VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belagavi-590018, Karnataka

Phase-I

Project work report

On

“Plant Leaf Disease Identification”

Submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science & Engineering

Submitted By

USN	NAME
1BI16CS067	Vivek K
1BI16CS019	Anshuman Chatterjee
1BI16CS198	Gagan Giri
1BI16CS215	Chavva Venkata Pratheek Reddy

Under the guidance of

Tejaswini P S

Assistant Professor

Department of CS&E, BIT

Bengaluru-560 004

For the academic year 2019-20

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BANGALORE INSTITUTE OF TECHNOLOGY

K.R. Road, V.V. Puram, Bengaluru-560 004

ACKNOWLEDGEMENT

The knowledge & satisfaction that accompany the successful completion of any task would be incomplete without mention of people who made it possible, whose guidance and encouragement crowned our effort with success. We would like to thank all and acknowledge the help we have received to carry out phase 1 of this final year project.

We would like to convey my thanks to Head of Department **Dr. ASHA T.** for being kind enough to provide the necessary support to carry out the project.

We are most humbled to mention the enthusiastic influence provided by our guide **Tejaswini P S**, on the project for their ideas, time to time suggestions for being a constant guide and co-operation showed during the venture and making this phase of the project fruitful.

We would also take this opportunity to thank our friends and family for their constant support and help. We are very much pleased to express our sincere gratitude to the friendly co-operation showed by all the **staff members** of the Computer Science Department, BIT.

ABSTRACT

When plants and crops are affected by diseases it affects the agricultural production of the country. Advances in artificial intelligence, image processing and graphical processing units (GPUs) can expand and improve the practice of plant protection and growth. Automatic detection using image processing techniques provide fast and accurate results. But all farmers and agriculturists will not have access to advanced technologies as they are expensive and complicated to utilize. To overcome this, our application can be used to detect plant diseases by capturing leaf images on edge devices.

CONTENTS

	Page. No.
1. INTRODUCTION	1
1.1 Overview	1
1.2 Problem Statement	1
1.3 Existing System	1
1.4 Proposed System	2
1.5 Motivation	2
1.6 Objectives	2
2. LITERATURE SURVEY	3
3. SYSTEM ANALYSIS AND MODELLING	18
3.1 System Analysis	19
4. SYSTEM ARCHITECTURE	20
5. SOFTWARE REQUIREMENT SPECIFICATION	22
5.1 Functional Requirements	23
5.2 Non-functional Requirements	23
5.3 System Requirements	24
6. APPLICATIONS	25
7. REFERENCES	26
8. PROJECT PLANNING REPORT	27

LIST OF FIGURES

Title	Page. No.
2.1 Image processing	3
2.2 CNN layer Architecture	4
2.3 System Architecture	6
2.4 Stages of feature extraction	8
2.5 Disease identification classifier used in the system	11
2.6 CNN layers used in the system	12
2.7 Insect detection procedure	16
3.1 System Model	19
4.1: System architecture for the proposed system	20
8.1 Activity Bar Chart	27

INTRODUCTION

1.1 Overview

The problem of efficient plant disease protection is closely related to the problems of sustainable agriculture. Timely and accurate diagnosis of plant diseases is one of the pillars of precision agriculture. Advances in computer vision present an opportunity to expand and enhance the practice of precise plant protection and extend the market of computer vision applications in the field of agriculture.

Convolution Neural Network needs large amount of training data. To construct best deep CNN model, image augmentation technique is used to enhance the performance of the model. There are different convolutions that are performed in several layers of the Deep CNN. They generate various representations of the training data, starting from more common ones in the first larger layers and becoming more detailed in the deeper layers.

The main important goal of this project is to develop an application that provides high accuracy in detection of plant diseases which can be easily accessed by end users on edge devices.

1.2 Problem Statement

A deep CNN based machine learning model that can be implemented as a smart phone application for the recognition of plant diseases using plant leaf images.

1.3 Existing System

- In [1] the author proposed a method to use machine learning for plant disease incidence and severity measurements from leaf images. The method extracts features from the images and then uses a regression model to make predictions.
- In [2] the author presents a way of analysing plant disease detection using image processing techniques like using image pre-processing, image segmentation, feature extraction and classification.
- In [3] author proposed a method of predicting plant disease using KNN algorithm. This is an unsupervised algorithm that groups the data into groups and each disease is considered a group and the data is segmented accordingly.
- In [4] author proposed an application of decision tree technology for image classification using remote sensing data. The author compared the regression techniques with the decision tree algorithm and achieved a better accuracy with

decision tree.

- In [5] author proposed a vision-based pest detection based on SVM classification method. The author used SVM to achieve a larger decision boundary, so the classification is less prone to errors.
- In [6] The author analyses various concepts of deep learning and deep neural networks showing its strengths in classification and achieving human level performance on tasks.

1.4 Proposed Solution

The smartphone application will be in the form of a single leaf image. Then a Convolution Neural Network model which is trained with data is fed with the input image. This model extracts the required features from the input image and compares them with existing data and identifies the disease.

1.5 Motivation

There are many applications existing in the same domain which need high computational power to get accurate results. We believe that effectiveness of an application is when it reaches the respective end users which in this case is the farmers. These applications are not accessible to our farmers due to expense. Thus, our application bridges the gap between state of the art technologies and farmers.

1.6 Objectives

- To design a CNN model to detect plant disease from leaf image.
- To implement an app to capture and process the leaf image and to obtain the desired output
- To minimize the usage of computational power and reduce the training time for the model.
- To enhance the accessibility to the farmers by reducing expensive and complicated hardware.

Chapter 2

LITERATURE SURVEY

[1] Machine learning for plant disease: Incidence and severity measurements from leaf images. Godliver Owomugisha, 2016

(a) Healthy (b) CBB (c) CGM (d) CMD (e) CBSD

Fig. 1: Sample images associated with the five disease classes of the classification problem.

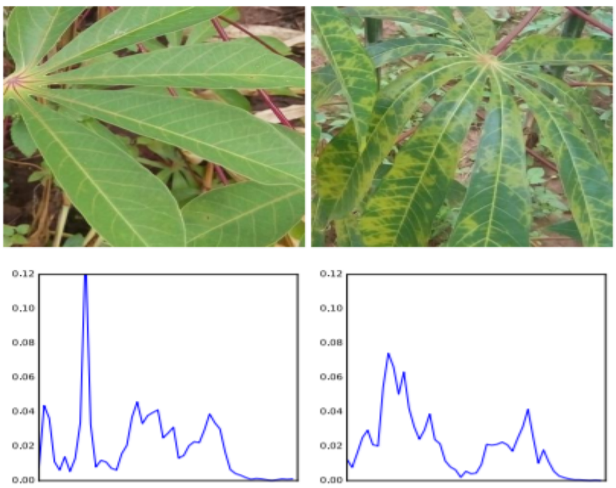

Fig. 2: Examples of histograms (bottom) extracted from the corresponding healthy and diseased images (top).

3: Image with ORB interest keypoints identified

Fig 2.1 Image processing

Methodology:

- THE LEAF IMAGE DATA: The data we used consists of 7,386 images of leaves of cassava plants. The images are in 5 categories; the healthy class of images (1476 examples) and the four classes of diseased images representing the 4 diseases; CMD (3012 images), CBSD (1751 images), CBB (425 images), and CGM (722 images). Figure 1 depicts typical leaf images of the 4 disease classes. For the 4 disease classes, each data subset is broken down further into 4 subsets representing disease severities 2 - 5 (severity level 1 is the healthy class).
- Each of the diseases cause some unique symptomatic features to appear on the leaves

as shown in Figure 1. The four major disease are taken into consideration are ; Cassava brown streak disease (CBSD), Cassava mosaic disease (CMD), Cassava Bacterial Blight (CBB) and Cassava green mite (CGM).

- FEATURE EXTRACTION: The feature extraction process is done using open source feature extraction tools. We thus settled for Color and Oriented FAST and Rotated BRIEF (ORB). Scale Invariant Feature Transforms (SIFT) and Speeded Up Robust Features (SURF) features cannot be used because they are not open source and applicable only on comparatively smaller datasets.
- ORB Combination of two algorithm Features from Accelerated Segment Test (FAST) and Binary Robust Independent Elementary Features (BRIEF) which are used to detect key points
- CLASSIFICATION OF DISEASE INCIDENCE: Scikitlearn, a machine learning toolbox, was used to train the classifiers. Three classifiers were trained:-
  - Linear SVC: A linear Support Vector Classifier was trained on the data. To obtain appropriate algorithm parameters, a grid search over a limited parameter space of C was done for both ORB and color features, C ∈ [1, 10, 100, 1000]
  - KNN: A linear Support Vector Classifier was trained on the data. To obtain appropriate algorithm parameters, a grid search over a limited parameter space of C was done for both ORB and color features, C ∈ [1, 10, 100, 1000].
  - Extra Tree: Extremely Randomized Trees have been shown in the literature to perform well because they average over very many weak learners on various subsamples of the data. We find the appropriate number of trees in the forest to use using grid search of 5 parameters for ORB features
- CLASSIFICATION OF DISEASE SEVERITY: We split up each of the classes into 4 subclasses; the healthy class, severity level 2, severity level 3 and severity level 4; severity-4 possessing the most severe symptoms of the 4.

LIMITATION:

- The color is not the best feature because all the leaves affected by the disease turn yellow.
- The smart phone application is based on client server architecture which is costly to maintain.
- It uses SVM classifier which requires to find a lot of parameters to obtain the best efficiency.

[2] Plant Disease Detection Using Image Processing. Sachin D. Khirade,2015

ARCHITECTURE:

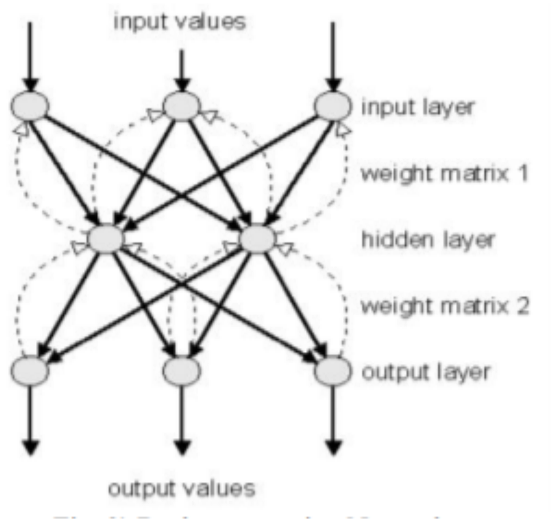

Fig 2.2 CNN layer Architecture

METHODOLOGY:

- The images of the plant leaf are captured through the camera. This image is in RGB form.
- To remove noise in image or other object removal, different pre-processing techniques is considered.
  - Image clipping i.e. cropping of the leaf image to get the interested image region.
  - Image smoothing is done using the smoothing filter.
  - Image enhancement is carried out for increasing the contrast. the RGB images into the grey images using colour conversion using equation f(x)=0.2989*R + 0.5870*G + 0.114.*B
- Image segmentation means partitioning of image into various part of same features or having some similarity.
- The segmentation can be done using various methods like Otsu threshold algorithm, Boundary and spot detection algorithm, K-means algorithm, converting RGB image into HIS model
- Segmentation using Boundary and spot detection algorithm: The RGB image is converted into the HIS model for segmenting. Boundary detection and spot detection helps to find the infected part of the leaf using 8 connectivity of pixels.
- K-means algorithm : The K-means clustering is used for classification of object based on a set of features into K number of classes. The classification of object is done by

minimizing the sum of the squares of the distance between the object and the corresponding cluster.

- Otsu threshold algorithm : Thresholding creates binary images from grey-level images by setting all pixels below some threshold to zero and all pixels above that threshold to one then find the mean of each cluster. Square the difference between the means. Multiply the number of pixels in one cluster times the number in the other.
- They have found that morphological result gives better result than the other features. Texture means how the colour is distributed in the image, the roughness, hardness of the image. It can also be used for the detection of infected plant areas.
- Feature extraction is done by using colour co-occurrence method and leaf colour extraction using H and B components.
- Color co-occurrence Method: In this method both color and texture are taken into account to get an unique features for that image. For that the RGB image is converted into the HSI translation.
- Leaf color extraction using H and B components: The input image is enhanced by using anisotropic diffusion technique to preserve the information of the affected pixels before separating the color from the background. To distinguish between grape leaf and the non-grape leaf part, H and B components from HIS and LAB color space is considered. A back propagation neural network is implemented to recognize colors of disease leaf.
- Classification is done by Back propagation algorithm.
- BPNN algorithm is used in a recurrent network. Once trained, the neural network weights are fixed and can be used to compute output values for new query images which are not present in the learning database

LIMITATION:

- Based on the manual filtering tool i.e.; HSI Translations
- RGB Images are converted into grey scale images at preprocessing stage
- The biggest limitation with Otsu is its assumption of binary classes: It partitions the grayscale histogram to two classes. However in real-world segmentation problems we most generally deal with images having more than two class of segments.

[3] Prediction of bacterial associations with plants using a supervised machine-learning approach. López-Solanilla, Martínez-García, 2019

ARCHITECTURE:

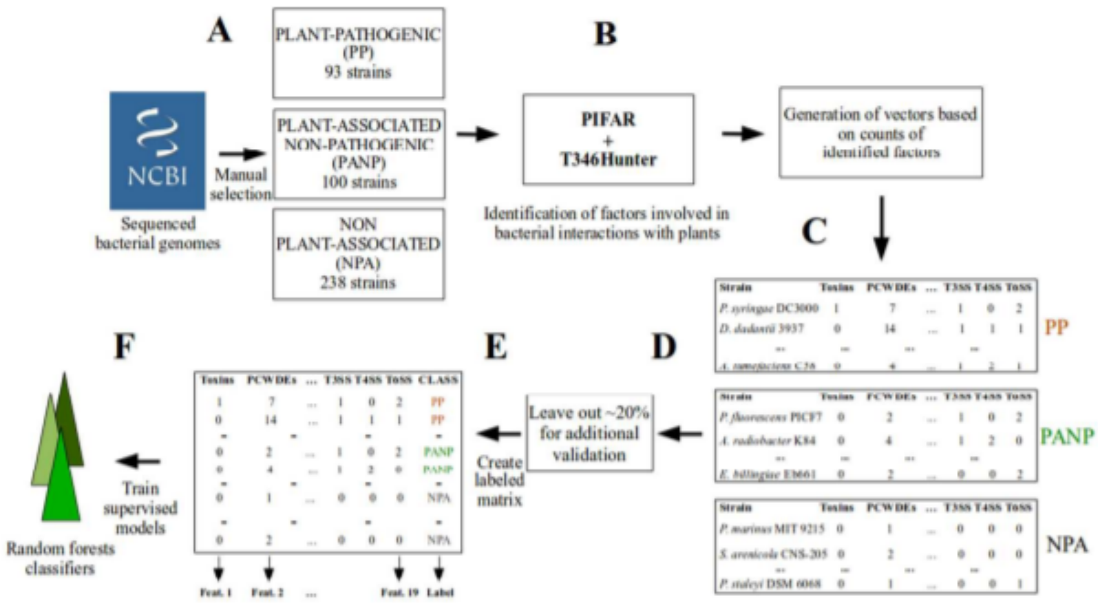

Fig 2.3 System architecture

Methodology:

- To develop our supervised machine-learning approach, we used a manually curated database of plant-bacteria interaction factors, which was deposited in a freely accessible web-server called PIFAR (Plant-bacteria Interaction Factors Resource).. The website has two main purposes: to maintain comprehensive information on gene products that have been implicated in bacterial interactions with plants and to help researchers identify these products in input genome sequences.
- Factors were selected by manual inspection of the available scientific literature. In the case of virulence, a gene is considered a component of a factor when there is reported evidence of diminished virulence in the corresponding mutant. Similar criteria were adopted for other type of interactions.
- Three classifiers were trained to discriminate between different bacterial lifestyles. We selected 431 sequenced bacterial strains that were separated into three classes: plant-pathogenic, plant-associated-non-pathogenic and non-plant-associated. This classification was based on literature searches and NCBI annotations

- Then, we performed a systematic screening of our database against these strains. For each genome that we analyzed, we obtained its repertoire of plant-associated bacterial factors. Based on the presence/absence of these factors, a vector was generated for each genome so that each position contained the count of the identified factors by type.

LIMITATION:

- It is completely a data centric approach and all the data are present in the server.
- It is a client server model and cannot be used in the remote areas where internet facility is not good.
- Three models that are built are specific to the data that are present on the server.

[4] Deep neural network-based Recognition of Plant Diseases by Leaf Image Classification. Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, Dubravko Culibrk, and Darko Stefanovic

Methodology:

- The input image is taken as pixels in a floating-point array and processed.
- Each layer of the network acts as a filter or kernel that extracts a specific component of the Image as shown below

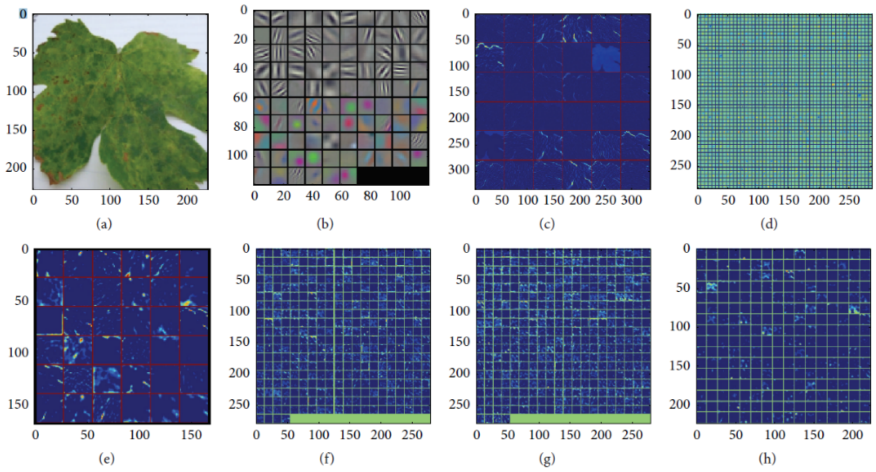

Fig 2.4 Stages of feature extraction

- Each kernel works by skipping a specific number of pixels from the start of the image. The number of pixels skipped is decided by the formulae below.

$$M_x^n = \frac{M_x^{n-1} - K_x^n}{S_x^n + 1} + 1,$$

$$M_y^n = \frac{M_y^{n-1} - K_y^n}{S_y^n + 1} + 1,$$

- This way of using the above shown skipping rule along with the filters or kernels is called “feature maps”.
- Then a Sigmoid activation function is used on the values and passed to the next layer.
- Then the layer is normalized so that there are no overblown or nullified weights.
- The architecture used is the CaffeNet Architecture with slight modification made to fit the prediction desired in this case.

Limitations

- The dataset used is not vast hence there is a chance of overfitting.
- The proposed mobile application uses server components that cost a lot to build and maintain
- The server-side components also add latency to the prediction generated by the model on the mobile application.
- Using a pre-built model like CaffeNet can be a waste of resources to train as it is a very huge model used for predicting a vast variety of classes
- Instead using a custom-built model will produce better results with lesser compute resources to train.

[5] A Review of advanced techniques for detecting plant diseases.

Sindhuja Sankaran, Ashish Mishra, Reza Ehsani, Cristina Davis

Methodology:

- This paper compares various techniques used to detect various diseases in plants.
- The authors divided the techniques into two categories direct methods and indirect methods.
- Direct methods are more biological whereas indirect methods are more of scientific and computer-based methods.
- The indirect techniques include techniques such as spectroscopic techniques and Imaging techniques.
- The spectroscopic Imaging techniques include visible light, infrared fluorescence and multispectral bands for analysing a leaf.
- The imaging techniques include Fluorescence and hyperspectral imaging techniques for detecting plant diseases.
- First the images are scanned using the above-mentioned techniques and then various statistical methods are applied on the obtained data.
- The statistical methods include analysis of variance, correlation and regression analysis.
- More complex techniques like Linear regression, principal component analysis and discriminant analysis are used based on the spectrum of the light used.

Limitations

- The imaging techniques used are very complex like using infrared light or x-ray spectrograph.
- These techniques require the use of high-end machines that require experts being on the field to collect data in real-time.
- The methods specified are not scalable and cannot be used in a large scale.

[6] Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case. Johannes A, Picon, Alvarez-Gila, Echazarra, Rodriguez-Vaamonde, Navajas AD

Architecture:

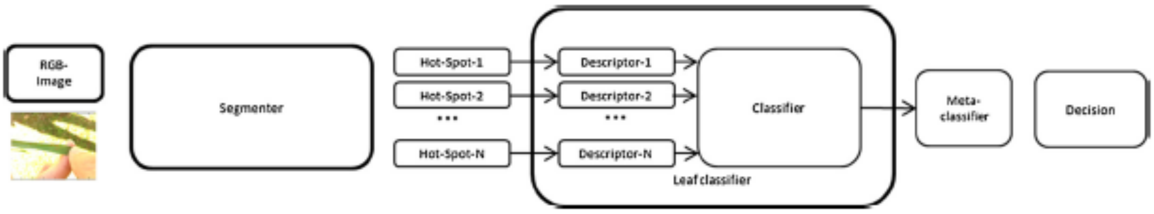

Fig 2.5 Disease identification classifier used in the system

Methodology:

The RGB image of the leaf captured from the mobile's camera is sent through the segmentor for primary segmentation. It is done in order to detect and select any suspicious sub-region on the leaf as a disease candidate (Hot-Spots). Once each Hot-Spot is properly identified, a metaclassifier module weights all individual decision to make a global assessment. Each disease candidate is then analyzed by the extraction of specific visual features of the candidate region and the use of a specific statistical inference model for the specific disease. The following steps are used:

a) Image preprocessing: The acquired image is processed by means of color constancy algorithms to minimize the natural illumination variability effects.

b) Disease identification algorithm that comprises the following steps:

- Extraction of disease candidate regions: The segmented leaf isolated image is corrected to achieve color constancy, normalized and candidate sub-regions susceptible of containing diseases, called Hot-Spots, are extracted.
- Each extracted Hot-Spot region is analyzed in detail by local descriptors that extract and categorize each region in terms of its visual characteristics. Next, each Hot-Spot is checked against different disease detection inference models.
- All this information is gathered and processed by a high-level classifier, called meta-classifier, that is able to extract the complementarities of the different inherent features that are embedded within an image.

Limitation:

- The proposed system is limited to wheat plant and uses client server architecture.
- Maintaining a server is costly.

[7] Deep learning for plant identification using vein morphological patterns. GrinblatGL, UzalLC, LareseMG, GranittoPM.

Architecture:

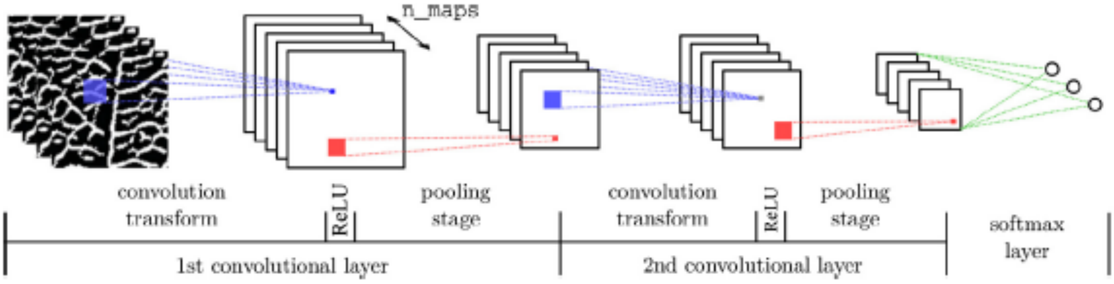

Fig 2.6 CNN layers used in the system

- A three-layer version of the CNN is considered in this paper.
- The first two are convolutional layers and the last one is a SoftMax layer and has 6 layers.

Methodology:

In this paper, the introduction of a CNN avoids the use of handcrafted feature extractors. The traditional classification algorithms are replaced by a deep convolutional network. This deep learning approach improves the accuracy of detection and shows increase as the model depth increases. CNN can automatically learn from the training set the appropriate features to solve the classification problem. vein morphological patterns act as a leaf fingerprint. The set of images of first foliage leaves acquired with a standard flatbed scanner is sent through the convolutional layer which consists of stages like convolutional transform and pooling stage and the classification done at the soft max layer.

Limitation:

This model is limited to plants for which plants can be identified by their veins and doesn't predict disease.

[8] Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance. T. Rumpf, A.-K. Mahlein, U. Steinerb, E.-C. Oerkeb.

The main aim of this paper is

- to discriminate diseased from non-diseased sugar beet leaves,
- to differentiate between the diseases Cercospora leaf spot, leaf rust and powdery mildew, and
- to identify diseases even before specific symptoms became visible.

Methodology:

1. Greenhouse experiments were conducted to assess spectral characteristics of sugar beet leaves under controlled conditions.
2. Healthy leaves were kept inoculated with the pathogens Cercospora beticola, Uromyces betae or Erysiphe betae causing Cercospora leaf spot, sugar beet rust and powdery mildew respectively for a period of 21 days.
3. Different methods were used for the differentiation between the two classes, non-inoculated, healthy leaves and leaves inoculated with one of the three leaf pathogens.
4. VIs calculated from characteristic reflectance spectral and the SPAD-value was used for classification. The results showed that the specificity of the classification was always lower than the sensitivity
5. In comparison with the classification results of Decision Trees and ANNs the classification error of SVMs was always lower.
6. As a conclusion SVM classification provides better accuracy. In addition high performance in learning the best model, the comparison of the different classifiers shows that SVMs use the inherent information of the vegetation indices in an optimal way.

Limitations:

- Difficulties emerge especially at early development stages of the characteristic symptoms. Slight variations can be accounted for the original source of the reflectance data. This results in a number of problems that are typical for such single point measurements.
- Due to the very small size of sugar beet rust colonies (0.5–1.5 mm), a precise classification at early stages or in the case that only few pustules occur is very complex. Also, a distinctive detection of powdery mildew at early stages is challenging.

[9] Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. Yu-Dong Zhang & Zheng chao Don & Xianqing Chen

Methodology:

In this work, data preprocessing of dataset was done in four steps: First, we move the fruit into the center of the image. Second, the image was cropped and resized to a 256 × 256 matrix. Third, split-and merge algorithm was used to remove the background. Fourth, each image was labelled manually to one of the 18 fruit types. Three different data augmentation types used were image rotation, gamma correction and noise injection.

The structure of our CNN is a 13-layer deep neural network. The image input layer just inputs the preprocessed fruit image directly. The fully connected (FC) layer multiplies the input by a weight matrix and then adds a bias vector. The softmax layer used the softmax function, also known as the multiclass generalization of logistic regression.

Limitations:

- Dataset used in the work is clean and hence it does not give similar accuracy on imperfect images.
- Time taken for training and computation is more as it is a 13 layer CNN structure.

[10] Vision-based pest detection based on SVM classification method [10]

Methodology:

All images were obtained from a strawberry greenhouse by a robot that moved along the pots row and it displaced the computer vision system in a horizontal direction. The non-flower regions of captured images are considered as background and removed by applying the gamma operator. The gamma operator, c, enhances the contrast of the brighter regions. Histogram equalization and contrast stretching was used to remove any remaining background. As supplementary stage to remove background, holes created inside the target were filled using the filling operation. After detecting the flowers, the flowers were obtained as background and pests were obtained as target.

This work uses Support Vector Machines(SVMs) to detect of thrips on the crop canopy images using SVM classification method. Mean square error (MSE), root of mean square error (RMSE), mean absolute error (MAE) and mean percent error (MPE) were used for evaluation of the classification.

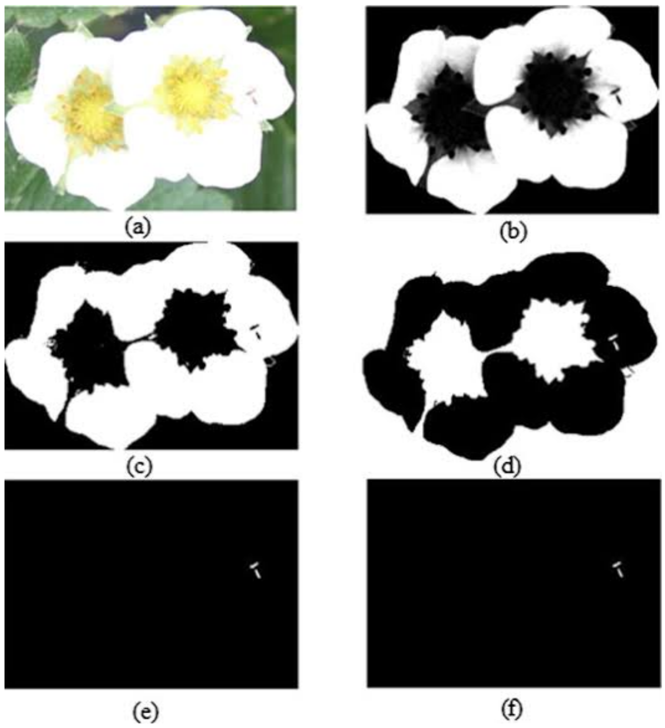

Fig 2.7 Insect detection procedure. (a) Original image; (b) Result of gamma operator; (c) Converting to binary; (d) Reversing image; (e) Extraction pest; (f) Color pest image.

Limitations

- SVM takes huge compute power to run and manual filters like gamma correction should be made precise.
- It uses hand craft features like tanh and RBF and p degree polynomial for activation.
- Dataset consisted of images taken is laboratory setting.

Chapter 3

SYSTEM ANALYSIS AND MODELLING

The system is targeted to be useful for end customers and not an enterprise. This leads to a design philosophy that has to be simple yet sophisticated. The users should get the maximum features that they will be willing to use and at the same time, the system must be easy to use.

3.1 System Analysis

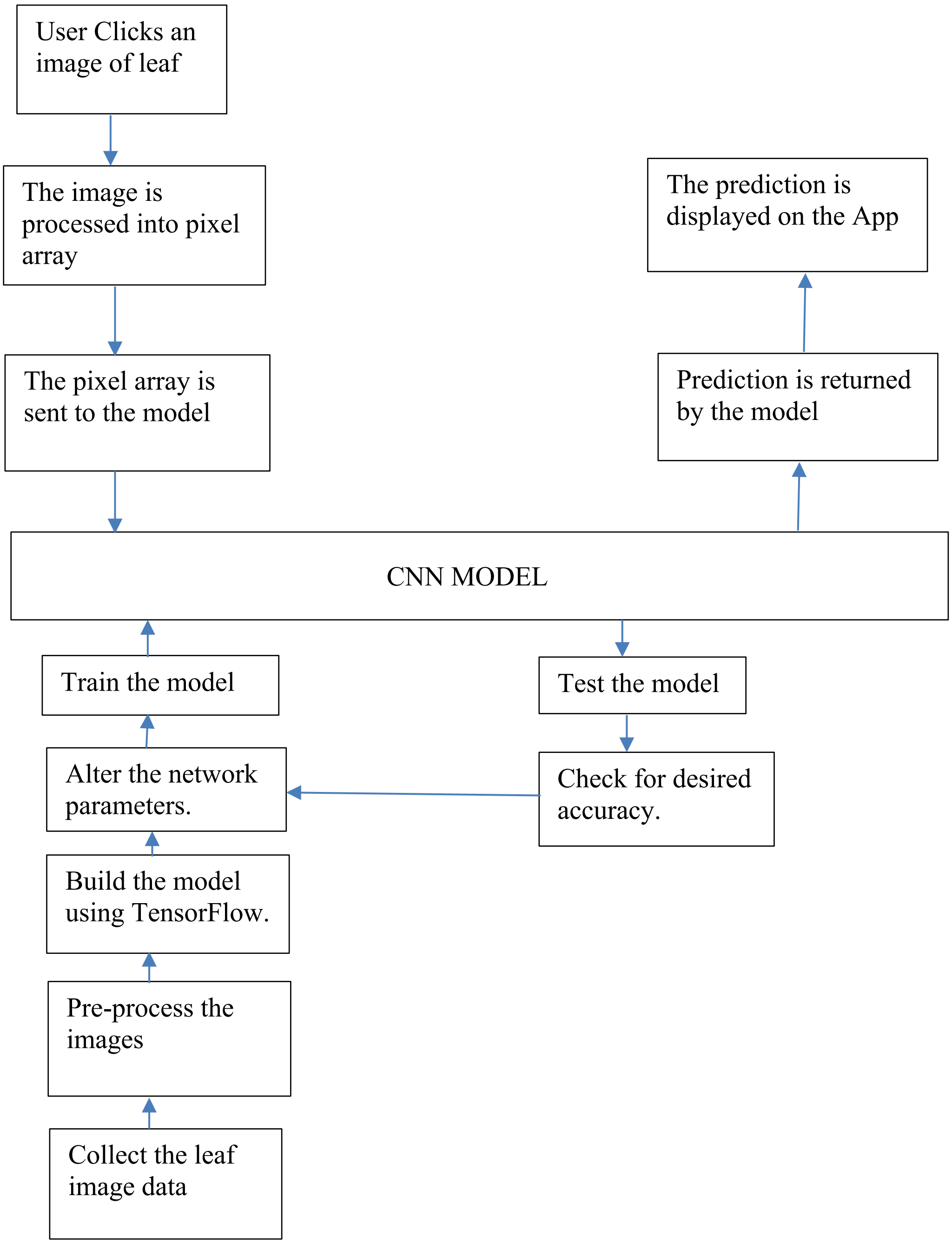

Fig 3.1 System Model

- The model is a CNN model built with 8 layers of convolution and 8 layers of pooling.
- The image is first pre-processed, techniques like normalization, grey-scaling, etc are used.
- Then the model is built using TensorFlow.
- Various network parameters like optimizer, loss function is altered and the model is compiled.
- Then the model is trained with the data.
- Then a TF to TFlite converter is used to quantize and convert the model to run it on android.
- Then the model is deployed on to android and functions to predict are written.

Chapter 4

SYSTEM ARCHITECTURE

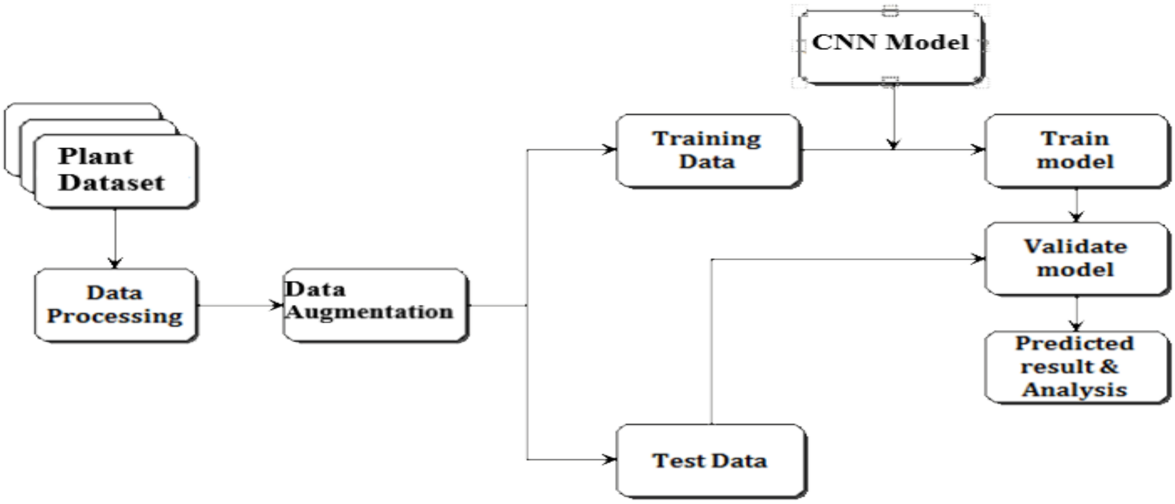

Fig 4.1: System architecture for the proposed system

The platform shown in the above figure consists of two primary phases. These two phases are the pillars of our model. They are,

- Upload phase.
- Fetch phase.

Upload phase

This is the first phase of the application. Here, a user uploads an analog document, with the intention to convert the uploaded analog document to the digital document and store it into the centralized database for the further retrievals. First, the user uploads the analog document, which is fed given as input to the OCR engine. In the OCR engine, the characters in the analog document are read and converted into an annotated document, which is returned as an output. This annotated document is later passed on to the Keyphrase engine as an input, so as to extract the key phrase from the document for easier fetching and categorizing of the documents. Once the keyphrases are extracted, the annotated document along with the corresponding keyphrases are stored in the centralized storage unit.

Fetch phase

This is the second phase of the application. The user tries to fetch the documents from centralized storage using key phrases. Here we see that a user enters a keyphrase i.e., the search word into the search engine hoping to fetch the documents related to the keyphrase in the database. The keyphrase engine not only checks the search word but also looks for the context of the search for the more accurate and reliable retrieval of documents. The documents corresponding to the keyphrases are returned as the output of the developed module.

Chapter 5

SOFTWARE REQUIREMENT SPECIFICATIONS

A software requirements specification (SRS) is a description of a software system to be developed. It is modelled after business requirements specification (CONOPS), also known as a stakeholder requirements specification (SRS). The software requirements specification lays out functional and non-functional requirements, and it may include a set of use cases that describe user interactions that the software must provide.

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function (in a market-driven project, these roles may be played by the marketing and development divisions). Software requirements specification is a rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules. Used appropriately, software requirements specifications can help prevent software project failure.

The software requirements specification document lists sufficient and necessary requirements for the project development. To derive the requirements, the developer needs to have a clear and thorough understanding of the products under development. This is achieved through detailed and continuous communications with the project team and customer throughout the software development process.

The SRS may be one of a contract deliverable Data Item Descriptions or have other forms of organizationally-mandated content.

5.1 Functional Requirements

Capturing leaf image from edge device

User should capture a image for prediction of the disease.

Generating prediction from CNN model

CNN model should be trained with proper training data.

To increase the efficiency in feature extraction and to detect the disease

Efficiency of the trained model should be ideal.

TensorFlow Lite

TensorFlow Lite is used to transfer the model into application which can run on edge devices.

5.2 Non-Functional Requirements

Performance

For good user experience, the processing of the model should be expeditious.

Accuracy

The prediction of the disease should be accurate so that farmers give proper diagnosis to the plants.

Integrity

The final developed system should be designed in a way that it is coherent and should be available at any time.

Availability

Farmers should be able to use the app without any need of internet in remote areas.

Usability

User interface should be unsophisticated so that any user can use it with ease.

5.3 System Requirements

Hardware Requirements

- Desktops/Laptops running a windows/Linux OS
- Android mobiles running Android 8.0+

Software Requirements

- Ubuntu 18.10+
- Android 8.0+
- Windows 10+
- TensorFlow
- Android Studio
- Visual Studio Code
- Git/GitHub
- Signal/Slack

Chapter 6

APPLICATIONS

- Improve the practice of precise plant protection and growth.
- Agricultural production cost can be significantly decreased.
- Farmers can monitor the plants time to time.
- It can be used on low powered devices like Android.
- It can be used in integration with IOT devices like drones for live monitoring of plants.

REFERENCES

[1] Godliver Owomugisha and Ernest Mwebaze, Machine learning for plant disease: Incidence and severity measurements from leaf images. In2016 15th IEEE International conference on machine learning and Applications

[2] Khirade SD, Patil AB. Plant disease detection using image processing. In: 2015 International Conference on Computing Communication Control and Automation.; 2015. p. 768–71. doi: 10.1109/ICCUBEA.2015.153.

[3] Martínez-García PM, López-Solanilla E, Ramos C, Rodríguez-Palenzuela P. Prediction of bacterial associations with plants using a supervised machine- learning approach. Environ Microbiol 2016;18(12):4847–61. doi: 10.1111/1462-2920.13389

[4] Sladojevic S, Arsenovic M, Anderla A, Dubravko Culibrk DS. Deep neural networks-based recognition of plant diseases by leaf image classification. Comput Intell Neurosci 2016; 2016:1–12. doi: 10.1155/2016/3289801

[5] Sankaran S, Mishra A, Ehsani R, Davis C. A review of advanced techniques for detecting plant diseases. Comput Electron Agric 2010;72(1):1–13. doi: 10. 1016/j.compag.2010.02.007

[6] Johannes A, Picon A, Alvarez-Gila A, Echazarra J, Rodriguez-Vaamonde S, Navajas AD, et al. Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case. Comput Electron Agric 2017;138:200–9. doi: 10.1016/j.compag.2017.04.013

[7] Grinblat GL, Uzal LC, Larese MG, Granitto PM. Deep learning for plant identification using vein morphological patterns. Comput Electron Agric 2016;127:418–24. doi: 10.1016/j.compag.2016.07.003

[8] Rumpf T, Mahlein A-K, Steiner U, Oerke E-C, Dehne H-W, Plümer L. Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance. Comput Electron Agric 2010;74(1):91–9. doi: 10.1016/j.compag.2010.06.009

[9] Yu-Dong Zhang & Zhengchao Don & Xianqing Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation.

[10] Ebrahimi MA, Khoshtaghaza MH, Minaei S, Jamshidi B. Vision-based pest detection based on SVM classification method. Comput Electron Agric 2017;

[11] Plant Disease Detection Using CNNs and GANs as an Augmentative Approach. 2018 Rutu Gandhi.

[12]

PROJECT PLANNING REPORT

Task	Task Description	Duration	Start Date	End Date
T1	Problem identification and analysis	7	29-Jul-19	4-Aug-19
T2	Requirement elicitation and analysis	7	5-Aug-19	11-Aug-19
T3	Literature review	28	12-Aug-19	8-Sep-19
T4	Project proposal synopsis	7	9-Sep-19	15-Sep-19
T5	Dataset acquisition	14	16-Sep-19	29-Sep-19
T6	Pre-synopsis presentation	7	30-Sep-19	6-Oct-19
T7	Proposed block diagram	14	7-Oct-19	20-Oct-19
T8	Preprocessing 1(Feature Extraction)	14	21-Oct-19	3-Nov-19
T9	Preprocessing 2(Noise removal)	14	4-Nov-19	17-Nov-19
T10	Phase 1 final presentation	7	18-Nov-19	24-Nov-19
T11	Phase 1 report	5	25-Nov-19	30-Nov-19
T12	Detection	28	1-Feb-20	28-Feb-20
T13	Interim presentation	7	29-Feb-20	7-Mar-20
T14	Classification	28	8-Mar-20	4-Apr-20
T15	Verification and validation	14	5-Apr-20	18-Apr-20
T16	Front end application	7	19-Apr-20	25-Apr-20
T17	Draft report	7	26-Apr-20	2-May-20
T18	Project delivery and presentation	7	3-May-20	9-May-20
T19	Final report	7	10-May-20	17-May-20

Table 8.1 Task Description Table

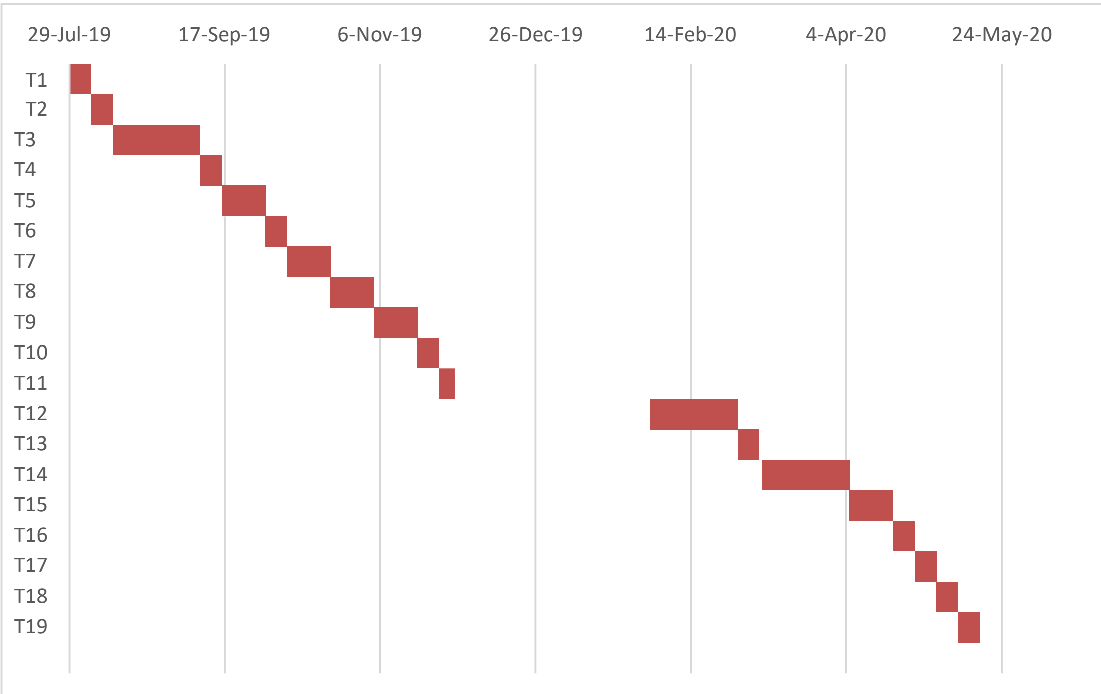

Fig 8.1 Activity Bar Chart