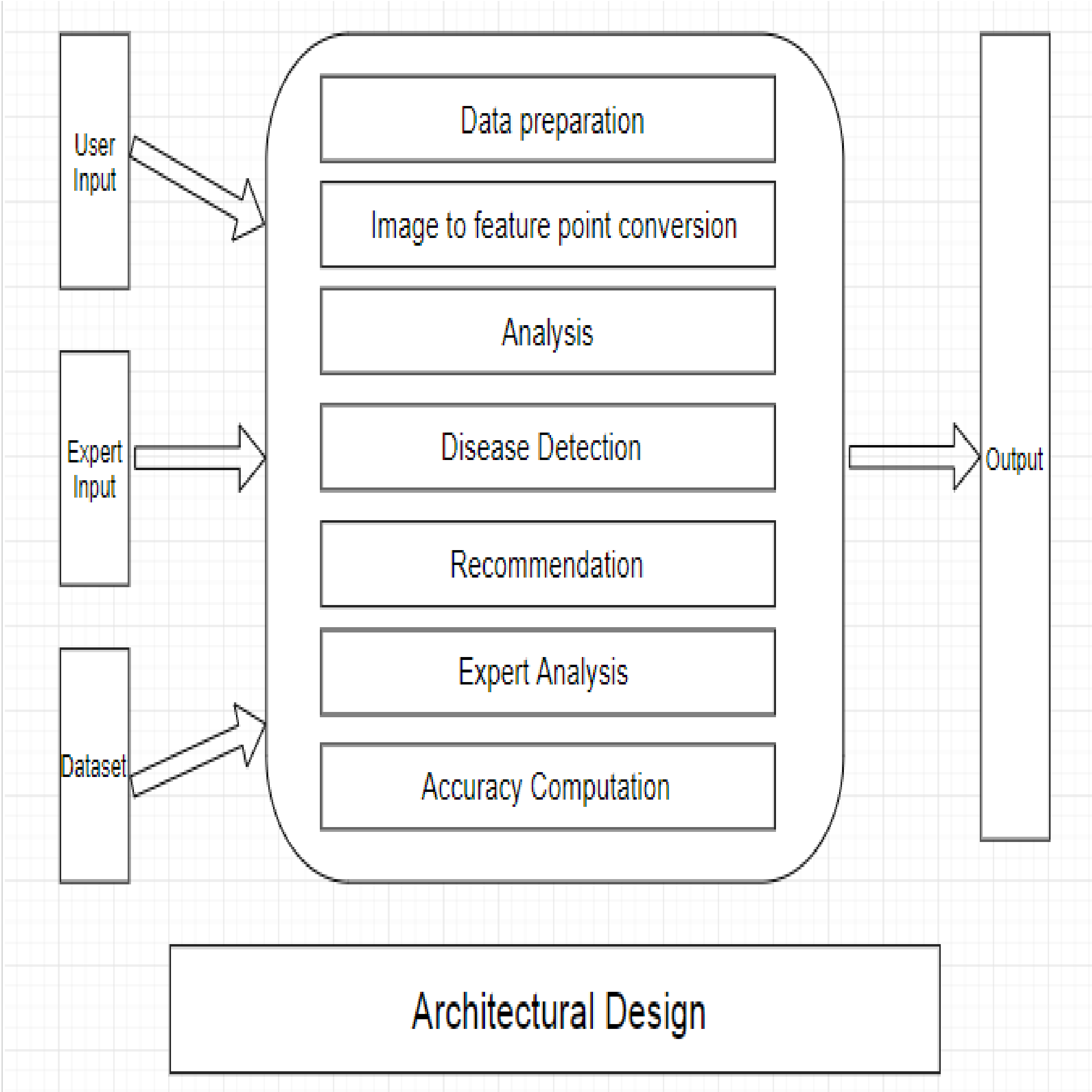

Figure 5.1: Block Diagram

5.1.2 Deployment Architecture

The proposed system is deployed on serverless platform.

A serverless architecture is a way to build and run applications and services without having to manage infrastructure. By using a serverless architecture, the core system is focused instead of worrying about managing and operating servers or runtimes, either in the cloud or on-premises. This reduced overhead hence developers reclaim time and energy that can be spent on developing great applications which scale and that are reliable. Serverless is a cloud computing execution model where the cloud provider dynamically manages the allocation and provisioning of servers. A serverless application runs in stateless compute containers that are event-triggered, ephemeral (may last for one invocation), and fully managed by the cloud provider.

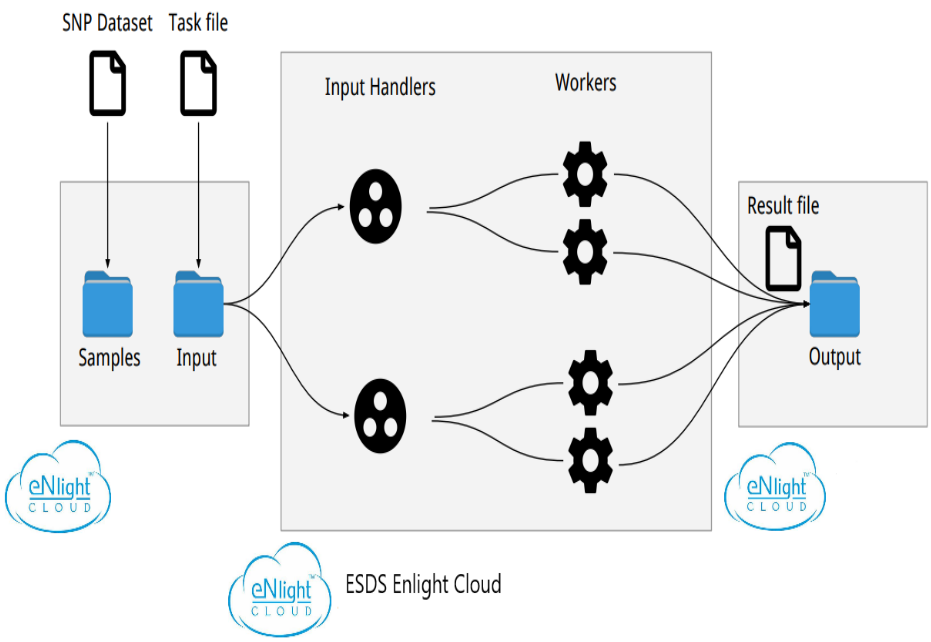

Figure 5.2: Serverless Architecture

5.1.3 CNN Architecture

The proposed system is implemented using CNN.
CNN architectures vary with the type of the problem at hand. The proposed model consists of five convolutional layers. There are 3 Convolutional layers are followed by a Max Pooling layer. Again a max pooling layer is present after next 2 Convolutional layers. The final layer is fully connected GAP(Global Average Pooling) . ReLu activation function is applied to the output of every convolutional layer and fully concted layer. The first convolution layer filters the input image with 32 kernels of size 3x3. The output of preceding layer is given as an input for the second convolutional layer with 64 kernels of size 4x4. The last convolutional layer has 128 kernels of which the output is given as an input for the second convolutional layer with 64 kernels of size 4x4. The last convolutional layer has 128 kernels of size 1x1 followed by a fully connected layer of 512 neurons. The output of this layer is given to softmax function which produces a probability distribution of the four output classes.

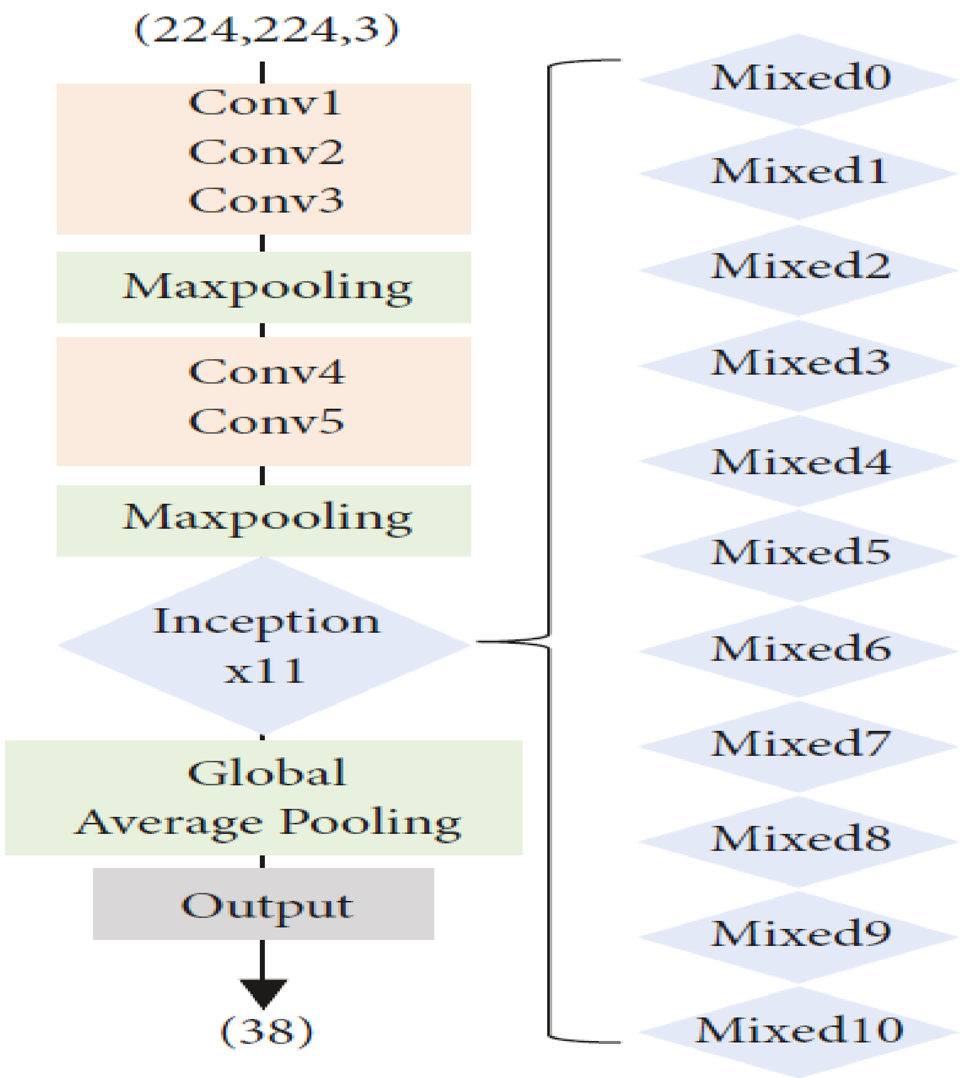

Figure 5.3: CNN Architecture

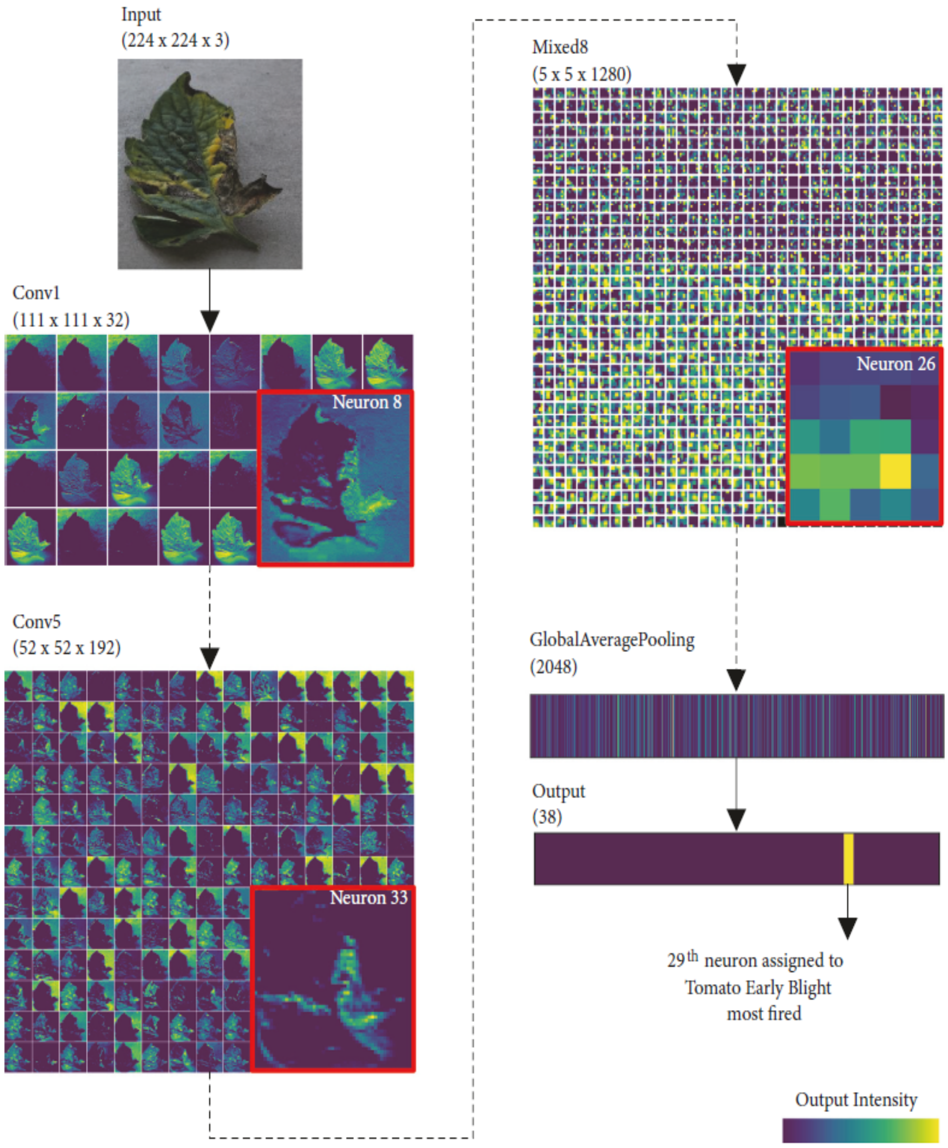

Figure 5.4: CNN block diagram

CHAPTER 6

SUMMARY AND CONCLUSION

6.1 SUMMARY

System is an Integrated and collaborative serverless platform with an automated image-based classification system for the identification of plant diseases and advising the preventive measures. End users clicks images in desired resolution through their smart phones and upload it through android application platform. System suggests the solution for the images received through the model trained by the dataset of images of diseased plants. In worst case if the model does not come to a conclusion, web interface allows experts to perform disease analytics with geographical visualizations with the help of disease density maps with spread forecasting which is to be rendered from that Cloud based repository. The support for the identified disease is provided to the end user through the android application. It makes easy for the user to identify the disease and take preventive measures over it.

6.2 CONCLUSION

Proposed system is structured and designed as per sponsor requirement. Thus, the proposed system is estimated to be completed within 180 days. As part of the contract, the setup acceptance procedures for preliminary and final acceptance of the project results. In order to create enough flexibility supplement this system by a risk management plan, and by integrating change and claim processes into the contract.Thus the system is planned to deploy on severless platform. Hence, concluded project planning phase report.

REFERENCES

[1] CNCF Serverless White Paper
https://github.com/cncf/wgserverless/blob/master/whitepapers/serverless-overview/cncf_serverless_whitepaper_v1.0.pdf

[2] Comparison of CNN Models for Application in Crop Health Assessment with Participatory Sensing.(Prakruti Bhatt, Sanat Sarangi, and Srinivasu Pappula)

[3] Omkar Kulkarni;Crop Disease Detection Using Deep Learning.

[4] Crops Disease Diagnosing using Image-based Deep Learning Mechanism :Hyeon Park, Eun JeeSook and Se-Han Kim

[5] Proceedings of the Third International Conference on Electronics Communication and Aerospace Technology [ICECA 2019] IEEE Conference Record 45616; IEEE Xplore ISBN: 978-1-7281-0167-5 ”CNN based Leaf Disease Identification and Remedy Recommendation System” : Suma V R Amog Shetty, Rishab F Tated, Sunku Rohan

[6] ”Leaf Disease Detection and Recommendation of Pesticides using Convolution Neural Network”:Pranali,K.Kosamkar,Shubham Rudrawar,Dr.V.Y.Kulkarni,Shubhan Salmpuria,Krushna Mantri,Nishant Gadekar

[7] ”Plant Disease Detection Using CNNs and GANs as an Augmentative Approach”:Rutu Gandhi,Shubham Nimbalkar, Nandita Yelamanchili,Surabhi Ponkshe.

[8] 2018 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM) ”An Artificial Intelligence and Cloud Based Collaborative Platform for Plant Disease Identification, Tracking and Forecasting for Farmers”: Kaushik Kunal Singh

[9] ”Detection and Classification of Plant Leaf Diseases by using Deep Learning Algorithm” : M. Akila,P. Deepan

[10] FourthInternational Conferenceon Industrialand InformationSystems, ICIIS 2009, 28 - 3I December2009, Sri Lanka ”An Image Recognition System for

Crop Disease Identification of Paddy fields in Sri Lanka" :G. Anthonys', N. Wickramarachchf

[11] 2019,5th International Conference on Advanced Computing Communication Systems (ICACCS)"Performance Analysis of Different CNN Architecture with Different Optimisers for Plant Disease Classification"

CHAPTER 7

MATHEMATICAL MODEL

System: S = {I, O, F}

where,

S : System model

I : Set of Inputs

$I = \{I_1, I_2, I_3\}$

where,

$I_1 = \{I_{11}, I_{12}, I_{13}, \ldots\ldots\ldots I_{1n}\}$

I_1 : Set of images taken input from user

$I_2 = \{I_{21}, I_{22}, I_{23}, \ldots\ldots\ldots I_{2n}\}$

I_2 : Set of textual input about pesticide details

$I_3 = \{I_{31}, I_{32}, I_{33}, \ldots\ldots\ldots I_{3n}\}$

I_3 : Set of textual input from experts

O : Set of Outputs

$O = \{D, R, V, A\}$

where,

$D = \{D_1, D_2, D_3, \ldots\ldots\ldots\ldots D_n\}$

D : Set of dieses predicted

$R = \{R_1, R_2, R_3, \ldots\ldots\ldots\ldots R_n\}$

R : Set of recommendation about pesticides and fertilizers

$V = \{V_1, V_2, V_3, \ldots\ldots\ldots\ldots V_n\}$

V : Virtualization report for data analytic experts

$A = \{A_1, A_2, A_3, \ldots\ldots\ldots\ldots A_n\}$

A : Accuracy report

F : Set of functions applied

$F = \{ F_c \}$

The proposed CNN is implemented as shown below:

$$F_c : I \rightarrow x\,1 \rightarrow w1 \rightarrow x\,2 \rightarrow \ldots \rightarrow x\,L{-}1 \rightarrow wL{-}1 \rightarrow x\,L \rightarrow wL \rightarrow O$$

Above equation shows how the proposed model of the CNN runs layer by layer. The input x 1 is an image of a leaf disease or a sound one with order 3 tensors. The x 1 was the input to the first layer's input collectively known as tensor W1. The output of the first layer was x 2 which also acted as the input to the second layer processing. The processing proceeded until all layers in the CNN had been finished, which gave an output of x L. One additional layer, however, is added for backward error propagation for learning parameter values in the CNN. The last layer is a loss layer. A simple loss function used is given by below equation:

$$z = \frac{1}{2} \parallel t - x^L \parallel^2$$

In above equation, t denotes the corresponding target value for the input x L . The loss function formula in above is used to measure the discrepancy between CNN prediction x L and t.

Alternatively, the output prediction of the CNN is given as shown in below equation.

$$arg_i max\ x_i^L$$

The loss layer is not needed in the prediction, but is useful in learning of CNN parameters using a set of diseased and healthy maize leaf data sets as training sets. Our CNN model used the stochastic gradient descent (SGD) in order to learn the model parameters. Instead of outputting a prediction, we compared the prediction with target t corresponding to input. The loss Z is then a supervision signal, guiding how the parameters of the model is to be updated. The SGD way of modifying the parameters is as shown in below.

$$w^i \leftarrow w^i - \eta \frac{\partial z}{dw^i}$$

where η represents the learning rate. The learning rate is aim to be chosen as 0.01. Equation given below has a superscript "time" index (e.g., training epochs/iterations).

$$\left(w^i\right)^{t+1} = \left(w^i\right)^t - \eta \frac{\partial z}{\partial \left(w^i\right)^t}$$

The activation function used in the convolution layer was the rectified linear unit (ReLU) function. The ReLU can be regarded as a truncation performed individually for every element in the input. The ReLU is presented by Equation below

$$j_{i,j,d} = \max\{0, x^{l}_{i,j,d}\}$$

The l-th layer, had inputs that formed an order 3 tensor x1 with

$$x^{l} \in R^{H^{l} \times W^{l} \times D^{l}}$$

Thus, for this reason it needs a triplet index set { i^{l}, j^{l}, d^{l}} to locate any specific element in x l . Based on Equation above, it is obvious that in below,

$$\frac{\partial y_{i,j,d}}{\partial y^{l}_{i,j,d}} = [\![x^{i}_{i,j,d} > 0]\!]$$

where [[.]] was the indicator function, being 1 if its argument is true, and 0 otherwise. Hence, it has

$$\left[\frac{\partial z}{\partial x^{l}}\right]_{i,j,d} = \begin{cases} \left[\frac{dz}{dy}\right]_{i,j,d} & if\ x^{l}_{i,j,d} > 0 \\ 0 & otherwise \end{cases}$$

y is an alias of x^{l+1} . An alias means that a variable can be reshaped into another form. To be specific, the function max (0, x) is not differentiable at x = 0. Focusing on the convolution layer, the normalized Kernel is used to convolve the input images of both diseased and healthy leaves. The convolution operation is explained in below

$$(x,y) = \sum_{m=-\frac{M}{2}}^{\frac{M}{2}} \sum_{n=-\frac{N}{2}}^{\frac{N}{2}} h(m,n) f(x-m, y-m)$$

h(m, n) is a filtering mask of size M × N. Each element in this filter mask represents the weights used in the linear combination. It is at this stage where the ReLU activation function was used with the convoluted input images of both diseased and healthy leaves in order to bring non-linearity.

CHAPTER 8

PLAGIARISM REPORT

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	253	Date	October 17,2019
Characters	1679	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	16 Unique Sentences

Content Checked For Plagiarism

Plant diseases are a threat to farmers, consumers, the environment and the global economy. Almost 40% of the world's crop yield is damaged due to diseases and pest infestation. According to the survey in 2012, Maharashtra has the highest rate of farmer suicides and one of the major reasons for this is the failure of crops. The use of random pesticides and insecticides also is harmful to crops. These adverse effects on the yield can be avoided through early detection and proper expert's guidance. We propose an integrated and collaborative platform with an automated image-based classification system for the identification of plant diseases and advising the preventive measures for the same. Since existing datasets focus across several countries and none focuses on Indian crops specifically, there is a need for establishing a local dataset to be of use to Indian farmers. It uses Generative Adversarial Networks (GANs) to augment the limited number of local images available. The classification is done by a Convolutional Neural Network (CNN). The AI model continuously learns from user-uploaded images suggestions given by experts to enhance its accuracy. For preventive measures, disease density maps along with spread forecasting are rendered from a Cloud-based repository of geo-tagged images and micro-climatic factors. A web interface will allow experts to perform disease analytic with geographical visualizations. Some pesticides reduce soil fertility if applied frequently. So, with the help of input from the user about previous medication, appropriate recommendations will be provided to the user for proper medication with retaining maximum soil fertility

Sources	Similarity

Figure 8.1: Plagiarism Report for Abstract

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	231	Date	October 17,2019
Characters	1552	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	17 Unique Sentences

Content Checked For Plagiarism

While India has progressed in a number of different sectors, the development of agriculture in India has not grown in parallel with the new available technologies. • Crop diseases adversely affect the agricultural yield, they even causing famine and socio-economic distress. India has the highest number of Farmer's suicidal cases. • In developing countries, the majority of agricultural produce comes from small scale farmers and major loss is reported due to pests and diseases. • Timely and accurate identification of crop condition is vital for implementing appropriate measures immediately, specifically where certain diseases have no treatment or spread rapidly • Diagnosis through visual examination could be time consuming and error prone while existing lab based diagnosis methods could turn out too expensive. • There is a need for a system where farmers can identify crop diseases at an early stage and perform proper medications with the help of computer algorithms. There should be a collaborative platform that connects experts and farmers for better crop yield. An Integrated and collaborative platform with an automated image-based classification system for the identification of plant diseases and advising the preventive measures. Farmers can use a mobile app to capture the image of an infected plant or leaf and will get user-friendly recommendations. • For Preventive measures, disease density maps with spreading forecasting are rendered from a Cloud-based repository of geo-tagged images and microclimatic factors.

Sources	Similarity

Figure 8.2: Plagiarism Report for Introduction

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	752	Date	October 17,2019
Characters	4653	Exclude Url	

4% Plagiarism	96% Unique	2 Plagiarized Sentences	46 Unique Sentences

Content Checked For Plagiarism

To improvise a system which will classify and detect disease of a plant from an image given by an mobile app or web portal and give proper user friendly assistance to the user. • To create cloud based repository of geo-location tagged images uploaded by farmers with micro-climatic factors and update it with new entries which will help for analytics. • To use a web scrapper to find plant images and create local database of plant images using Generative Adversarial Networks (GANs) to augment the limited number of local images available. • To create a web interface allows experts to perform disease analytics with geographical visualizations with the help of disease density maps with spread forecasting which to be rendered from that Cloud based repository • To use a web scrapper to find plant and crop images and similar datasets. • To create a database of augmented plant images using local images available which focuses on Indian crops. • To standardize all the datasets retrieved in one form and preprocess it. (Same extension, same resolution, remove inappropriate data) • To deploy the CNN model efficiently on Serverless platform. Build APIs and Actions for the same. • To reduce the cost of computations with the help of Serverless features. • To securely retrieve images of infected plant or leaf to the cloud and call appropriate APIs for triggering the CNN. • To accurately detect the disease of the plant from the input image and give user-friendly assistance regarding the disease. To achieve high accuracy and increase it with the help of experts recommendations. • To provide disease density maps with spreading forecasting from cloud-based repository of geo-tagged images and climatic factors for analysis and prediction. The system takes image as a input, and provides assistance message as a output. • For the expert interface, it will show images and will take its names (names of disease) and assistance message increase the accuracy. • The system make use of Experts to increase the scope of prediction, currently it does not connect users with experts The dataset available which focuses on Indian Crops and plants are very Less. Thus, We will need to generate local dataset of plant images using Image Augmentation. We will be using Agile Methodology to develop the entire application as discussed with ESDS. At first small model will be trained with limited dataset and will be deployed on ESDS eNlight Serverless platform for testing. After Successful Implementation, further actions and more bigger model will get deployed and increments will be done periodically. Basic steps of the system are described below: 1. Step 1: Data preprocessing: all the images in dataset are resized to 100x100 pixel format. 2. Data is divided into two parts 80% training set, 20% test set. 3. Data augmentation: augmentation process is applied of Training set to rotate, resize and adding some random noise to images in order to avoid over fitting. 4. Feature extraction: Features would be extracted in starting layers of CNN architecture using convolutional operation. 5. Training the model: In our case we will use LeNet based architecture [9]. Once architecture is developed we will train the model with Training set features. 6. Evaluation: Accuracy of model would be evaluated with the help of Test set. 7. Tuning: If results are not satisfactory tune the model by changing the parameters of architecture such as kernel size, Nodes in last fully connected layer. 8. Store the weights: final model which has trained save it in model name.h5 configuration file so that it can be used for new data. 9. Mobile Application: application would be developed using java for android to upload images on server, call the APIs and display the results. 10. Server Side application: this application responsible for preprocessing the image uploaded by user and classify it based on its features and give the results in the form of JSON objects. 11. Update the Cloud database and other information for further analytics. 12. extract the features and evaluate with trained model. 13. Sending back the results to application. 14. Display the results on smartphone. The entire Cloud-based application is based on Serverless Computing which is a cloud Computing Execution model. Whole Application is segregated into many actions interfaced with their APIs which gets called whenever needed. Hardware Resources Required are needed before deploying to the server i.e. if model training is done on local computer at initial stage. Since the application will be hosted on ESDS eNlight Serverless platform, it is provided by ESDS.

Figure 8.3: Plagiarism Report for Problem Definition and scope : Part 1

Sources	Similarity
Crop Disease Detection using Deep Convolutional Neural Networks...Compare text step 6: evaluation: accuracy of model would be evaluated with the help of test set. step 7: tuning: if results are not satisfactory tune the model bystep 9: application android: application would be developed using java for android to upload images on server and display the results. https://www.ijert.org/crop-disease-detection-using-deep-convolutional-neural-networks	5%

Figure 8.4: Plagiarism Report for Problem Definition and scope : Part 2

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	266	Date	October 17,2019
Characters	1651	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	13 Unique Sentences

Content Checked For Plagiarism

Hardware Resources Required are needed before deploying to the server i.e. if model training is done on local computer at initial stage. Since the application will be hosted on ESDS eNlight Serverless platform, it is provided by ESDS. Risk Identification 1. Datasets that primarily focuses on Indian Plants are so less. 2. Overfitting of CNN algorithm 3. In case the Servers went down 4. Poor Implementation of Serverless functions 3.2.2 Risk Analysis 1. Datasets that primarily focuses on Indian plants are less. Those who are available on internet does not contain enough images or have inappropriate data. This can result in model not getting trained well enough to classify other images. 2. Overfitting is the big risk while using CNN with less data. Overfitting happens when the model learns the detail and noise in the training data to the extent that it negatively impacts the performance of model on new data. 3. If servers went down, whole application will get unusable. This is the risk where the entire application becomes inaccessible to users. 4. This is the risk associated with the Cloud Provider. If they have implemented the serverless functionality poorly, this can negatively impact the performance of the application. Team of 4 members KKWIEER, Department of Computer Engineering 2019 18 • 1 Project Guide • 1 Project Co-ordinator 3.4.1 Team structure A team of 4 members 3.4.2 Management reporting and communication • Weekly Reporting to guide about the work. • Weekly Reporting to ESDS Mentor (Sponsors) • Project Idea Presentation (1st Sem) • Progress Presentation (1st Sem) • Project Execution and Demo (2nd Sem)

Sources	Similarity

Figure 8.5: Plagiarism Report for Project Plan

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	457	Date	October 17,2019
Characters	2921	Exclude Url	

5% Plagiarism	95% Unique	1 Plagiarized Sentences	20 Unique Sentences

Content Checked For Plagiarism

The purpose of the project to ease the assistance process of Plants diseases to the users with the help of CNN based classification. The system gives user friendly assistance about medication and care and it also gives visualized data about the crop diseases over large area for analytics. In order to address the present limitations and overcome the issues in current plant disease detection approaches, the entire process of the proposed approach is described in this section along with series of experiments addressing current issues in plant disease detection tasks. The proposed method focuses on several important stages in the development of a plant disease detection model including the introduction of a new dataset and augmentation methods, analyzing different classification and the object detection algorithms while proposing a novel approach for plant disease detection. One of the crucial drawbacks of the present study in the area of plant disease detection is a major drop in the classification performance of the models on real images taken in fields compared to the images from a controlled environment. The reason for it is the lack of large public datasets of plant diseases and most of the present achievements are based on the PlantVillage dataset. PlantVillage consists of 54,323 images divided in 38 classes of diseased and healthy plants based on 14 different crop species; it is available online from the crowd-sourced platform Kaggle(Supplementary Materials). All images are taken as a single leaf on a solid background which is labeled only by a class name. Due to this, PlantVillage is used as a performance metric in many papers, which can be sometimes misleading when used in real on-field data, taking into account the purity of the dataset. The Object detection algorithms were explored in the plant disease detection task. Due to this, the PlantVillage dataset was additionally labeled generating metadata holding two types of bounding boxes, one marking the entire leaf and the second one marking only the infected areas. All additional labeling was performed and verified by agricultural experts. In order to adapt the plant disease detection model for more practical usage, a new dataset was introduced, the Plant Disease dataset. This dataset is a continuation of the author's previous work. New images of healthy and diseased leaves have been taken in fields under various weather conditions and at numerous angles during different daylight intensity. Alongside this, all images have an inconsistent background, meaning that sometimes a single leaf is in focus, and sometimes images hold many leaves of different crops whether they are healthy or infected with different diseases. This way, dataset consists of images that mimic practical situations where the model could be potentially used. All images from the dataset were manually labeled and verified by agricultural experts.

Sources	Similarity
Symmetry | Free Full-Text | Solving Current Limitations of Deep...Compare text PlantVillage consists of 54,323 images divided into 38 classes of diseased and healthy plants based on 14 different crop species; it is available online from the crowd-sourced platform Kaggle [48] (Supplementary Materials). All images are taken as a single leaf on a solid background labeled... https://www.mdpi.com/2073-8994/11/7/939/htm	10%

Figure 8.6: Plagiarism Report for Software Requirements and Specification

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	749	Date	October 17,2019
Characters	4692	Exclude Url	

6% Plagiarism	94% Unique	2 Plagiarized Sentences	31 Unique Sentences

Content Checked For Plagiarism

In order to overcome issues like overfitting, DL algorithms often rely on big datasets. This presents a general obstacle when training algorithms for wider practical usage. Gathering data can be a time-consuming process, which could also require having domain experts for labeling tasks. In order to enlarge the existing datasets, augmentation techniques are a common approach. Two augmentation strategies were used. The first included traditional augmentation methods widely applied in many plant disease detection studies. Most of these methods include simple transformations such as rotations or pixel-wise changes like blurring or noising in order to introduce distortion to images. In this research, rotations by various angles, and perspective transformations along with size preserving shearing, shifting, and mirroring was applied to the training dataset. The second approach included training GANs responsible for generating syntactic data based on the existing dataset. A plain GAN architecture consisting of two neural networks was used as a participant of the competing game, where the discriminator tries to determine whether the data is real or fake, while the generator tries to create the data with all of the required features to trick the discriminator into thinking that the data are real. By applying this approach, a dedicated GAN could generate novel images that could be used in the training phase of a plant disease classier. This type of network has proven more successful in different benchmarks than the Variational Auto Encoders (VAEs), Restricted Boltzmann Machines, etc. Over the years, many of the new GAN architectures have been introduced. In this paper, several architectures were applied to explore the possibility of using syntactic data to train the plant disease classifier. Due to its simplicity in design (5-layer discriminator/generator), DCGAN was used as a proof of concept whether the syntactic data could have features like color, shape, and texture like the real plant leaves. Based on Figure 1, it can be observed that the syntactic data are displaying the desired features. DCGAN architecture is designed for generating 64 x 64 images, while training for higher resolutions is highly unstable as one network becomes stronger than the other, which prevents the learning process. In order to use this syntactic data in the training phase, higher resolution images must be generated as most of the state-of-the-art convolutional neural networks are designed for input image sizes around 256 x 256. For this reason, Progressively Growing GAN (ProGAN) was used to generate plant leaves at a 256 x 256 size. ProGAN starts by creating a tiny image of 4 x 4 or 8 x 8 pixels until the image is considered realistic by the discriminator. When the initial learning process is complete, ProGAN adds higher-resolution layers that are the next to be trained. This process continues until 1024 1024 pixels (or lower dimensions based on the number of layers) is trained. Although the network was stable during the training phase for the plant leaf images, it did not manage to generate images with a rich set of features that could represent the plant leaves well enough. The architecture that most successfully generated plant leaf images in higher dimensions (256 256 used in the experiment) was Style GAN, a new design that combines ProGAN and neural style transfer. The original architecture was adopted to generate the desired input dimension of 256 x 256. This was achieved by adding a styled convolutional block of 128 64 3, following a 64 3 convolutional block as the last layer to the generator network, and a 64 128 3 convolutional block following a 3 64 convolution as the first layer of the discriminator network. To train Style GAN on the entire PlantVillage dataset for generating novel plant leaves, the following parameters were used: a learning rate of 3 103, a minibatch size of 5, and an AMSGrad optimizer. In order to explore the learned features of the trained Style GAN, the guided backpropagation technique was used. This method uses a simple backward pass of the activation of a single neuron after a forward pass through the network to visualize the part of an image that mostly activates a certain neuron. Note that the same GAN architectures were applied to the PlantDisease dataset, but the trained models were highly affected via the background features and the generated images could not distinguish the leaves and their structure from the background. Developing such GAN-based models for generating plant leaves with complex backgrounds was out of the scope of this paper, but could be of great significance for further research.

Figure 8.7: Plagiarism Report for Software Requirements and Specification

Sources	Similarity
Symmetry | Free Full-Text | Solving Current Limitations of Deep...Compare text the current limitations and shortcomings of existing plant disease detection models are presented and discussed in this paper.two approaches were used to augment the number of images in the dataset: traditional augmentation methods and state-of-the-art style generative adversarial networks. https://www.mdpi.com/2073-8994/11/7/939/htm	7%

Figure 8.8: Plagiarism Report for Software Requirements and Specification

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	949	Date	October 17,2019
Characters	5953	Exclude Url	

4% Plagiarism	96% Unique	2 Plagiarized Sentences	49 Unique Sentences

Content Checked For Plagiarism

Proper and large dataset is required for all classification research during the training and the testing phase. The dataset for the experiment is downloaded from the PlantVillage database which contains different plant leaf images and their labels. It contains a collection of images taken at different environment. A dataset containing 12,673 leaf images of four classes including healthy leaves is downloaded. The quantity of information and the diversity within the images varies among the studies. Three types of datasets can be defined, depending on their level of complexity. The first type consists of images captured under controlled conditions. In this case, images show one leaf picked up in the field and placed on a uniform background, in an environment with controlled illumination. This simplifies the image analysis process by removing any kind of variability related to external conditions or plant morphology in order to focus on symptom expression. Classes are defined by diseases and the species (in the case of multi-species models). Classes reflect disease severity levels. Both the intensity and the stages of infection can lead to a high degree of variability in symptoms. This variability can be expressed through separate classes or integrated in global classes. However, since it is difficult to obtain enough images for all expressions, especially over a single growing season, its recommended to continuously add new images to the training dataset. Different types of strategies can be chosen either at a data level or at an algorithm level to minimize the detrimental effect of imbalance. System divided their healthy leaves' class into 12 clusters of 110 images for training and 27 for testing for instance, thereby providing classes of the same size (between 102 and 144 for training and between 23 and 36 for testing). Having a taxonomy that only includes individual diseases is a simplification of reality. Very often, diseases, nutritional disorders, and/or pests can be present at the same time, combining their symptoms. Creating a class for each phytosanitary problem combination does not seem to be a suitable solution since the number of possible classes would increase considerably. The association of a label to all or part of an image—is a laborious but unavoidable step in supervised learning. Repetitive and time-consuming, it must be carried out by an expert in identifying crop diseases, which makes this task difficult to delegate. The annotation method depends on the general approach chosen for image analysis. For classification, it consists of associating a label to each image, either by integrating it in the metadata or by organizing the images, e.g., into folders corresponding to the different classes. For object detection, the coordinates of the target within the image must be entered. This is done by delineating regions of interest that are often rectangular but may also correspond more precisely to the object in question. For the identification of diseases, the annotation step raises the question of the analysis scale and of the importance given to the context. Each of these levels are valid and provide complementary features. With a close symptom view, textural elements stand out . A complete view of the leaf reveals symptom patterns. Finally, a view of the whole plant provides a spatial perspective of the symptoms. When using deep neural networks, three separate datasets are required to develop a model. The first set, the training set, is the collection of images to be used by the network to automatically learn its hidden parameters, such as weights and biases. The second set, the validation set, is used to manually adjust hyperparameters, which are essentially the settings that cannot be automatically learned during training. These include among others the learning rate, the batch size and the network architecture. For more information about hyperparameters. The values of these hyperparameters are often set empirically, as they are linked to the problem, the dataset, and the model architecture. Therefore, there are no good predefined values, as they must be tuned based on the performance (in terms of accuracy) obtained on the validation set.

This means that information about the validation data indirectly leaks into the model, resulting in an artificial ability to perform well on these images. For that reason, the validation images should only be used to tune the hyperparameters; the final evaluation of the model's performance is done using the test set, discussed in the next paragraph. The model being trained can be evaluated on the validation set at the end of each epoch, allowing the training process to be monitored and to detect overfitting. The training and validation sets come from the same data source that is subdivided. Most of the images go

Figure 8.9: Plagiarism Report for Detailed Design Document

detect overfitting. The training and validation sets come from the same data source that is subdivided. Most of the images go for training (between 70 and 85\% depending of the size of the dataset). Tried five different separation ratios and both concluded that using 80\% for training and 20\% for validation was ideal for their data. Another way to divide the images into training and validation sets is cross validation. The third dataset that is needed is the test set. It is used when the training phase has been completed, with the objective of evaluating the model's final generalization ability. The accuracy on the test set is thus the most important metric to compute, as it provides an overview of the model's performance beyond the hyperparameter exploration process. The test set must be independent from the training and validation sets, so it cannot be obtained from a simple subdivision. However we analyzed (31.5\%) formed their test set this way. Worse yet, 11 of the studies (58\%) did not even have a test set. (10.5\%) performed evaluation on an explicitly different test set. Having those three datasets is essential, since the observed data variability in the agricultural setting is quite important.

Sources	Similarity
Frontiers | Convolutional Neural Networks for the Automatic...Compare text the annotation method depends on the general approach chosen for image analysis. for classification, it consists of associating a label tofigure 3. selecting an analysis scale: from a scale close up on the main symptoms (a) to a scale providing more contextual features (d). example of a... https://www.frontiersin.org/articles/10.3389/fpls.2019.00941/full	4%
Wenzhun Huang's research works | Xijing University, Xijing and other...Compare text however, most of the methods of existing kernel gc (kgc) image segmentation not only suffers from different types of noise, but also suffers fromto enhance the robustness and efficiency of the current iot systems, we adopt the sparse coded dictionary learning theory to detect the size of the data and... https://www.researchgate.net/scientific-contributions/2055634333_Wenzhun_Huang	2%

Figure 8.10: Plagiarism Report for Detailed Design Document

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	541	Date	October 17,2019
Characters	3539	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	10 Unique Sentences

Content Checked For Plagiarism

Before sending images to the network, two pre-processing steps are often necessary. First, the images must typically be resized to match the size of the input layer of the CNN. The sizes are quite standard from one network to another, with for example 227 × 227 for AlexNet, 224 × 224 for DenseNet, ResNet, and VGG, and 299 × 299 for Inception. Secondly, the images must be normalized to help the model to converge more quickly as well as to better generalize on unseen data (Chollet, 2017). Other pre-processing operations have been proposed. Transformed their images to grayscale. Compared accuracies obtained on grayscale with those from color images. The performance was slightly higher on the color models, with the f1-score improving from 1.34 to 3.33 %. Even if using color images helps the identification process, as the performance decreases only slightly during the grayscale transformation, this highlights that the network relies mainly on other features to identify diseases. In fact, background management is one of the challenging elements in implementing the automatic methods for identifying phytosanitary problems in imagery. With conventional image processing methods, leaf segmentation is a preliminary step to the analysis. The performance obtained by is marginally better with the background, improving the f1-score by slightly <1\%. Since background segmentation is not an option on images taken in the field, and since it is the strength of the CNNs to manage complex backgrounds, background suppression is unnecessary. In order to overcome issues like overfitting, DL algorithms often rely on big datasets. This presents a general obstacle when training algorithms for wider practical usage. Gathering data can be a time-consuming process, which could also require having domain experts for the labeling tasks. In order to enlarge the existing datasets, augmentation techniques are a common approach. Two augmentation strategies were used.Most of these methods include simple transformations such as rotations or pixel-wise changes like blurring or noising in order to introduce distortion to images. In this research, rotations by various angles, and perspective transformations along with size preserving shearing, shifting, and mirroring were applied to the training dataset. The second approach included training GANs responsible for generating syntactic data based on the existing dataset. A plain GAN architecture consisting of two neural networks was used as a participant of the competing game, where the discriminator tries to determine whether the data is real or fake, while the generator tries to create the data with all of the required features to trick the discriminator into thinking that the data are real. By applying this approach, a dedicated GAN could generate novel images that could be used in the training phase of a plant disease classifier. This type of network has proven more successful on different benchmarks than theVariational Auto Encoders (VAEs), Restricted Boltzmann Machines, etc. Over the years, many new GAN architectures have been introduced. In this paper, several architectures were applied to explore the possibility of using syntactic data to train the plant disease classifier. Due to its simplicity in design (5-layer discriminator/generator), DCGAN was used as a proof of concept whether the syntactic data could have features like color, shape, and texture like the real plant leaves. It can be observed that the syntactic data are displaying the desired features.

Sources	Similarity

Figure 8.11: Plagiarism Report for Detailed Design Document

SmallSEOTools

PLAGIARISM SCAN REPORT

Words	80	Date	October 17,2019
Characters	529	Exclude Url	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	4 Unique Sentences

Content Checked For Plagiarism

Proposed system is structured and designed as per sponsor requirement. Thus, the proposed system is estimated to be completed within stated time frame. As part of the contract, the setup acceptance procedures for preliminary and final acceptance of the project results. In order to create enough flexibility supplement this system by a risk management plan, and by integrating change and claim processes into the contract.Thus the system is planned to deploy on severless platform. Hence, concluded project planning phase report.

Sources	Similarity

Figure 8.12: Plagiarism Report for Summary and Conclusion

ANNEXURE A

SPONSORSHIP DETAIL (IF ANY)

Sponsored by ESDS.