

DEEP NEURAL NETWORK BASED RECOGNITION OF COTTON DISEASE DETECTION BY LEAF IMAGE CLASSIFICATION

Submitted in partial fulfillment of the requirements for the award
of Bachelor of Engineering degree in Computer Science and
Engineering

By

VIKHIL KUMAR KOLIPAKA (Reg. No. 37110348)

&

JALA MALLESH (Reg. No. 37110282)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

**Accredited with Grade "A" by NAAC JEPPIAAR
NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119**

MARCH - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **VIKHIL KUMAR KOLIPAKA (Reg. No. 37110348)** and **JALA MALLESH (Reg. No. 37110282)** who carried out the project entitled **"DEEP NEURAL NETWORK BASED RECOGNITION OF COTTON DISEASE DETECTION BY LEAF IMAGE CLASSIFICATION"** under my supervision from March 2021 to April 2021.

Internal Guide

Dr.S.Revathy,M.E.,Ph.d,

Head of the Department

Dr. S. VIGNESHWARI M.E., Ph.D., AND Dr. L. Lakshmanan, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

internalExaminer

External Examiner

DECLARATION

I **VIKHIL KUMAR KOLIPAKA (Reg.no 37110348)** and **JALA MALLESH (Reg.no 37110282)** hereby declare that the Project Report entitled **“DEEP NEURAL NETWORK BASED RECOGNITION OF COTTON DISEASE DETECTION BY LEAF IMAGE CLASSIFICATION”** is done by me under the guidance of **Dr.S.Revathy,M.E.,Ph.d.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:16/04/2021

PLACE: CHENNAI

(K.VIKHIL KUMAR)
SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA INSTUTITE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean, School of Computing, Dr. S. Vigneswari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. S. Revathy, M.E., Ph.d.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

The cotton plant disease detection is that the method of sleuthing disease by analyzing their visual properties. The visual properties extraction method from the photographs is understood because the feature extraction. The feature extraction method may be done victimization the varied feature descriptors like SIFT, SURF or different most fitted candidate. The feature descriptors square measure then passed to the classifier for the analysis of the feature. The classifier is that the algorithmic program, that is employed to classify the feature on the premise of its similarity with the coaching dataset. The coaching dataset is that the assortment of options antecedently extracted from the better-known objects (the leaves with specific sickness during this case). The leaves with sickness square measure classified on the premise of their similarity with the coaching dataset of sickness samples antecedently represented by the feature descriptors. during this paper, our aim is to unravel the cotton sickness detection drawback victimization the image process techniques mechanically from the input image. The sickness classification can based upon the visibility of the sickness on the cotton leaves, that any may be used for the identification victimization the classifier. The planned model implementation would be done victimization the MATLAB machine and also the planned model results would be obtained within the kind of the accuracy, precision, recall, period of time and lots of different similar parameters.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	viii
1.	INTRODUCTION	
	1.1 INTRODUCTION	1
2.	LITERATURE SURVEY	6
3.	AIM AND SCOPE	8
	3.1 AIM OF PROJECT	8
	3.2 <i>OBJECTIVES</i>	8
	3.3 SCOPE	8
4.	EXPEIMENTAL METHODLOGY AND ALGORITHMS	9
	4.1.1 PRE-PROCESSING	9
	4.1.2 PREPROCESSING OF COTTON LEAF IMAGE	10
	4.1.3 IMAGE SEGMENTATION	10
	4.1.4 FEATURE EXTRACTION	11
	4.1.5 DISEASE IDENTIFICATION	12
	4.1.6 CONTRAST ENHANCEMENT	14
	4.1.7 IMAGE OPTIMIZATION	16
	4.1.8 CLASSIFICATION	16
	4.1.9 IMAGE ENHANCEMENT	17

	4.2.1 METHODOLOGY	18
	4.2.2 DATA PRE-PROCESSING	19
	4.3.3 FEATURE EXTRACTION	20
	4.4.4 SEGMENTATION	20
	4.3 DATASET DESIGN	22
	4.3.1 TRAINING PHASE	22
	4.3.2 DATA ORIGIN AND CHARACTERISTICS	23
	4.3.3 CLASS TAXONOMY DEFINITION	26
	4.3.4 Class Taxonomy Definition	27
	4.3.5 DATA ANNOTATION	28
	4.3.6 DATASET DIVISION	29
	4.4 COLOR TRANSFORMATION STRUCTURE	30
	4.5 HUE COLOR ATTRIBUTE	31
	4.6 MASKING INEXPERIENCED PIXELS	32
	4.7 COLOR CO-OCCURRENCE TECHNIQUE	33
5	RESULTS AND DISCUSSION	35
	SCREENSHOTS	36
6.	CONCLUSION AND FUTURE WORK	38
	CONCLUSION	38
	FUTURE WORK	38
	REFERENCES	40
	APPENDIX	41
	SOURCE CODE	41

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
4.1	SYSTEM ARCHTICTURE	9
4.2	LEAFS OF DISEASED COTTON PLANT	10
4.3	LEAFS OF FRESH COTTON PLANT	10
4.4	BLOCK DIAGRAM	18
5.1	GRAPH OF TRAINING ACCURACY VS TEST ACCURACY	35
5.2	GRAPH OF TRAINING LOSS VS TEST LOSS	36
5.3	SCREENSHOT OF OUTPUT PAGE 1	36
5.4	SCREENSHOT OF OUTPUT PAGE 2	36
5.5	SCREENSHOT OF OUTPUT PAGE 3	37
5.6	SCREENSHOT OF OUTPUT PAGE 4	37

CHAPTER - 1

INTRODUCTION

1.1 INTRODUCTION

Mostly in remote villages, a farmer seeks agro recommendation on sickness or pester by describing visible symptoms to associate knowledgeable over the phone. During this case, data might be misinterpreted. What is more the work time are often giant as specialists aren't accessible in giant numbers. The opposite manner is to physically carry a specimen to the knowledgeable that is each time intense and expensive. Instead, farmer visits the native search keeper commerce agri product and ask for recommendation. However, industrial interests of the search keeper might interfere and therefore the farmer might receive an upscale and / or ineffective answer. Additionally, the believability of such recommendation isn't warranted. It's a usual apply by farmers to discover sickness while not victimization technology and it's tough to measure severity of the sickness [3]. Hence, there's a requirement for readying of an answer, which may offer authentic, correct and timely agro advices to a farmer at an occasional value. This paper may be a part of add progress towards this direction. India is that the world's largest cotton producer [13]. It's been calculable by the Cotton Association of {india|India|Republic of India|Bharat|Asian country|Asian nation} (CAI) that India has created thirty five.7 million bales (1 bale = one hundred seventy Kilogram) of cotton in 2015-16. The Gujarat state may be a main contributor with production of nine.65 million bales. These giant numbers square measure in spite of white fly attack in geographical region state and pink ball worm onslaught in Gujarat throughout this year. Additionally pink ball worm appears to possess affected genetically changed Bt cotton [25]. In post Bt cotton era, thanks to widespread cultivation of hybrid crop, management practices have modified and have become a lot of input-intensive [8]. The cotton diseases square measure classified as: 1) microorganism, e.g. gray mildew, microorganism blight, crown gall; 2) fungous e.g. anthracnose, leaf spot, alternaria; 3) infective agent, e.g. leaf curl, leaf crumple, leaf roll; 4) pester, e.g. white flies, pink ball worm. Majority of the sickness symptoms

seem on the leaf of the plant. regarding 2 hundredth to twenty fifth of the cotton production is lost thanks to diseases [1]. The leaf displays most sickness symptoms with amendment in color, texture, and morphology. The patterns varies with growth of the plant and depends on external factors like temperature, lack, weather conditions. machine-driven detection of sickness during a plant has been tried in past few years. Recent engineering innovations in sensors, control, remote sensing technologies, laptop vision and artificial intelligence have generated suggests that for preciseness agriculture to scale back human effort and build it a lot of economical [32]. Visual image based mostly sickness detection plays an important role in effective plant management. during a machine vision based mostly answer, image is divided into its constituents objects, appropriate options square measure extracted from the foreground and sickness region is classed. The system planned in [21] delivers personalised agro-advisory victimization text and pictures. Today, satellite based mostly imaging solutions like the "SIBWA" [7], "Quick Bird" [15] square measure gaining quality. "SIBWA" uses terribly High Resolution imagination (VHRI) of land to estimate variations of soil fertility, land size and form whereas fast Bird became terribly attention-grabbing for remote sensing analysis [11] by grouping reflection factor knowledge from 450 metric linear unit higher than the planet surface. On the opposite hand, mobile services ("mKrishi", Nokia Life tools, IFFCO Kisan Sanchar Limited) [15] and ICRISAT inexperienced phablet square measure gaining quality in India for providing agricultural extension, social property and resource. Tele-centers like "esagu" [21], [15] maintains databases for queries that square measure solved victimization automation tools or by associate expert's facilitate. The "eAGROBOT" (a prototype) may be a ground {based|based mostly|primarily based mostly} agricultural golem that overcomes challenges of satellite based systems and help-desk solutions in m-Services [20]. several existing machine vision based mostly strategies use form as a feature to section the image and search them from remote location [31]. Fourier descriptors square measure the favored leaf characteristics once handling form [16], [30]. The circumstances of similarity transformations, presence of noise, and occlusion are taken into thought in [24]. however whereas handling naturally captured pictures, overlapping leaves makes form detection tough and so existing form recognizing algorithms in visual domain don't work with efficiency. though underneath sure overlapping conditions of tiny plants active form model are used to discover form in [19], [28]. additionally

spectral pictures and stereo vision techniques are introduced to resolve issues of leaf extraction and segmentation of overlapping leaves [18]. a number of the opposite approaches handling overlapping leaves square measure as follows. Author [12] developed associate algorithmic program to extract the leaf from a plant image mechanically during a controlled condition. more it cleans the leaf image of unwarranted objects. It reportable eighty six accuracy in extracting a minimum of one clean leaf from twenty one plant pictures. Authorsin [14] developed the mobile application for identification of the tree species from pictures of their leaves. They used machine vision to section leaves from the background, extract leaf curvature and determine the tree species. The algorithmic program runs a binary classifier to 1st section leaf and non-leafy half. If this step detects that input image don't have a single leaf on a light-weight and plain background, the user is asked to require the image once more with higher than conditions. The algorithmic program presently covers 184 tree species of north jap North American nation. Authors in [26] segments pictures of a plant within the field and performs classification. It additionally estimates the three- dimensional(3D) plant data for higher segmentation. The system needs a lot of then one image of the plant for 3D reconstruction. additionally the leaves to be divided got to be within the center of the image. Authors in [4] conferred a theme for observation plant leaves victimization infrared camera and golem. The system are often used for machine-driven plant searching that is useful for detection of disease or mineral deficiencies. The planned answer is predicated on many assumptions; one. Boundaries of the leaves square measure visible; a pair of. Leaves square measure giant for analysis; three. The leaves remains stationary during searching. 4. The searching needs controlled setting. In [18] a system for leaf extraction from the cover has been developed victimization fuzzy bunch and genetic algorithmic program (GA). The non occluded individual leaves square measure reassembled victimization GA. The 83.5% plants within the second week of testing made a minimum of one pc extracted leaf for identification by ancient options like form, texture or color. The authors in [27] propose a leaf image classification with the difficult background supported previous form data. Then Hu geometric and Zernike moments square measure wont to extract the form options. Twenty leaf categories square measure with success known with a mean accuracy of ninety two.6%. what is more within the ending of most cotton sickness, leaves dry out and becomes brown. Color of

spectral pictures and stereo vision techniques are introduced to resolve issues of leaf extraction and segmentation of overlapping leaves [18]. a number of the opposite approaches handling overlapping leaves square measure as follows. Author [12] developed associate algorithmic program to extract the leaf from a plant image mechanically during a controlled condition. more it cleans the leaf image of unwarranted objects. It reportable eighty six accuracy in extracting a minimum of one clean leaf from twenty one plant pictures. Authorsin [14] developed the mobile application for identification of the tree species from pictures of their leaves. They used machine vision to section leaves from the background, extract leaf curvature and determine the tree species. The algorithmic program runs a binary classifier to 1st section leaf and non-leafy half. If this step detects that input image don't have a single leaf on a light-weight and plain background, the user is asked to require the image once more with higher than conditions. The algorithmic program presently covers 184 tree species of north jap North American nation. Authors in [26] segments pictures of a plant within the field and performs classification. It additionally estimates the three- dimensional(3D) plant data for higher segmentation. The system needs a lot of then one image of the plant for 3D reconstruction. additionally the leaves to be divided got to be within the center of the image. Authors in [4] conferred a theme for observation plant leaves victimization infrared camera and golem. The system are often used for machine-driven plant searching that is useful for detection of disease or mineral deficiencies. The planned answer is predicated on many assumptions; one. Boundaries of the leaves square measure visible; a pair of. Leaves square measure giant for analysis; three. The leaves remains stationary during searching. 4. The searching needs controlled setting. In [18] a system for leaf extraction from the cover has been developed victimization fuzzy bunch and genetic algorithmic program (GA). The non occluded individual leaves square measure reassembled victimization GA. The 83.5% plants within the second week of testing made a minimum of one pc extracted leaf for identification by ancient options like form, texture or color. The authors in [27] propose a leaf image classification with the difficult background supported previous form data. Then Hu geometric and Zernike moments square measure wont to extract the form options. Twenty leaf categories square measure with success known with a mean accuracy of ninety two.6%. what is more within the ending of most cotton sickness, leaves dry out and becomes brown. Color of

dry leaves merges with color extracting color data in visual domain don't work o.k. [29]. Authors in [23] used active of soil that is additionally brown during this a part of Bharat. so standard ways of contour model to section cotton leaf captured from field so Hu-moments options square measure extracted for coaching reconciling neuro-fuzzy abstract thought system. The classification accuracy was found to be eighty fifth. The Principal part Analysis (PCA)/K-Nearest Neighbour (KNN) primarily based classifier was utilized in [9] to diagnose sickness on cotton leaves. inexperienced channel of RGB image was taken into thought for trustworthy feature assortment. In [10], ninetieth red spots in cotton crops was detected victimization Eigen fea- tures extraction technique. In [3], authors planned detection and classification of plant leaf and stem diseases victimization K- means that clump followed by a pre-trained neural network. The according preciseness was ninety three. In [22] authors planned sickness detection dispensed victimization edge detection in color domain. R, G, B color feature primarily based image segmentation is dispensed to search out target regions i.e. pathologic spots. Later, options like boundary, shape, color and texture square measure extracted to acknowledge diseases and roll out recommendation. In [6] 3 completely different approaches victimization color texture square measure went to classify pictures. the feel is computed from co- incidence matrix of the image. In [5], authors planned a system for recognizing the medicative plant leaves. The options embrace gray textures, gray tone spacial dependency matrices (GTSDM) and native binary pattern (LBP) opera- tors. Six completely different classifiers square measure went to classify the plant leaves. a unique boundary detection theme supported edge flow was planned in [17]. This theme utilizes a prognostic writing model to spot the direction of modification in color and texture at every image location at a given scale, and constructs a footing flow vector.

CHAPTER – 2

LITERATURE SURVEY

Ole Mathis Opstad Kruse et al. [1] evaluated the performance of four classification strategies in distinguishing pixels representing dislocated areas on the leaf surfaces. suited a pattern MIA approach combined with T2, RSS statistics and Linear discriminant analysis supervised classification strategies and K-means cluster unsupervised methodology was used. LDA method considerably outperformed 3 alternative classification methods in constituent identification with considerably higher accuracy, precision, true positive rate & F-score likewise as significantly lower false positive rate and computation time. A true positive rate of eightieth indicates that two hundredth of the dislocated pixels were incorrectly known as healthy.

S. Arivazhagan et al. [2] focuses on recognition of plant leaf using machine learning techniques by exploitation the form features, Vein options, Color features/Moments, Texture Features & Zernike Moments.

Yinmao Song et al. [6] delineated the feature extraction methods of crop illness supported laptop image process technology very well. The characteristics of crop diseases square measure complexity and variability thanks to big selection of crop diseases, in addition to severity of diseases. The crop illness ought to be analyzed and effectively known in line with the screening and representative characteristics parameter values supported color, texture and form.

Di Cui et al. in [18] represented methodology for quick & correct detection & classification of plant diseases. Authors used Otsu segmentation, K-means cluster & back propagation feed forward neural network for cluster and classification of diseases that have an effect on on plant leaves.

Alexander A. Doudkin in —Three Level Neural Network for data clusterization on pictures of infected crop fieldll planned neural network approach for segmentation of agricultural landed fields in remote sensing knowledge. A neural network algorithm supported back propagation is employed for segmentation of color pictures of crop field infected by diseases that amendment usual color of crops.[19].

Viraj Gulhane at el. within the within the Detection of diseases on cotton leaves and its doable doable provides the colour image segmentation technique to extract the color options of the

cotton leaves. this method provides straightforward thanks to extract the various options of morbid leaf of cotton image. After that the unattended network and back propagation neural network square measure applied to cluster the ensuing color pixels and extract cotton leaf color from diseases a part of image respectively.[12].

H.Al-Hiary, Bani-Ahmad,M-Reyalat and M.Braik within the paper titled —Fast and correct detection and classification of plant disease developed additional correct and quick detection technique of disease. K-Means bunch and neural network have been developed for bunch and classification of diseases that have an effect on on plant leaf. They found that planned approach, which can considerably support associate degree correct detection of leaf diseases in very little process effort [13].

An algorithmic program for illness spot segmentation exploitation Image processing technique in plant leaf is enforced by Piyush Chaudhari, Anand K. Chaudhari, Dr. A.N.Cheeran and Sharda Godara. Color rework of RGB image are often used for higher segmentation of illness spot. during this paper, they compared the effect of 3 completely different color area i.e. CIELAB, HIS & YbCr. Median filter is employed for image smoothing. They found that CIELAB color model is healthier than different 2 models.[14].

CHAPTER - 3

AIM AND SCOPE

3.1 AIM OF PROJECT

To Design a Responsive web Page For Cotton Disease Detection

3.2 OBJECTIVES

The aim of this project is to design and build a model by deep learning that helps, To identify the cotton diseases at early or the initial stage. For extraction of useful features using trained model by Deep Learning. To identify the cotton diseases at early or the initial stage using deep neural network.

In future, the model can be implemented on other plant diseases detection like wheat disease detection, Paddy disease detection, turmeric disease detection with large dataset using CNN.

3.3 SCOPE

web-based system has been successfully implemented for cotton disease detection for cotton leaves using Convolutional Neural Network. ... Alternaria Macrospora and Bacterial Blight are successfully being detected by system with training accuracy 90% using training dataset of 1951 images.

ADVANTAGES

- The main advantages of our solution include high processing speed and high classification accuracy. A Cotton Disease recognition system can work as a universal Detector, recognizing general abnormalities on the leaves.
- Automatic detection of the diseases by just seeing the symptoms on the plant leaves makes it easier as well as cheaper.

CHAPTER – 4

EXPERIMENTAL AND METHODS ; ALGORITHMS USED

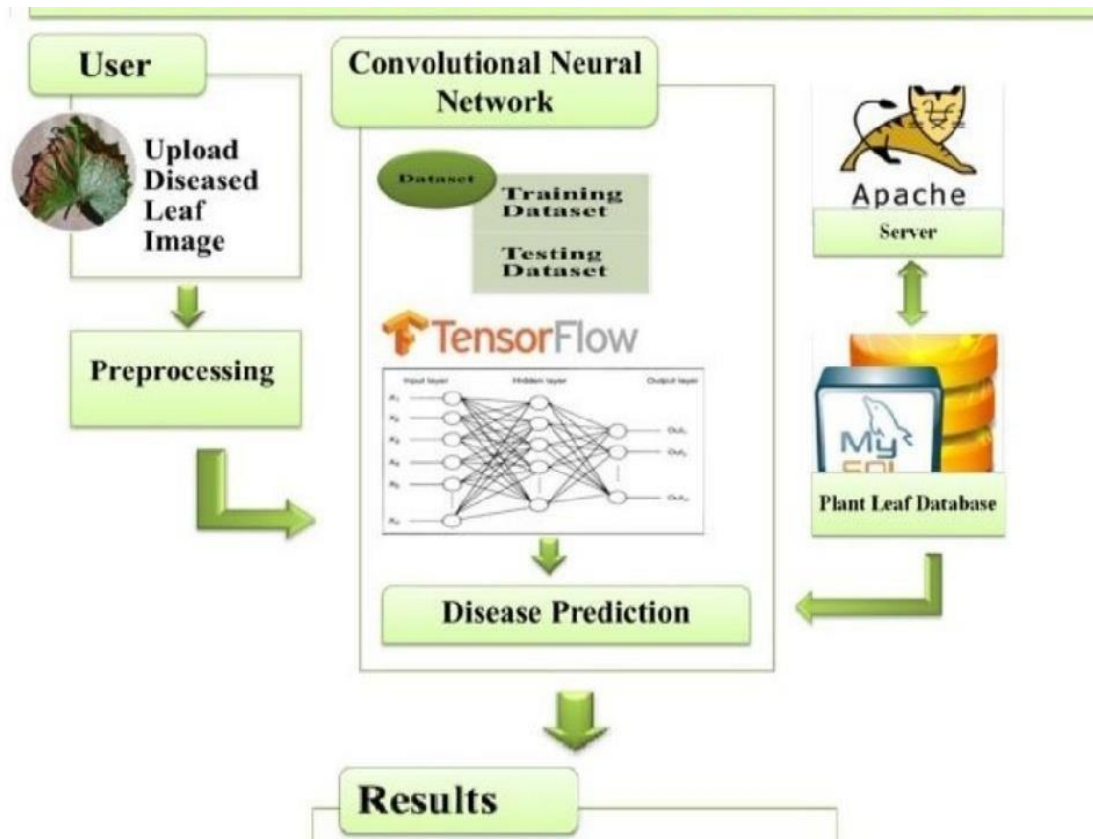


Fig 4.1 SYSTEM ARCHITECTURE

4.1.1 PRE-PROCESSING

It is common practice to have the pre-processing of Cotton leaf images before it has been extracted and classified. There are several steps used for the detection of plant leaf diseases as shown in fig.5. The processing scheme consists of image acquisition through digital camera or web. Image pre-processing includes image enhancement and image segmentation where the affected and useful areas are segmented; feature extraction and classification are carrying out. Finally the presence of diseases on the plant leaf will be identified. In the initial step, RGB images of leaf samples were picked up. The step-by-step procedure is shown as below.

- 1) RGB image acquisition
- 2) Preprocessing of image using histogram equalization

- 3) Resize the image
- 4) K-mean Algorithm for image segmentation
- 5) Computing features extraction
- 6) Classification & Recognition using neural networks
- 7) Statistical analysis

4.1.2 Preprocessing of Cotton Leaf Image

The input image has to be preprocessed because images are corrupted by a type of multiplicative noise like light intensity and shadow on a cotton leaf images that may contain useful information about the leaf spot that can be used in the diagnosis. The preprocessing is done with the contrast enhancement using Histogram equalization.



Fig 4.2 LEAFS OF DISEASED COTTON PLANT

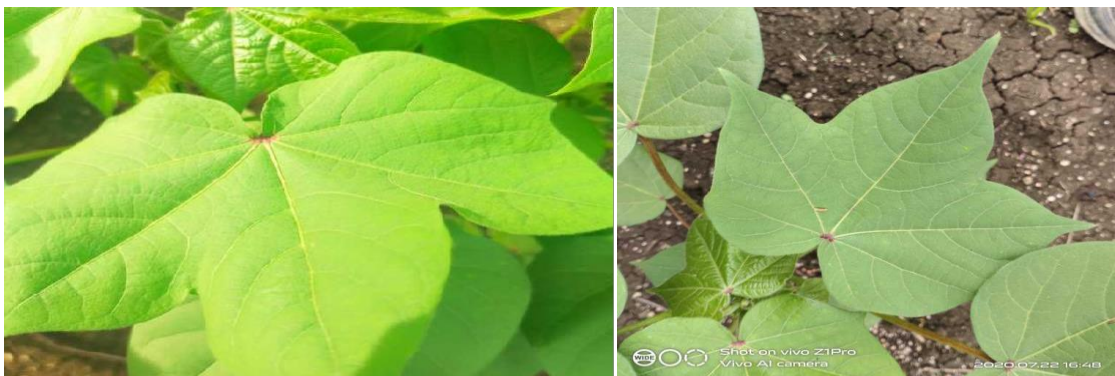


Fig 4.3 LEAFS OF FRESH COTTON PLANT

4.1.3 Image Segmentation

In the complex environment, the most crucial task is how to segment the images while localizing and detecting diseased plant leaves, since the major aim of image

segmentation is to set the symptom information apart from the background. There are many researchers making a deep investigation on it. In 2017, Ali et al. applied the Delta E color difference algorithm to separate the disease-infected area [13]. In general, four major methods are used to perform the image segmentation which are discussed the detail in the following paragraph [14].

Some researchers integrate the region of interest (ROI) and other methods to segment images. For example, Kao et al. claimed that the convolutional autoencoders served as the background filter to determine the ROI in an image [15]. The second method only concerns region segmentation. In 2013, Pujari et al. claimed that images were divided into various regions which had a special meaning and extracted the images' feature [16]. Akram and other colleagues provided an image processing model with real-time synchronous processing. By dividing the image into three color spaces, it can carry out contrast stretching, feature vector, and salient region recognition [17]. In addition, other researchers chose deep learning techniques to segment and detect images. Marko et al. recommended a depth-based target detection algorithm and used the two-stage algorithm to optimize plant disease images detection [18]. At last, the thresholding is common in segmentation. In 2018, Li et al. applied multilevel thresholding techniques based on gray histogram for image segmentation [19]. Mohamed and Diego presented a new multiobjective metaheuristic on the basis of a multiverse optimization algorithm to segment grayscale images via multilevel thresholding [20].

However, there is a fact that cannot be ignored. Because of the complexity of color information in the complicated environment, the machine vision algorithm based on color, ROI, and threshold performs poorly in practice.

4.1.4 Feature Extraction

The feature extraction of plant disease faces many problems in identifying plant disease. The distinct image features include textures, shape, color, and motion-related attributes, which are the essential conditions for disease feature extraction.

[21, 22]. Raza and his colleagues described a method that uses color and texture features to extract disease spots [23]. Hu et al. proposed the Dempster–Shafer (D-S) evidence theory and multifeature fusion for extracting features as well as the results were processed by introducing variance to improve decision rules of D-S evidence theory [24]. In addition, Turkoglu depicted improved versions of the Local Binary Patterns (LBP) methodology, which uses the original LBP local quadratic value to transform the image into grayscale and processes the R and G channels of the image by considering overall and region [25]. Li et al. researched an IoT feature extraction for the intelligent city based on the deep migration learning model [26]. There was an application in music, which can extract meaningful audio features in order to enable the visualizations to be responsive to the music [27]. And in recent studies, lots of novel approaches have been put forward to implementing feature extraction. For example, concerning the challenging task that the extraction of relevant and distinct features from electroencephalogram (EEG), Meziani et al. proposed two new spectral estimators that were robust against non-Gaussian, nonlinear, and nonstationary signals [28]. What is more, as Liu et al. reported, the high-dimensional time–frequency spectrum features were extracted by using the residual neural network and the improved signal-to-clutter ratio (SCR) [29]. Xu et al. introduced a feature extraction method based on the Hilbert marginal spectrum to perform the wear of milling tools [30].

4.1.5 Disease Identification

As for the precise identification, so many techniques are developed and researched to get accurate results. The identification model focused on using class labels for training images and built a fine-grained image classification system [31]. Zhang et al. reported a recognition method for plant disease leaf images based on a hybrid clustering [32]. In 2017, Patil et al. described a content-based image retrieval (CBIR) system to extract texture features and means value to compute color features, and support vector machine (SVM) classifier was used for classification [33]. Through above researches, the major goal was to design the classification schemes and image analysis for feature extraction and identification. Recently, other new approaches have been introduced to identify the disease more accurately and precisely. A novel system based on the selection of pictures and short text.

descriptions helped nonexperts in identifying plant diseases that can be used remotely from a desktop as well as in a smart phone or personal digital assistants [34]. Pertot et al. presented a scheme that used mobile phones for real-time on-field imaging of diseased plants and used mobile devices for leaf image segmentation and spotting of disease patch with improved *k*-means clustering [35]. Yang et al. presented a microscopy image detection methodology based on the synergistic judgment of texture and shape features and the decision tree-confusion matrix [36].

Additionally, the convolutional neural network is numerously utilized in identifying diseases. Chad et al. established a system capable of automatically identifying plant disease in field-acquired images of maize plants [37]. Ni et al. used the deepconvolution neural network to train 1632 images of corn kernels and designed an automatic corn detector [38]. Lu et al. proposed a rice diseases identification method based on deep convolutional neural networks (CNNs) techniques [39]. Zhang et al. designed an agricultural machinery image recognition network using the deep learning algorithm [40]. Zhang et al. improved deep convolution neural network to improve the accuracy of maize leaf disease identification [41]. Images were input into two deep learning-based architectures, namely, AlexNet and VGG-16 net, to perform detection [42]. Coulibaly et al. suggested a method using transfer learning for feature extraction to build an identification system [43]. However, due to the requirement for high hardware resources and traditional neural network models of high quality and quantity of data sets in the training process, the training wastes much time that is not conducive to the promotion and use of the model. In this paper, we recommend a transfer learning model for identification combined with the pretrained model, using the dataset of disease leaves to train the model.

From the above research findings, some achievements have been achieved in three aspects: leaf image segmentation, leaf lesion feature extraction, and leaf disease recognition. However, there are still many problems to be solved to realize plant disease identification in the complex environment.

It improves the perceptibility of objects in the prospect by enhancing the intensity difference between objects and their background. It is typically performed contrast stretch followed by tonal enhancement.

Descriptions helped nonexperts in identifying plant diseases that can be used remotely from a desktop as well as in a smart phone or personal digital assistants [34]. Pertot et al. presented a scheme that used mobile phones for real-time on-field imaging of diseased plants and used mobile devices for leaf image segmentation and spotting of disease patch with improved k -means clustering [35]. Yang et al. presented a microscopy image detection methodology based on the synergistic judgment of texture and shape features and the decision tree-confusion matrix [36].

Additionally, the convolutional neural network is numerously utilized in identifying diseases. Chad et al. established a system capable of automatically identifying plant disease in field-acquired images of maize plants [37]. Ni et al. used the deepconvolution neural network to train 1632 images of corn kernels and designed an automatic corn detector [38]. Lu et al. proposed a rice diseases identification method based on deep convolutional neural networks (CNNs) techniques [39]. Zhang et al. designed an agricultural machinery image recognition network using the deep learning algorithm [40]. Zhang et al. improved deep convolution neural network to improve the accuracy of maize leaf disease identification [41]. Images were input into two deep learning-based architectures, namely, AlexNet and VGG-16 net, to perform detection [42]. Coulibaly et al. suggested a method using transfer learning for feature extraction to build an identification system [43]. However, due to the requirement for high hardware resources and traditional neural network models of high quality and quantity of data sets in the training process, the training wastes much time that is not conducive to the promotion and use of the model. In this paper, we recommend a transfer learning model for identification combined with the pretrained model, using the dataset of disease leaves to train the model.

From the above research findings, some achievements have been achieved in three aspects: leaf image segmentation, leaf lesion feature extraction, and leaf disease recognition. However, there are still many problems to be solved to realize plant disease identification in the complex environment.

4.1.6 Contrast Enhancement

It improves the perceptibility of objects in the prospect by enhancing the intensity difference between objects and their background. It is typically performed contrast stretch followed by tonal enhancement. This procedure could both be performed in

single step. A contrast stretch improves the intensity differences consistently across the dynamic range of the image, whereas tonal enhancements improve the intensity differences in the highlight (bright), midtone (grays), or shadow (dark) regions at the expense of the brightness differences in the further regions.

The leaf spot in the capture image generally contains reflection from source, which forms some intense spot in the cotton leaf, but pixel value within the cotton leaf is over a particular threshold then it is replaced by pixel value of some neighbourhood pixel. This operation fills all intense leaf spot present in cotton leaf area. Ni et al. used the deep convolution neural network to train 1632 images of corn kernels and designed an automatic corn detector [38]. Lu et al. proposed a rice diseases identification method based on deep convolutional neural networks (CNNs) techniques [39]. Zhang et al. designed an agricultural machinery image recognition network using the deep learning algorithm [40]. Zhang et al. improved deep convolution neural network to improve the accuracy of maize leaf disease identification [41]. Images were input into two deep learning-based architectures, namely, AlexNet and VGG-16 net, to perform detection [42]. Coulibaly et al. suggested a method using transfer learning for feature extraction to build an identification system [43]. However, due to the requirement for high hardware resources and traditional neural network models of high quality and quantity of data sets in the training process, the training wastes much time that is not conducive to the promotion and use of the model. In this paper, we recommend a transfer learning model for identification combined with the pretrained model, using the dataset of disease leaves to train the model.

Instance-based classifiers, such as the k-mean classifier operate on the premises that arrangement of unknown instances can be done by concerning the unknown to the known considering to some distance/match function. The instinct is that 2 instances distant separately in the instance space defined by the appropriate distance function are less probable than 2 closely located instances to belong to the similar class.

The objective of the k-mean clustering algorithm is to use a database in which the data points are separated into several separate classes to predict the classification of a new sample point.

The non-parametric k-mean classifier is tested in this study. It classifies a test sample to a class according to the majority of the training neighbours in the feature space by using the minimum Euclidean distance criterion. The algorithm for the nearest neighbour rule

single step. A contrast stretch improves the intensity differences consistently across the dynamic range of the image, whereas tonal enhancements improve the intensity differences in the highlight (bright), midtone (grays), or shadow (dark) regions at the expense of the brightness differences in the further regions.

4.1.7 Image Segmentation

The leaf spot in the capture image generally contains reflection from source, which forms some intense spot in the cotton leaf, but pixel value within the cotton leaf is over a particular threshold then it is replaced by pixel value of some neighbourhood pixel. This operation fills all intense leaf spot present in cotton leaf area. Ni et al. used the deep convolution neural network to train 1632 images of corn kernels and designed an automatic corn detector [38]. Lu et al. proposed a rice diseases identification method based on deep convolutional neural networks (CNNs) techniques [39]. Zhang et al. designed an agricultural machinery image recognition network using the deep learning algorithm [40]. Zhang et al. improved deep convolution neural network to improve the accuracy of maize leaf disease identification [41]. Images were input into two deep learning-based architectures, namely, AlexNet and VGG-16 net, to perform detection [42]. Coulibaly et al. suggested a method using transfer learning for feature extraction to build an identification system [43]. However, due to the requirement for high hardware resources and traditional neural network models of high quality and quantity of data sets in the training process, the training wastes much time that is not conducive to the promotion and use of the model. In this paper, we recommend a transfer learning model for identification combined with the pretrained model, using the dataset of disease leaves to train the model.

4.1.8 Classification

Instance-based classifiers, such as the k-mean classifier operate on the premises that arrangement of unknown instances can be done by concerning the unknown to the known considering to some distance/match function. The instinct is that 2 instances distant separately in the instance space defined by the appropriate distance function are less probable than 2 closely located instances to belong to the similar class.

The objective of the k-mean clustering algorithm is to use a database in which the data points are separated into several separate classes to predict the classification of a new sample point.

The non-parametric k-mean classifier is tested in this study. It classifies a test sample to

a class according to the majority of the training neighbours in the feature space by using the minimum Euclidean distance criterion. The algorithm for the nearest neighbour rule is summarized as follows; given an unknown feature vector x and a distance measure, then Out of the N training vectors, identify the k nearest neighbours, regardless of class label.

Out of these k samples, identify the number of vectors, k_i , that belong to class $w_i, i=1, 2, \dots, M$. Assign x to the class w_i with the maximum number k_i of samples.

4.1.9 IMAGE ENHANCEMENT

The image enhancement of normalized image has been carried out due to reasons of low contrast, background illumination and non uniform brightness.

Hu et al. proposed the Dempster–Shafer (D-S) evidence theory and multifeature fusion for extracting features as well as the results were processed by introducing variance to improve decision rules of D-S evidence theory [24]. In addition, Turkoglu depicted improved versions of the Local Binary Patterns (LBP) methodology, which uses the original LBP local quadratic value to transform the image into grayscale and processes the R and G channels of the image by considering overall and region [25]. Li et al. researched an IoT feature extraction for the intelligent city based on the deep migration learning model [26]. There was an application in music, which can extract meaningful audio features in order to enable the visualizations to be responsive to the music [27]. And in recent studies, lots of novel approaches have been put forward to implementing feature extraction. For example, concerning the challenging task that the extraction of relevant and distinct features from electroencephalogram (EEG), Meziani et al. proposed two new spectral estimators that were robust against non-Gaussian, nonlinear, and nonstationary signals [28]. What is more, as Liu et al. reported, the high-dimensional time–frequency spectrum features were extracted by using the residual neural network and the improved signal-to-clutter ratio (SCR) [29]. Xu et al. introduced a feature extraction method based on the Hilbert marginal spectrum to perform the wear of milling tools [30]. Images were input into two deep learning-based architectures, namely, AlexNet and VGG-16 net, to perform detection [42]. Coulibaly et al. suggested a method using transfer learning for feature extraction to build an identification system [43]. However, due to the requirement for high hardware resources and traditional neural network models of high quality and quantity of data sets in the training process,

problem can be overcome by removal of background illumination in order to get a good distributed texture image.

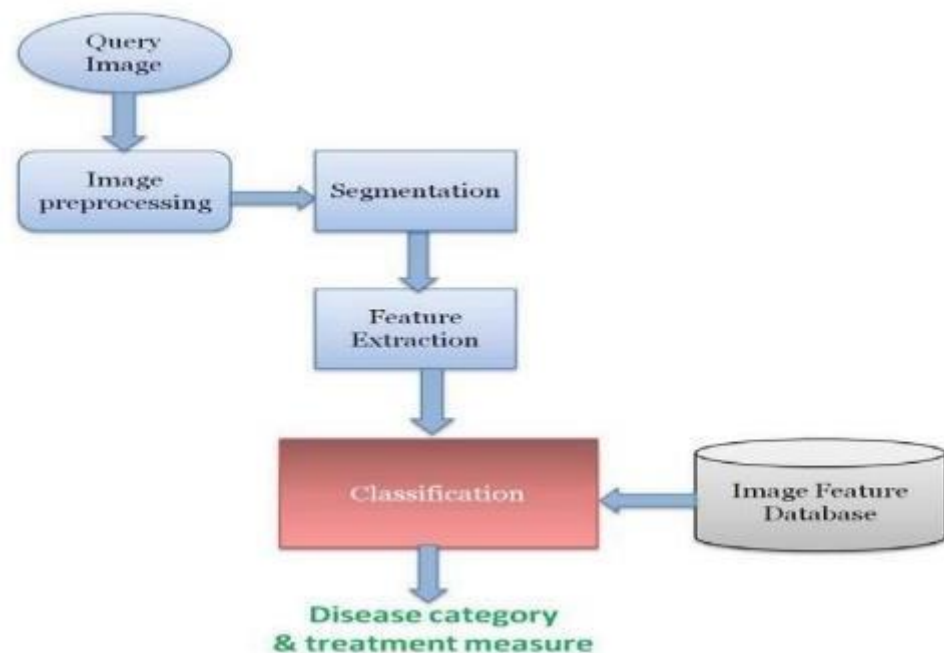


Fig – 4.4 BLOCK DIAGRAM

4.2.1 METHODOLOGY

- **Input the dataset** : Data is taken from Kaggle open source repository. It was a collection of cotton leafs which are random images irrespective of size and orientation. The input path is set to the location where the images resides.
- **Preparing the images for Train/Test**: Now images which are presented as lists are converted into arrays for categorization. Then splitting of images to train dataset and test dataset.
- **Classify cotton leafs using CNN**: Converting the labels into one hot encoding and building a CNN model takes place.
- **Maxpool & Dropout**: They are taken into consideration there by maxpooling

and dropping out the non required elements. Then compiling the model with train and test data takes place for maximizing the results.

- **Loading and Verification:** After compiling testing accuracy on test dataset is taken into consideration then the input images are given to the model to identify then if its present in the predefined class the results are more accurate if its not present it tries to test from the model and provides the output result.

4.2.2 Data Pre-processing

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. Mohanty et al. (2016) and Oppenheim and Shani (2017) transformed their images to grayscale. Mohanty et al. (2016) 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% (for details on the f1-score, see Powers, 2011). 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 the same study, the authors also evaluated the impact of background suppression. In fact, background management is one of the challenging elements in the implementation of automatic methods for identifying phytosanitary problems in imagery. With conventional image processing methods, leaf segmentation is a preliminary step to the analysis (Barbedo, 2016). The performance obtained by Mohanty et al. (2016) 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.

4.2.3 Feature Extraction

The feature extraction of plant disease faces many problems in identifying plant disease. The distinct image features include textures, shape, color, and motion- related attributes, which are the essential conditions for disease feature extraction [21, 22]. Raza and his colleagues described a method that uses color and texture features to extract disease spots [23]. Hu et al. proposed the Dempster–Shafer (D-S) evidence theory and multifeature fusion for extracting features as well as the results were processed by introducing variance to improve decision rules of D-S evidence theory [24]. In addition, Turkoglu depicted improved versions of the Local Binary Patterns (LBP) methodology, which uses the original LBP local quadratic value to transform the image into grayscale and processes the R and G channels of the image by considering overall and region [25]. Li et al. researched an IoT feature extraction for the intelligent city based on the deep migration learning model [26]. There was an application in music, which can extract meaningful audio features in order to enable the visualizations to be responsive to the music [27]. And in recent studies, lots of novel approaches have been put forward to implementing feature extraction. For example, concerning the challenging task that the extraction of relevant and distinct features from electroencephalogram (EEG), Meziani et al. proposed two new spectral estimators that were robust against non-Gaussian, nonlinear, and nonstationary signals [28]. What is more, as Liu et al. reported, the high-dimensional time–frequency spectrum features were extracted by using the residual neural network and the improved signal-to-clutter ratio (SCR) [29]. Xu et al. introduced a feature extraction method based on the Hilbert marginal spectrum to perform the wear of milling tools [30].

4.2.4 SEGMENTATION

From the higher than steps, the infected portion of the leaf is extracted. The infected region is then segmental into variety of patches of equal size. the scale of the patch is chosen in such some way that the numerous data isn't lost. during this approach patch size of 32*32 is taken. subsequent step is to extract the helpful segments. Not all segments contains vital quantity of that data. that the patches that square measure having quite one-half of the knowledge square measure taken under consideration for the more analysis.

The feature extraction of plant disease faces many problems in identifying plant disease. The distinct image features include textures, shape, color, and motion- related attributes, which are the essential conditions for disease feature extraction [21, 22]. Raza and his colleagues described a method that uses color and texture features to extract disease spots [23]. Hu et al. proposed the Dempster–Shafer (D-S) evidence theory and multifeature fusion for extracting features as well as the results were processed by introducing variance to improve decision rules of D-S evidence theory [24]. In addition, Turkoglu depicted improved versions of the Local Binary Patterns (LBP) methodology, which uses the original LBP local quadratic value to transform the image into grayscale and processes the R and G channels of the image by considering overall and region [25]. Li et al. researched an IoT feature extraction for the intelligent city based on the deep migration learning model [26]. There was an application in music, which can extract meaningful audio features in order to enable the visualizations to be responsive to the music [27]. And in recent studies, lots of novel approaches have been put forward to implementing feature extraction. For example, concerning the challenging task that the extraction of relevant and distinct features from electroencephalogram (EEG), Meziani et al. proposed two new spectral estimators that were robust against non-Gaussian, nonlinear, and nonstationary signals [28]. What is more, as Liu et al. reported, the high-dimensional time–frequency spectrum features were extracted by using the residual neural network and the improved signal-to-clutter ratio (SCR) [29]. Xu et al. introduced a feature extraction method based on the Hilbert marginal spectrum to perform the wear of milling tools [30].

From the higher than steps, the infected portion of the leaf is extracted. The infected region is then segmental into variety of patches of equal size. the scale of the patch is chosen in such some way that the numerous data isn't lost. during this approach patch size of 32*32 is taken. subsequent step is to extract the helpful segments. Not all segments contains vital quantity of that data. that the patches that square measure having quite one-half of the knowledge square measure taken under consideration for the more analysis.

4.3 Dataset Design

4.3.1 Training Phase

During the training phase, the model's internal weights are automatically updated over several iterations. External factors such as the training strategy, architecture, regularization techniques, or the value of the hyperparameters influence this training process.

Comparing studies and their results to extract insights on how to define the training phase is complicated because they do not use the same data and they do not provide all the parameters required to reproduce their experiments. It is also difficult to appreciate the significance of the conclusions made in these studies because their experiments are not performed multiple times to evaluate the impact of random initializations and training sample ordering. Nevertheless, we decided to present some comparisons of training and architectural strategies while keeping in mind that some of their results are potentially biased. 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 (Figure 2A). This simplifies the image analysis process by removing any variability related to external conditions or plant morphology in order to focus on symptom expression. A total of 13 of the 19 studies used such images. The second and slightly more complex type of dataset consists of images captured under uncontrolled conditions, but focusing on a particular plant organ, generally a leaf. In this case, images have a complex background but the largest area is occupied by the object of interest (Figure 2B). Only 3 of the 19 studies used such images. Finally, the last type of dataset consists of images captured under uncontrolled conditions and without focusing on a particular plant organ. These images therefore reflect what an operator would see in the field, It is also difficult to appreciate the significance of the conclusions made in these studies because their experiments are not performed multiple times to evaluate the impact of random initializations and training sample ordering. Nevertheless, we decided to present some comparisons of training and architectural strategies while keeping in mind that some of their results are potentially biased. 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 (Figure 2A). This simplifies the image analysis process by

removing any variability related to external conditions or plant morphology in order to focus on symptom expression.

4.3.2 Data Origin and Characteristics

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 (Figure 2). 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 (Figure 2A). This simplifies the image analysis process by removing any variability related to external conditions or plant morphology in order to focus on symptom expression. A total of 13 of the 19 studies used such images. The second and slightly more complex type of dataset consists of images captured under uncontrolled conditions, but focusing on a particular plant organ, generally a leaf. In this case, images have a complex background but the largest area is occupied by the object of interest (Figure 2B). Only 3 of the 19 studies used such images. Finally, the last type of dataset consists of images captured under uncontrolled conditions and without focusing on a particular plant organ. These images therefore reflect what an operator would see in the field, with all the complexity associated with foliage architecture (Figure 2C). It is also difficult to appreciate the significance of the conclusions made in these studies because their experiments are not performed multiple times to evaluate the impact of random initializations and training sample ordering. Nevertheless, we decided to present some comparisons of training and architectural strategies while keeping in mind that some of their results are potentially biased. 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 (Figure 2A). This simplifies the image analysis process by removing any variability related to external conditions or plant morphology in order to focus on symptom expression. A total of 13 of the 19 studies used such images. The second and slightly more complex type of dataset consists of images captured under uncontrolled conditions, but focusing on a particular plant organ, generally a leaf. In this case, images have a complex background but the largest area is occupied by the object of interest (Figure 2B). Only 3 of the 19 studies used such images. Finally, the last type of dataset consists of images captured under uncontrolled conditions and without focusing on a particular plant organ.

kind of dataset is the one best-suited to build an operational automatic phytosanitary monitoring tool. Only 3 of the 19 studies used such images.

A majority of the images used in these studies come from public datasets (11 out of 19 studies). The most widely used dataset is PlantVillage, a database initially described in Hughes and Salathé (2015) and now containing 87,848 photographs of leaves of healthy and infected plants (Ferentinos, 2018). A total of 25 species are represented through 58 classes, with 62.7% of the images taken under controlled conditions. Barbedo (2018b) used an open database containing 1,951 images of 12 plant species through 56 classes. Its acquisition conditions are mainly controlled. On the other hand, DeChant et al. (2017) used a dataset captured under uncontrolled conditions and with no focus on a particular plant organ. This dataset is highly specialized in the identification of Northern Leaf Blight (NLB) infected maize plants. A description of an extended version of this dataset can be found in Wiesner-Hanks et al. (2018). It consists of 18,222 annotated handheld, boom and drone images with 105,705 NLB lesions. The availability of such databases is significant, as it provides a high number of annotated images—a key factor of success in deep learning. These images can also be used for benchmarking, enabling a comparison of the accuracy of models created by different research groups. Collecting images in the field and gathering them into databases such as PlantVillage is an ideal solution to improve the community's research capabilities.

The images in private datasets (the type utilized by eight studies) were all handheld. A weakness of such datasets is that they sometimes lack important details, such as the capture conditions, acquisition date, varieties studied and the intensity of the analyzed symptoms. In some studies, too few or even no samples are illustrated, making it impossible to determine the application prospects of the trained models. Indeed, a model trained with a low-complexity image dataset (as defined earlier) will not be able to generalize to data from a more complex setting. This was underlined by Mohanty et al. (2016), whose accuracy went from 99.35% on a held-out test set to 31% on a test set with images taken under different conditions. The work of Ferentinos (2018) divided the PlantVillage dataset based on either laboratory or field conditions. Their model trained on laboratory images reached 33.27% accuracy when applied on field images, while their model trained on field images reached 65.69% accuracy when applied on laboratory images of the dataset.

As data is the key element of kind of dataset is the one best-suited to build an operational automatic phytosanitary monitoring tool. Only 3 of the 19 studies used such images.

A majority of the images used in these studies come from public datasets (11 out of 19 studies). The most widely used dataset is PlantVillage, a database initially described in Hughes and Salathé (2015) and now containing 87,848 photographs of leaves of healthy and infected plants (Ferentinos, 2018). A total of 25 species are represented through 58 classes, with 62.7% of the images taken under controlled conditions. Barbedo (2018b) used an open database containing 1,951 images of 12 plant species through 56 classes. Its acquisition conditions are mainly controlled. On the other hand, DeChant et al. (2017) used a dataset captured under uncontrolled conditions and with no focus on a particular plant organ. This dataset is highly specialized in the identification of Northern Leaf Blight (NLB) infected maize plants. A description of an extended version of this dataset can be found in Wiesner-Hanks et al. (2018). It consists of 18,222 annotated handheld, boom and drone images with 105,705 NLB lesions. The availability of such databases is significant, as it provides a high number of annotated images—a key factor of success in deep learning. These images can also be used for benchmarking, enabling a comparison of the accuracy of models created by different research groups. Collecting images in the field and gathering them into databases such as PlantVillage is an ideal solution to improve the community's research capabilities.

The images in private datasets (the type utilized by eight studies) were all handheld. A weakness of such datasets is that they sometimes lack important details, such as the capture conditions, acquisition date, varieties studied and the intensity of the analyzed symptoms. In some studies, too few or even no samples are illustrated, making it impossible to determine the application prospects of the trained models. Indeed, a model trained with a low-complexity image dataset (as defined earlier) will not be able to generalize to data from a more complex setting. This was underlined by Mohanty et al. (2016), whose accuracy went from 99.35% on a held-out test set to 31% on a test set with images taken under different conditions. The work of Ferentinos (2018) divided the PlantVillage dataset based on either laboratory or field conditions. Their model trained on laboratory images reached 33.27% accuracy when applied on field images, while their model trained on field images reached 65.69% accuracy when applied on laboratory images.

a successful CNN-based model, its characteristics and source(s) must be well described.

4.3.3 Class Taxonomy Definition

Classes are defined by diseases and species (in the case of multi-species models). In Wang et al. (2017), 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 (as did Wang et al., 2017) or integrated in global classes. However, since it is difficult to obtain enough images for all expressions, especially over a single growing season, Barbedo (2018a) recommends the continuous addition of new images to the training dataset. The number of classes vary greatly from one study to another—from 2 (DeChant et al., 2017) to 58 (Ferentinos, 2018). Four studies also dedicated a class to the background. Adding such a class is meaningful for a real world application, where the background is not uniform. To this end, Sladojevic et al. (2016) and Brahimi et al. (2018) used the publicly available Stanford Background dataset (Gould et al., 2009). On the other hand, Fuentes et al. (2017) extracted patches of healthy plants and background from their images and put them together in a transversal class of negatives. They relied on hard negatives to form this. A majority of the images used in these studies come from public datasets (11 out of 19 studies). The most widely used dataset is PlantVillage, a database initially described in Hughes and Salathé (2015) and now containing 87,848 photographs of leaves of healthy and infected plants (Ferentinos, 2018). A total of 25 species are represented through 58 classes, with 62.7% of the images taken under controlled conditions. Barbedo (2018b) used an open database containing 1,383 images of 12 plant species through 56 classes. Its acquisition conditions are mainly controlled. On the other hand, DeChant et al. (2017) used a dataset captured under uncontrolled conditions and with no focus on a particular plant organ. This dataset is highly specialized in the identification of Northern Leaf Blight (NLB) infected maize plants. A description of an extended version of this dataset can be found in Wiesner-Hanks et al. (2018). It consists of 18,222 annotated handheld, boom and drone images with 105,705 NLB lesions. The availability of such databases is significant, as it provides a high number of annotated images—a key factor of success in deep learning. These images can also be used for benchmarking, enabling a comparison of the accuracy of model by different research groups.

Collecting images in the field and gathering them into databases such as PlantVillage is an ideal solution to improve the community's research capabilities.

The images in private datasets (the type utilized by eight studies) were all handheld. A weakness of such datasets is that they sometimes lack important details, such as the capture conditions, acquisition date, varieties studied and the intensity of the analyzed symptoms. In some studies, too few or even no samples are illustrated, making it impossible to determine the application prospects of the trained models. Indeed, a model trained with a low-complexity image dataset (as defined earlier) will not be able to generalize to data from a more complex setting. This was underlined by Mohanty et al. (2016), whose accuracy went from 99.35% on a held-out test set to 31% on a test set with images taken under different conditions. The work of Ferentinos (2018) divided the PlantVillage dataset based on either laboratory or field conditions. Their model trained on laboratory images reached 33.27% accuracy when applied on field images, while their model trained on field images reached 65.69% accuracy when applied on laboratory images. As data is the key element of a successful CNN-based model, its characteristics and source(s) must be well described.

4.3.4 Class Taxonomy Definition

Classes are defined by diseases and species (in the case of multi-species models). In Wang et al. (2017), 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 (as did Wang et al., 2017) or integrated in global classes. However, since it is difficult to obtain enough images for all expressions, especially over a single growing season, Barbedo (2018a) recommends the continuous addition of new images to the training dataset. The number of classes vary greatly from one study to another—from 2 (DeChant et al., 2017) to 58 (Ferentinos, 2018). Four studies also dedicated a class to the background. Adding such a class is meaningful for a real world application, where the background is not uniform. To this end, Sladojevic et al. (2016) and Brahimi et al. (2018) used the publicly available Stanford Background dataset (Gould et al., 2009). On the other hand, Fuentes et al. (2017) extracted patches of healthy plants and infection can lead to a high degree of variability in symptoms.

Background from their images and put them together in a transversal class of negatives. They relied on hard negatives to form this class. Hard negatives are false positives obtained from previous evaluations that are integrated into the training set for the negative class for a new training. This practice aims to reduce the number of false positives by confronting the network to situations that it has previously failed to manage properly. The negative class is quite complex to construct, as it must integrate all the diversity of the real world without having an excessively higher number of images than the positive classes. Using hard negatives can help target the most relevant negative cases. Even if imbalance is a reflection of the field reality where healthy plants are in the majority, some classes should not have a much higher number of examples than others. The class imbalance problem affects both the convergence of the model and its ability to generalize (Buda et al., 2018). Different strategies can be chosen either at a data level or at an algorithm level to minimize the detrimental effect of imbalance (Krawczyk, 2016). Wang et al. (2017) chose undersampling. They divided their healthy leaves' class into 12 clusters of 110 images for training and 27 for testing, 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 (Barbedo, 2018a). It would be unfeasible to collect enough images for each class. To solve this problem, Barbedo (2018a) proposes to consider lesions individually and process only areas of symptomatic interest identified by the user. Another possibility is to train binary models (target disease versus the rest) and thus give primacy of detection to the disease of interest, even if it does not express “pure” symptoms. In any case, the greatest challenge is to cover enough symptom expressions so that the model can be applied in real world conditions.

4.3.5 Data Annotation

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. Indeed, what is the best piece of information to send to the network? The lesion, the leaf, or the whole plant? Each of these levels are valid and provide complementary features. With a close symptom view, textural elements stand out (Figure 3A). A complete view of the leaf reveals symptom patterns (Figure 3B). Finally, a view of the whole plant provides a spatial perspective of the symptoms. For example, some problems occur preferentially on young leaves.

4.3.6 Dataset Division

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 (see Goodfellow et al., 2016). 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 (Chollet, 2017). 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 for training (between 70 and 85% depending of the size of the dataset). Mohanty et al. (2016) and Zhang S. et al. (2018) 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 dataset is divided into several subsets that are randomly used for training or validation. Cross validation is useful when the dataset is small as it avoids any result bias caused by an arbitrary predetermined and fixed data separation. It is used in 5 of the 19 selected studies. Once the hyperparameters have been defined, further training can be done by gathering the training and validation sets to benefit from a greater number of images.

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, 6 of the studies we analyzed (31.5%) formed their test set this way. Worse yet, 11 of the studies (58%) did not even have a test set. Only 2 studies (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. Therefore, there must be a way to ensure that the generated models can operate under different conditions and in different fields. Using data acquired on a different plot for the test set can be a good and simple way to achieve data independence. Even though obtaining data in an agricultural setting can be complex, it should not lead to overlooking this critical aspect.

4.4 COLOR TRANSFORMATION STRUCTURE

Firstly, the RGB pictures of leaves square measure reborn into Hue Saturation Intensity (HSI) color area representation. the aim of the colour area is to facilitate the specification of colours in some normal, typically accepted means. HSI (hue, saturation, intensity) color model could be (hue, saturation, intensity) color model could be a well-liked color model as a result of it's supported human perception.

4.5 Hue COLOR ATTRIBUTE

It refers to the dominant color as viewed by someone. Saturation refers to the relative Purity or the quantity of white light-weight additional to hue and intensity refers to the amplitude of the sunshine.

Color areas are often reborn from one area to a different simply. once the transformation method, the H element is taken under consideration for more analysis. S and that i square measure born since it doesn't provide further data.

kind of dataset is the one best-suited to build an operational automatic phytosanitary monitoring tool. Only 3 of the 19 studies used such images.

A majority of the images used in these studies come from public datasets (11 out of 19 studies). The most widely used dataset is PlantVillage, a database initially described in Hughes and Salathé (2015) and now containing 87,848 photographs of leaves of healthy and infected plants (Ferentinos, 2018). A total of 25 species are represented through 58 classes, with 62.7% of the images taken under controlled conditions. Barbedo (2018b) used an open database containing 1,951 images of 12 plant species through 56 classes. Its acquisition conditions are mainly controlled. On the other hand, DeChant et al. (2017) used a dataset captured under uncontrolled conditions and with no focus on a particular plant organ. This dataset is highly specialized in the identification of Northern Leaf Blight (NLB) infected maize plants. A description of an extended version of this dataset can be found in Wiesner-Hanks et al. (2018). It consists of 18,222 annotated handheld, boom and drone images with 105,705 NLB lesions. The availability of such databases is significant, as it provides a high number of annotated images—a key factor of success in deep learning. These images can also be used for benchmarking, enabling a comparison of the accuracy of models created by different research groups. Collecting images in the field and gathering them into databases such as PlantVillage is an ideal solution to improve the community's research capabilities.

The images in private datasets (the type utilized by eight studies) were all handheld. A weakness of such datasets is that they sometimes lack important details, such as the capture conditions, acquisition date, varieties studied and the intensity of the analyzed symptoms. In some studies, too few or even no samples are illustrated, making it impossible to determine the application prospects of the trained models. Indeed, a model trained with a low-complexity image dataset (as defined earlier) will not be able to generalize to data from a more complex setting. This was underlined by Mohanty et al. (2016), whose accuracy went from 99.35% on a held-out test set to 31% on a test set with

images taken under different conditions. The work of Ferentinos (2018) divided the PlantVillage dataset based on either laboratory or field conditions. Their model trained on laboratory images reached 33.27% accuracy when applied on field images, while their model trained on field images reached 65.69% accuracy when applied on laboratory images.

4.6 MASKING INEXPERIENCED PIXELS

Here, we tend to determine in the main the inexperienced coloured pixels. After this, supported specified threshold worth computed for these pixels, the principally inexperienced components square measure cloaked as if the inexperienced element of the pixel intensity is a smaller amount than the pre-computed threshold worth, the red, inexperienced and blue parts of the this component is appointed to a zero worth. 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 (see Goodfellow et al., 2016). 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 (Chollet, 2017). 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, 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). Mohanty et al. (2016) and Zhang S. et al. (2018) 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 dataset is divided into several subsets that are randomly used for training or validation. Cross validation is useful when the dataset is small as it avoids any result bias caused by an arbitrary predetermined and fixed data separation. It is used in 5 of the 19 selected studies. Once the hyperparameters have been defined, further training can be done by gathering the training and validation sets to benefit from a greater number of images.

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, 6 of the studies we analyzed (31.5%) formed their test set this way. Worse yet, 11 of the studies (58%) did not even have a test set. Only 2 studies (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. Therefore, there must be a way to ensure that the generated models can operate under different conditions and in different fields. Using data acquired on a different plot for the test set can be a good and simple way to achieve data independence. Even though obtaining data in an agricultural setting can be complex, it should not lead to overlooking this critical aspect.

4.7 COLOR Co-OCCURENCE TECHNIQUE

The color co-occurrence texture analysis technique is developed through the abstraction Gray-level Dependence Matrices (SGDM). grey|the grey} level co-occurrence methodology could be a applied mathematics thanks to describe form by statistically sampling the means bound gray-levels occur in regard to alternative gray levels. These matrices live the likelihood that a component at one explicit grey level can occur at a definite distance and orientation from any component providing component encompasses a second explicit grey level.

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, 6 of the studies we analyzed (31.5%) formed their test set this way. Worse yet, 11 of the studies (58%) did not even have a test set. Only 2 studies (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. Therefore, there must be a way to ensure that the generated models can operate under different conditions and in different fields. Using data acquired on a different plot for the test set can be a good and simple way to achieve data independence. Even though obtaining data in an agricultural setting can be complex, it should not lead to overlooking this critical aspect.

This practice aims to reduce the number of false positives by confronting the network to situations that it has previously failed to manage properly. The negative class is quite complex to construct, as it must integrate all the diversity of the real world without having an excessively higher number of images than the positive classes. Using hard negatives can help target the most relevant negative cases. Even if imbalance is a reflection of the field reality where healthy plants are in the majority, some classes should not have a much higher number of examples than others. The class imbalance problem affects both the convergence of the model and its ability to generalize (Buda et al., 2018). Different strategies can be chosen either at a data level or at an algorithm level to minimize the detrimental effect of imbalance (Krawczyk, 2016). Wang et al. (2017) chose undersampling. They divided their healthy leaves' class into 12 clusters of 110 images for training and 27 for testing, 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 (Barbedo, 2018a). It would be unfeasible to collect enough images for each class. To solve this problem, Barbedo (2018a) proposes to consider lesions individually and process only areas of symptomatic interest identified by the user. Another possibility is to train binary models (target disease versus the rest) and thus give primacy of detection to the disease of interest, even if it does not express "pure" symptoms. In any case, the greatest challenge is to cover enough symptom expressions so that the model can be applied in real world conditions.

CHAPTER – 5

RESULTS AND DISCUSSION

The goal of this application is to develop a system that recognizes cotton leaf diseases. user need to transfer Image in Nursing image then, Image process starts with the digitized color image of the unhealthy leaf. Finally by applying the CNN plant disease will be foreseen.

The dataset contains 2 styles of unhealthy leaf pictures that ar Alternaria Macrospora and microorganism Blight and additionally Healthy Leaf pictures. The training dataset consists of total 1800 pictures while the testing dataset consists of 151 pictures. The accuracy of training is ninetieth whereas the accuracy of testing is ninety two.

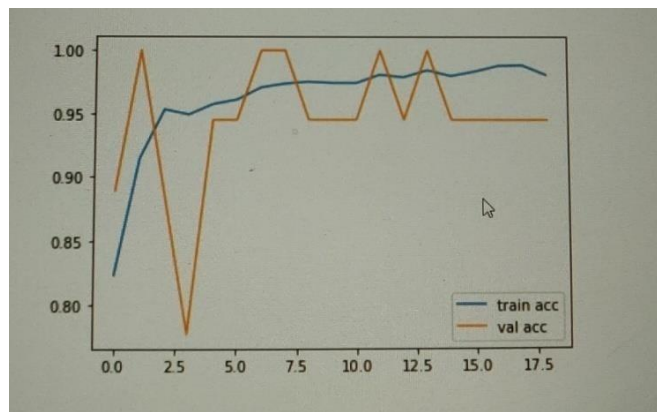


Figure 5.1 Graph of Training Accuracy vs Test Accuracy

In the above graph (Fig 5.1) we can see that when the accuracy of the training model increases, the accuracy of the test model also incr easing.

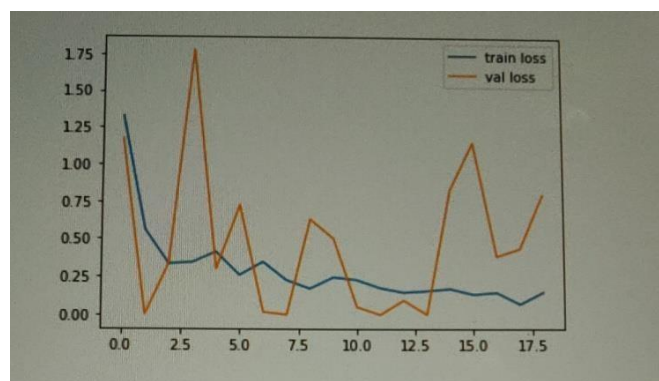


Figure 5.2 Graph of Training Loss vs Test Loss

In the above graph (Fig 5.2) we can see that when the loss of the training model decreasing, the loss of the test model also decreasing.

5.1 SCREENSHOTS

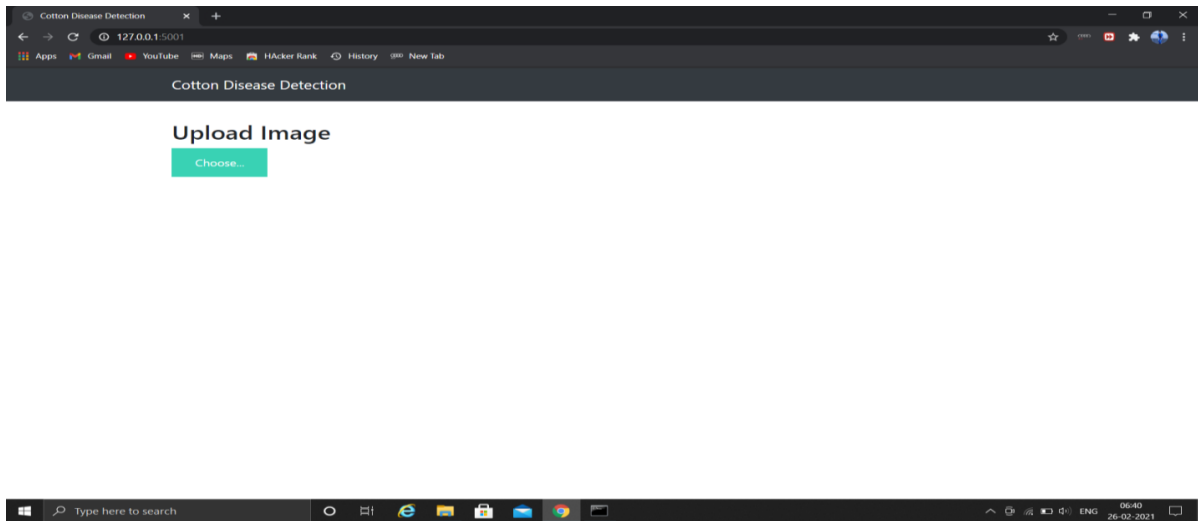


FIGURE - 5.3 SCREENSHOT OF OUTPUT PAGE -1

The Fig 5.3 is the home page of the project. It is the frontend of the project or it termed it can also termed as user interface. Where user can upload the leaf of cotton plant image to detect whether the particular cotton is diseased cotton leaf or fresh cotton leaf.

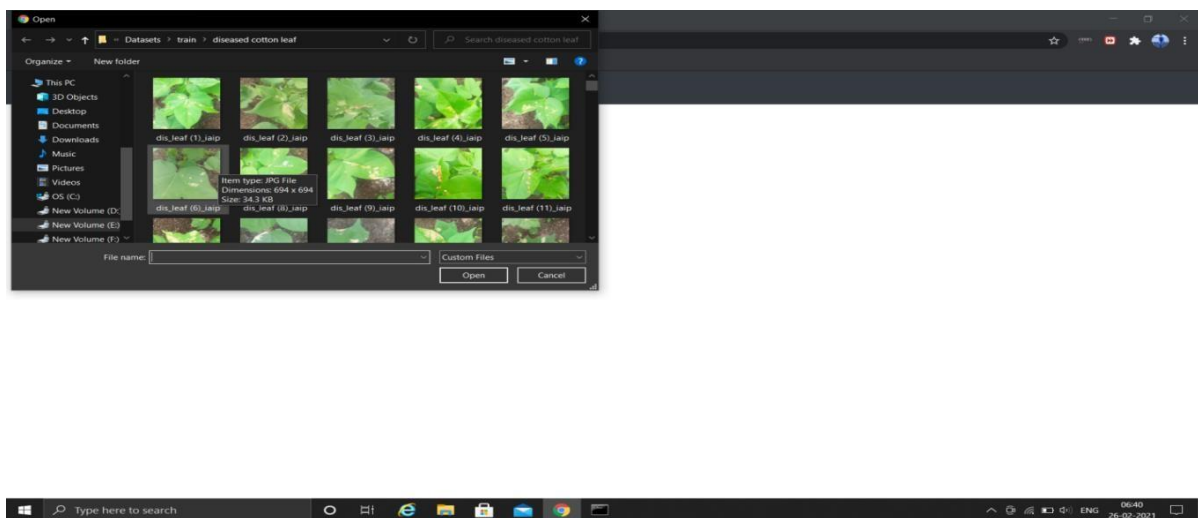


FIGURE – 5.4 SCREENSHOT OF OUTPUT PAGE -2

The Fig 5.4 shows that when the user want to upload the image, When the user click the upload image button then it takes to the local drive where the images are available and he can select the particular image from the local drive and upload.

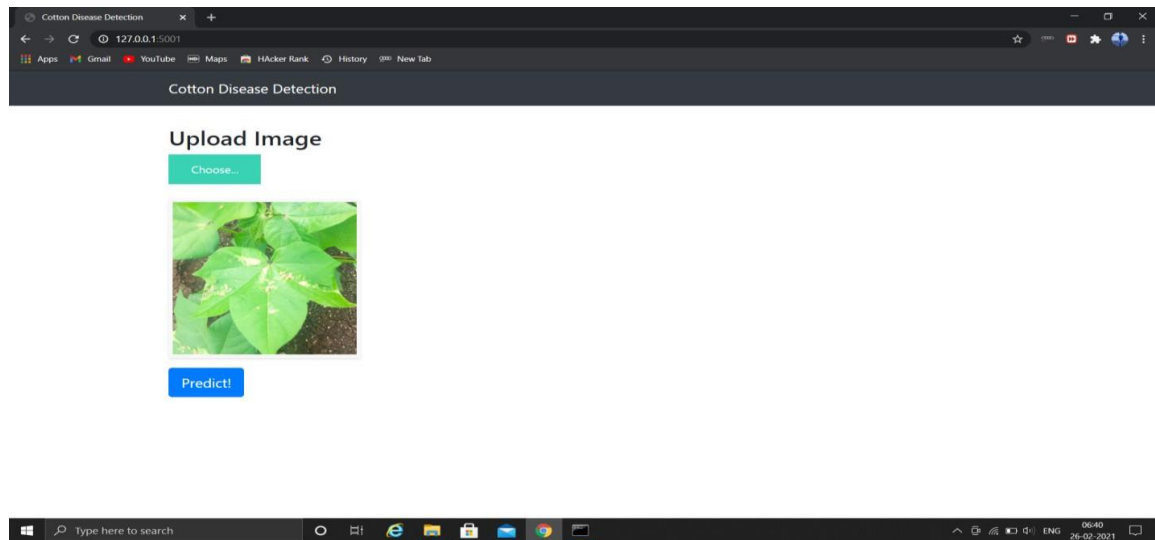


FIGURE – 5.5 SCREENSHOT OF OUTPUT PAGE -3

The Fig 5.5 shows that whatever the user upload the image of the leaf of cotton plant from the local drive, it will upload it here. After uploading the leaf of the cotton plant, if the user click the predict button it will detect whether the particular cotton leaf cotton plant is diseased cotton plant or fresh cotton plant.

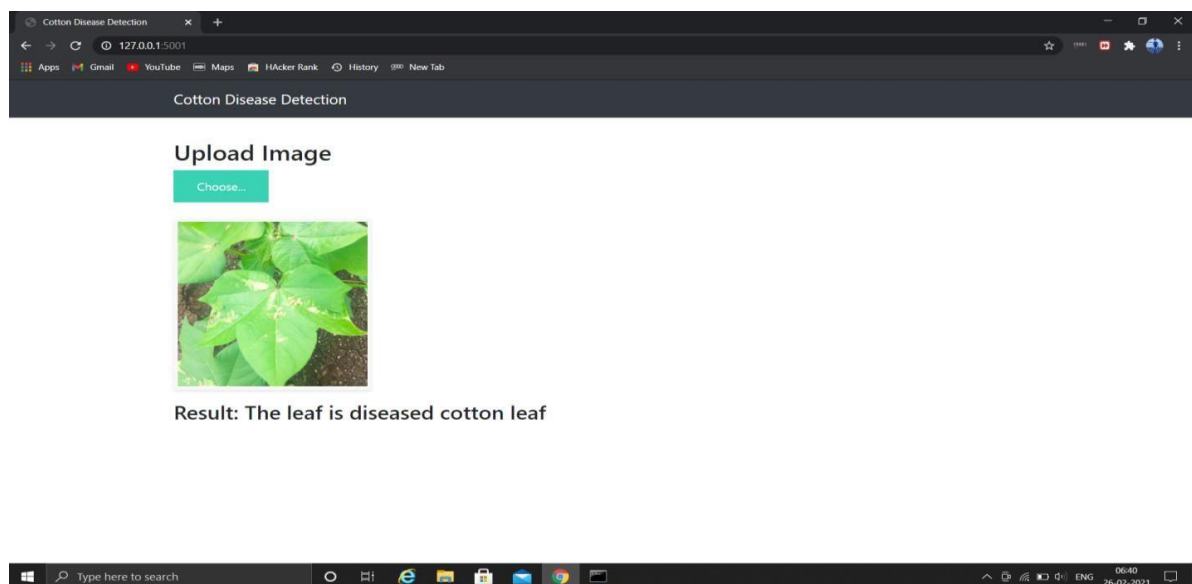


FIGURE- 5.6 SCREENSHOT OF OUTPUT PAGE -4

The Fig 5.6 shows the output, that the selected cotton plant is diseased cottonplant.

CHAPTER – 6

CONCLUSION AND FUTURE WORK

CONCLUSION

A new approach of using deep learning method was explored in order to automatically classify and detect cotton diseases from leaf images. The developed model was able to detect leaf presence and distinguish between healthy leaves and unhealthy leaves. The complete procedure was described, respectively, from collecting the images used for training and validation to image preprocessing and augmentation and finally the procedure of training the deep CNN and fine-tuning. Different tests were performed in order to check the performance of newly created model.

The proposed cotton disease detection system serves the farming community to improve their crop productivity by correctly classifying the disease type being occurred. The system is developed to detect cotton disease. This project is meant to detect diseases on a cotton leaf. Cotton plant disease does not only reduce their products but also deteriorate of their variety and its withdrawal from cultivation. The use of pesticides and fungicides in excess for the treatment of such diseases increases the danger of toxic residue level on agricultural products and has been identified as a major contributor to groundwater contamination. Again, farmers incur much loss due to the cost of these pesticides as applied on plants. Therefore, there is a need to take greater caution and minimize their use in order to protect water bodies and human life. In the future, we can development of real-time implementation of this algorithm in the farm for continuous monitoring and detection of plant diseases. In real time system, we can monitor and give an exact solution to avoid various diseases on the cotton plant.

FUTURE WORK

In future, the model can be implemented on other plant diseases detection like wheat disease detection, Paddy disease detection, turmeric disease detection with large dataset using CNN.

References

- [1] Sasaki Y, Okamoto T. Automatic diagnosis of plant disease 2 recognition between healthy and diseased leaf. *Journal of the Society of Agricultural Machinery*. 61(2): 119-126.
- [2] Serna JrDG, Pinto FAC, Queiroz DM, Viana PA. Fall Army worm damaged maize plant identification using digital images. *Biosystems Engineering*. 2003; 85: 449-454.
- [3] Chen Jiajuan, Ji Shouwen, Li Juan, Zhao Xuedu. Automatic Measurement of Danger Degree of Cotton Insect Pests Using Computer Vision. *Transactions of the Chinese Society of Agricultural Engineering*. 2001; 02: 157-160.
- [4] Jun Sun, Yan Wang, Xiaohong Wu, Xiaodong Zhang, Hongyan Gao. A New Image Segmentation Algorithm and Its Application in Lettuce Object Segmentation. *TELKOMNIKA*. 2012; 10(3): 557-563.
- [5] Hanshan Li, Zhiyong Lei. Research on Infrared Special Facula View Measurement Method Based on Image Processing Technology. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(6): 1422-1429.
- [6] Gonzalez RC, Wood RE. *Digital Image Processing*. Third Edition. BeiJing: Pubishing House of Electronics Industy. 2010.
- [7] Doutre C, Nasiopoulos P, Konstantinos N Platanioti. A Fast Demosaicking Method Directly Producing YCbCr 4:2:0 Outpu. *IEEE Transactions on Consumer Electronics*. 2007; 53(2).
- [8] Sanjay Kr, Singh DS, Chauhan, Mayank Vatsa, Richa Singh. A Robust Skin Color Based Face Detection Algorithm. *Tamkang Journal of Science and Engineering*. 2003; 6(4): 227-234.
- [9] NabiyeV V, Günay A. Problems of Cybernetics and Informatics. In V. NabiyeV (Ed.). *Towards a biometric purpose image filter according to skin detection*. 2008; 10-12.
- [10] Sun Hui-xian, Zhang Yu-hua, Luo Fei-lu. Color edge detection based on HSI color space. *Optical Technique*. 2009; 35(2).

- [11] Kekre HB, Sudeep D, Thepade, Parkar A. Performance Analysis of Kekre's Median Fast Search. In H. B. Kekre (Ed.). Kekre's Centroid Fast Search and Exhaustive Search Used for Coloring a Grayscale Image. 2010.
- [12] Toker CC, Chakraborty K. Quantitative Assessment of Lesion Characteristics and Disease Severity Using Digital Image Processing. Journal of Phytopathology, 2008,vol,145,no.7,pp.273-278 .
- [13] Al-Bashish, D., M. Braik, and S. Bani-Ahmad. 2011. Detection and classification of leaf diseases using K-means-based segmentation and neural networks based classification. Information Technology Journal, 10(2): 267-275.
- [14] P.Revathi, M.Hemalatha, "Classification of Cotton Leaf Spot Diseases Using Image Processing Edge Detection Techniques" ISBN, pp 169-173, 2012 IEEE.
- [15] Al-Hiary, S. Bani-Ahmad, M. Reyalat, M. Braik and Z. ALRahamneh, "Fast and Accurate Detection and Classification of Plant Diseases" IJCA, vol. 17(1), pp. 31-38, March 2011, IEEE2010.
- [16] Jayamala K. Patil, Raj Kumar, "Advances In Image Processing For Detection of Plant Diseases" JABAR, vol. 2(2), pp. 135-141, June2011. [6] Shen Weizheng, Wu Yachun, Chen zhanliang, Wei Hongda, "Grading M

APPENDIX

REFERENCES

- Dataset From kaggle.com
- **BOOKS:** Deep Learning, Python, Flask, Html, CSS & Flask

FRONT END CODE - HTML ,CSS & BOOTSTRAP

```
{% extends "base.html" %} {% block content %}
```

```
<h2>Upload Image</h2>
```

```
<div>
```

```
<form id="upload-file" method="post" enctype="multipart/form-data">
```

```
<label for="imageUpload" class="upload-label">
```

```
Choose...
```

```
</label>
```

```
<input type="file" name="file" id="imageUpload" accept=".png, .jpg, .jpeg">
```

```
</form>
```

```
<div class="image-section" style="display:none;">
```

```
<div class="img-preview">
```

```
<div id="imagePreview">
```

```
</div>
```

```
</div>
```

```
<div>
```

```
<button type="button" class="btn btn-primary btn-lg " id="btn-predict">Predict!</button>
```

```
</div>
```

```
</div>
```

```
<div class="loader" style="display:none;"></div>
```

```
<h3 id="result">
```

```
<span> </span>
```

```
</h3>
```

```
</div>
```

```
{% endblock %}
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Cotton Disease Detection</title>
<link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
<script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
<script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
<link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">

</head>

<body>
  <nav class="navbar navbar-dark bg-dark">
    <div class="container">
      <a class="navbar-brand" href="#">Cotton Disease Detection</a>

    </div>
  </nav>
  <div class="container">
    <div id="content" style="margin-top:2em">{% block content %}{% endblock %}</div>
  </div>
</body>

<footer>
  <script src="{{ url_for('static', filename='js/main.js') }}" type="text/javascript"></script>
</footer>

</html>

```

BACK END CODE : FLASK

```
# -*- coding: utf-8 -*-
"""

Created on Thu Jun 11 22:34:20 2020

@author: Krish Naik
"""

from __future__ import division, print_function
# coding=utf-8
import sys
import os
import glob
import re
import numpy as np
import tensorflow as tf
import tensorflow as tf

from tensorflow.compat.v1 import ConfigProto
from tensorflow.compat.v1 import InteractiveSession

config = ConfigProto()
config.gpu_options.per_process_gpu_memory_fraction = 0.2
config.gpu_options.allow_growth = True
session = InteractiveSession(config=config)
# Keras
from tensorflow.keras.applications.resnet50 import preprocess_input
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

# Flask utils
from flask import Flask, redirect, url_for, request, render_template
from werkzeug.utils import secure_filename
# from event.pywsgi import WSGIServer

# Define a flask app
app = Flask(__name__)

# Model saved with Keras model.save()
MODEL_PATH = 'model_resnet152V2.h5'

# Load your trained model
model = load_model(MODEL_PATH)
```

```

def model_predict(img_path, model):
    print(img_path)
    img = image.load_img(img_path, target_size=(224, 224))

    # Preprocessing the image
    x = image.img_to_array(img)
    # x = np.true_divide(x, 255)
    ## Scaling
    x=x/255
    x = np.expand_dims(x, axis=0)

    # Be careful how your trained model deals with the input
    # otherwise, it won't make correct prediction!
    # x = preprocess_input(x)

    preds = model.predict(x)
    preds=np.argmax(preds, axis=1)
    if preds==0:
        preds="The leaf is diseased cotton leaf"
    elif preds==1:
        preds="The leaf is diseased cotton plant"
    elif preds==2:
        preds="The leaf is fresh cotton leaf"
    else:
        preds="The leaf is fresh cotton plant"

    return preds

@app.route('/', methods=['GET'])
def index():
    # Main page
    return render_template('index.html')

@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':
        # Get the file from post request
        f = request.files['file']

        # Save the file to ./uploads
        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))

```

```
f.save(file_path)

# Make prediction
preds = model_predict(file_path, model)
result=preds
return result
return None

if __name__ == '__main__':
    app.run(port=5001, debug=True)
```

ALGORITHM DEVELOPMENT : DEEP LEARNING

```
from tensorflow.compat.v1 import ConfigProto
from tensorflow.compat.v1 import InteractiveSession

config = ConfigProto()
config.gpu_options.per_process_gpu_memory_fraction = 0.2
config.gpu_options.allow_growth = True
session = InteractiveSession(config=config)

# import the libraries as shown below

from tensorflow.keras.layers import Input, Lambda, Dense, Flatten
from tensorflow.keras.models import Model
#from tensorflow.keras.applications.resnet152V2 import ResNet152V2
#from keras.applications.vgg16 import VGG16
from tensorflow.keras.applications.inception_v3 import preprocess_input
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator,load_img
from tensorflow.keras.models import Sequential
import numpy as np
from glob import glob
#import matplotlib.pyplot as plt

# re-size all the images to this
IMAGE_SIZE = [224, 224]

train_path = 'Datasets/train'
valid_path = 'Datasets/test'

# Import the Vgg 16 library as shown below and add preprocessing layer to the front of VGG
# Here we will be using imagenet weights
import tensorflow
resnet152V2 =tensorflow.keras.applications.ResNet152V2(input_shape=IMAGE_SIZE +
[3], weights='imagenet', include_top=False)

# don't train existing weights
for layer in resnet152V2.layers:
    layer.trainable = False

# useful for getting number of output classes
folders = glob('Datasets/train/*')

# our layers - you can add more if you want
x = Flatten()(resnet152V2.output)

prediction = Dense(len(folders), activation='softmax')(x)

# create a model object
model = Model(inputs=resnet152V2.input, outputs=prediction)
```

```

# view the structure of the model
model.summary()

# tell the model what cost and optimization method to use
model.compile(
    loss='categorical_crossentropy',
    optimizer='adam',
    metrics=['accuracy']
)

# Using the Image Data Generator to import the images from the dataset
from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)

# Make sure you provide the same target size as initialised for the image size
training_set = train_datagen.flow_from_directory('Datasets/train',
                                                target_size = (224, 224),
                                                batch_size = 32,
                                                class_mode = 'categorical')

test_set = test_datagen.flow_from_directory('Datasets/test',
                                           target_size = (224, 224),
                                           batch_size = 32,
                                           class_mode = 'categorical')

# fit the model
# Run the cell. It will take some time to execute
r = model.fit_generator(
    training_set,
    validation_data=test_set,
    epochs=19
)

import matplotlib.pyplot as plt

# plot the loss
plt.plot(r.history['loss'], label='train loss')
plt.plot(r.history['val_loss'], label='val loss')
plt.legend()
plt.show()
plt.savefig('LossVal_loss')

```

```

# plot the accuracy
plt.plot(r.history['accuracy'], label='train acc')
plt.plot(r.history['val_accuracy'], label='val acc')
plt.legend()
plt.show()
plt.savefig('AccVal_acc')

# save it as a h5 file

from tensorflow.keras.models import load_model

model.save('model_resnet152V2.h5')

y_pred = model.predict(test_set)

y_pred

from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

model=load_model('model_resnet152V2.h5')

img_data

img=image.load_img('Datasets/Test',target_size=(224,224))
x=image.img_to_array(img)
x

x.shape

x=x/255
import numpy as np
x=np.expand_dims(x,axis=0)
img_data=preprocess_input(x)
img_data.shape

model.predict(img_data)
a=np.argmax(model.predict(img_data), axis=1)
a==1
import tensorflow as tf
tf.__version__

```

APPENDIX

A. Paper Acceptance Mail

B. Plagiarism Report

vikhilkumar984@gmail.com.docx

ORIGINALITY REPORT

12%	11%	11%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Qinghai He, Benxue Ma, Duanyang Qu, Qiang Zhang, Xinmin Hou, Jing Zhao. "Cotton Pests and Diseases Detection Based on Image Processing", TELKOMNIKA Indonesian Journal of Electrical Engineering, 2013 Publication	10%
2	Submitted to Indian Institute of Technology, Madras Student Paper	1%
3	en.wikipedia.org Internet Source	<1%
4	Submitted to University of Nottingham Student Paper	<1%

Exclude quotes Off
Exclude bibliography Off

Exclude matches Off

C. Journal Paper

DEEP NEURAL NETWORK BASED RECOGNITION OF COTTON DISEASE DETECTION BY LEAF IMAGE CLASSIFICATION

ABSTRACT

Illness of cotton plant infections is a method of diagnosing sicknesses by breaking down its visual attributes. Since highlights have been eliminated, I see how to eliminate visual highlights from photographs. The element evacuation strategy can be finished by giving depictions of different highlights (like SIFT, SURF or other profoundly qualified applicants). At that point the square of the square of the element depiction is moved to the component investigation separator. The classifier is a calculation program used to isolate highlights on a premise, for example, a preparation informational collection. A bunch of preparing information is a choice that is first given to a notable article (for this situation, the leaf has a particular infection). The leaves of the ailing square are characterized under a similar premise as the coach's information for the infection test addressed by the portrayal of the element. In this article, we will likely eliminate the indicative hardware from the imaging and to feature the picture preparing innovation. Illness grouping can be founded on the presence of microorganisms on cotton leaves, and any characterization can be utilized to distinguish casualties. The execution of the arranged model will be finished on the MATLAB machine, and the consequences of the arranged model will be accessible in the exactness, accuracy, memory, time, and numerous comparable boundaries.

LITERATURE SURVEY

Ole Mathis Opstad Kruse et al. [1] assessed the viability of the four-pronged separation procedure by isolating the pixels addressing the sporadic territories on the outside of the leaf. Reasonable for a blend of the MIA technique example and T2-controlled characterization procedure, RSS insights and direct separation investigation and the K-implies grouping solo strategy. The LDA

technique is obviously superior to the three characterization strategies for ID, and has a high exactness, accuracy, genuine positive rating and F rating likewise significantly diminishes bogus estimation and time figuring. The genuine positive proportion is one out of eighty, demonstrating that one of every a fifth of circuitous pixels are erroneously called solid pixels.

S. Arivazhagan et al. [2] centers around the utilization of AI procedures to recognize plant leaves utilizing structure factors, vascular alternatives, shading highlights/times, surface highlights and Zernike times.

Ingoma Yinmao et al. [6] A decent portrayal of PC picture preparing innovation upheld by a strategy for separating a plant microorganism. A factor in plant illnesses is the seriousness and seriousness of the infection, because of the huge number of plant sicknesses and the seriousness of the infection. Harvest illnesses ought to be dissected and better comprehended as far as tests and delegate boundary esteems (supporting shading, surface and structure).

UDi Cui et al. In [18], it addresses a method of diagnosing and ordering plant illnesses rapidly and precisely. The creator utilizes the Otsu division, K-implies the joining and appropriation behind the neural organization to gather and detach illnesses influencing the leaves of plants.

The neural organization approach created by Alexander A. Doudkin in the "Three level neural organization for information grouping of pictures of contaminated territories" is utilized to order farming fields with far off detecting data. Utilizing neural organization innovation that supports back spread, shading phaistos of contaminated estates are isolated, accordingly modifying the typical shade of plants [19].

AIM OF PROJECT

To Design a Responsive web Page For Cotton Disease Detection

OBJECTIVES

The reason for this undertaking is to plan and construct a model with inside and out learning helps that can distinguish cotton illnesses toward the start or the primary stage. Later on, the model can be utilized for plant infection location, for example, wheat illness disco hay, rice field sickness identification, and turmeric analysis utilizing CNN for a lot of information.

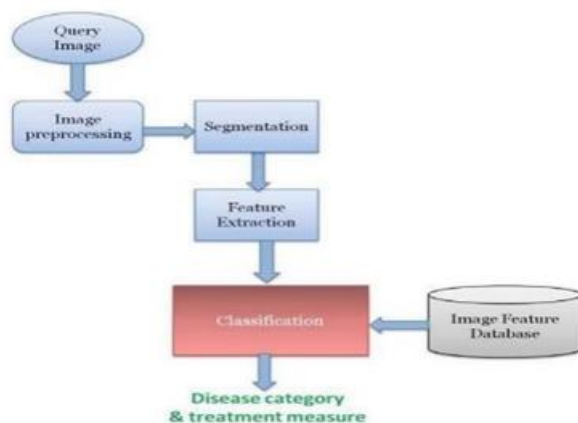
SCOPE

The utilization of convolutional neural organizations has effectively utilized a Web-based framework to recognize cotton leaf spot. ...The framework utilized the 1951 photography preparing information base to effectively recognize Alternaria and bacterial sicknesses with a preparation exactness of 90%.

ADVANTAGES

- The vital hints of our answer incorporate high goal speed and high partition precision. The cotton sickness checking framework can go about as an overall decorator to identify normal anomalies in leaves.
- The illness can be naturally recognized simply by finding the leaves on the leaves, so it is simpler and less expensive.

BLOCK DIAGRAM



METHODOLOGY

- Input the dataset: Data is taken from Kaggle open source repository. It was a collection of cotton leaves which are random images irrespective of size and orientation. The input path is set to the location where the images reside.
- Preparing the images for Train/Test: Now images which are presented as lists are converted into arrays for categorization. Then splitting of images to train dataset and test dataset.
- Classify cotton leaves using CNN: Converting the labels into one hot encoding and building a CNN model takes place.
- Maxpool & Dropout: They are taken into consideration there by maxpooling 15 15 and dropping out the non-required elements. Then compiling the model with train and test data takes place for maximizing the results.
- Loading and Verification: After compiling testing accuracy on test dataset is taken into consideration then the input images are given to the model to identify then if its present in the predefined class the results are more accurate if its not present it tries to test from the model and provides the output result.

Data Pre-processing

Prior to presenting a picture on an organization, two approaches are required. To begin with, the picture size ought to normally be acclimated to coordinate with the size of the CNN cover page. The size starting with one organization then onto the next is exceptionally normal. For instance, AlexNet goal is 227 x 227, DenseNet goal, ResNet and VGG is 224 x 224, and Inception goal is 299 x 299. , the picture ought to be utilized to assist the model with incorporating and better coordinate elusive information (IChollet, 2017). Other preliminary exercises might be recommended. Mohanty et al. (2016) and

Oppenheim and Shani (2017) convert their pictures into dim pictures. Mohanty et al. (2016) analyzed the exactness found in dark pictures with the precision found in shading photos. In the shading model, execution is somewhat higher, with the fl Score improved from 1.34% to 3.33% (for subtleties on the fl scale, see Power, 2011). Albeit the utilization of shading photos encourages the acknowledgment interaction, as execution diminishes marginally during the dark scale change measure, this features the way that the organization is profoundly reliant on different variables to recognize contaminations. In a similar report, the creators reconsidered the effect of back pressure. Truth be told, foundation the executives is one of the difficulties in the utilization of robotized strategies to picture preparing. leaf detachment is the distinguish phytosanitary issues in pictures. Utilizing conventional techniques for initial step for investigacion (Barbedo, 2016). The discoveries of Mohanty et al. (2016) Better foundation, improved fl focuses marginally.

Feature Extraction

The issue of plant sicknesses is confronting numerous difficulties in recognizing plant illnesses. Recognized picture highlights incorporate surface, shape, shading, and development related manifestations, which are essential conditions to kill infection qualities [21, 22]. Raza and partners depicted how to eliminate wounds utilizing shading and surface highlights [23]. Hu et al. Dempster-Shafer (D-S) hypothesis of proof and multi-layered coordination is proposed to prohibit factors, and the outcomes are dissected by acquainting fluctuation with improve the standards of D-S proof based hypothesis choices [24]. Likewise, Turkoglu depicts an improved form of the Local Binary Patten (LBP) strategy. which utilizes a subsequent neighborhood estimation of LBP to change the picture over to dim, and afterward examines the R and G channels of the picture by dissecting the entire space and space [25]. Li et al. As per the top to bottom learning model of transmission [26]. The Internet of Things highlight in savvy urban areas. There is an application in music that can deliver perceptible sound highlights so the visual can react to music [27]. What's more, in late exploration, various novel

methodologies have been proposed to decide include importance. For instance, regarding the difficult assignment of separating pertinent and particular highlights from the electroencephalogram (EEG), Mezirani et al. Two range entertainers are selected, who are solid in non-Gaussian, non-direct and non-straight images [28]. In particular, as Liu et al. As per reports, high-recurrence Fmge highlights are removed utilizing the excess neural organization and an improved sign to-mess (SCR) scale [29]. Xu et al. An element extraction technique dependent on the Hilbert edge range was acquainted with make the wear of the granulating device [30].

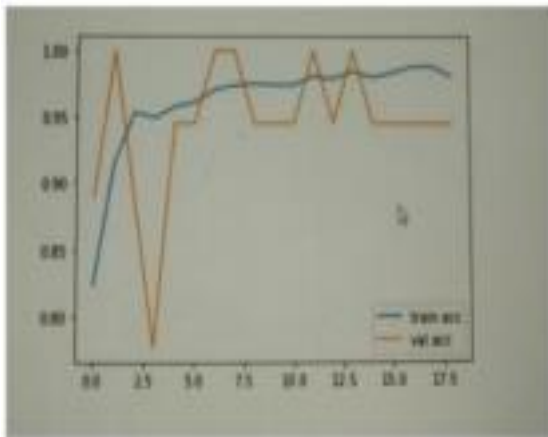
SEGMENTATION

From the higher than steps, the infected portion of the leaf is extracted. The infected region is then segmental into variety of patches of equal size. the scale of the patch is chosen in such some way that the numerous data isn't lost. during this approach patch size of 32 32 is taken. subsequent step is to extract the helpful segments. Not all segments contains vital quantity of that data. that the patches that square measure having quite one-half of the knowledge square measure taken under consideration for the more analysis.

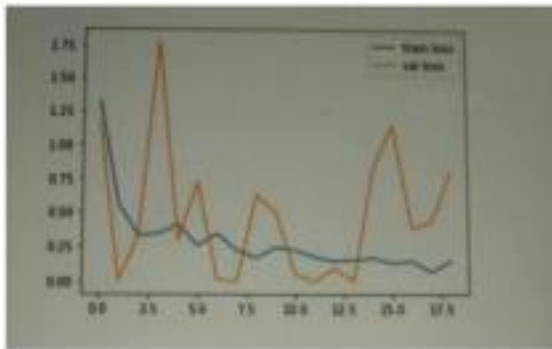
RESULTS AND DISCUSSION

The goal of this application is to develop a system that recognizes cotton leaf diseases. user need to transfer Image in Nursing image then, Image process starts with the digitized color image of the unhealthy leaf. Finally by applying the CNN plant disease will be foreseen.

The dataset contains 2 styles of unhealthy leaf pictures that are Alternaria Macrospora and microorganism Blight and additionally Healthy Leaf pictures. The training dataset consists of total 1800 pictures while the testing dataset consists of 151 pictures. The accuracy of training is ninetyth whereas the accuracy of testing is ninety-two.

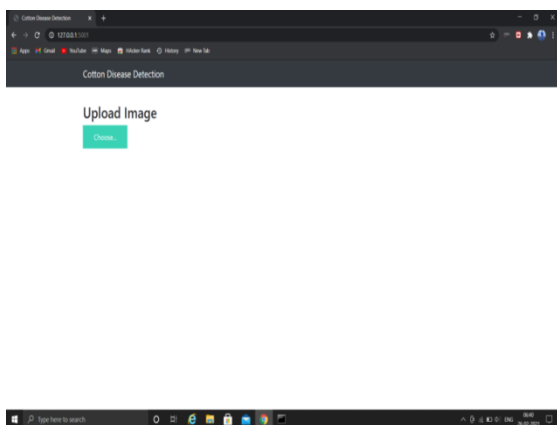


In the above graph we can see that when the accuracy of the training model increases, the accuracy of the test model also increasing.



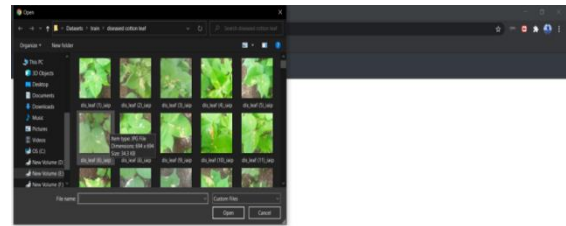
In the above graph we can see that when the loss of the training model decreasing, the loss of the test model also decreasing.

OUTPUT SCREENSHOTS

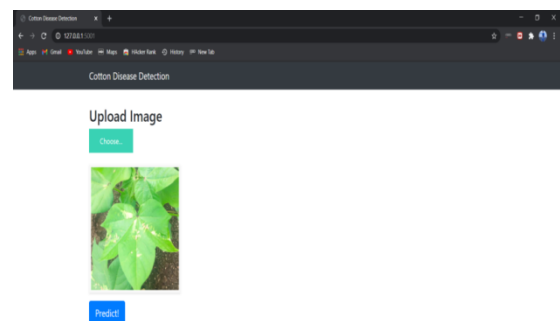


The above Figure is the home page of the project. It is the frontend of the project or it termed it can also termed as user interface.

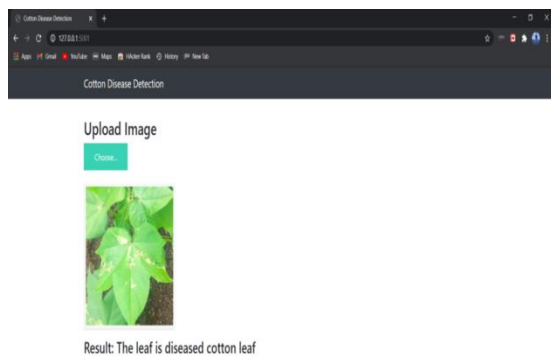
Where user can upload the leaf of cotton plant image to detect whether the particular cotton is diseased cotton leaf or fresh cotton leaf.



The above Figure shows that when the user want to upload the image, When the user click the upload image button then it takes to the local drive where the images are available and he can select the particular image from the local drive and upload.



The above Figure shows that whatever the user upload the image of the leaf of cotton plant from the local drive, it will upload it here. After uploading the leaf of the cotton plant, if the user click the predict button it will detect whether the particular cotton leaf/ cotton plant is diseased cotton plant or fresh cotton plant.



The above Figure shows the output, that the selected cotton plant is diseased cotton plant.

CONCLUSION

Investigate another technique for utilizing profound figuring out how to consequently distinguish cotton sicknesses in leaf pictures. This high level model can identify leaves and recognize solid leaves from terrible leaves-CNN profundity and great fix capacity. Different tests were directed to assess the presentation of the recently evolved model. The proposed cotton infectious prevention plan helps horticultural networks increment crop yields by precisely distinguishing the kinds of sicknesses that are happening. The undertaking expects to distinguish infections in cotton leaves, Cotton plant sicknesses diminish their profitability, yet in addition deteriorate their variety and withdrawal from cultivating. Extreme utilization of pesticides and organisms in the treatment of such infections can build the danger of harmful material in agrarian items, and has been recognized as the principle factor causing groundwater contamination. Additionally, ranchers have endured enormous misfortunes because of the expense of these pesticides utilized in crops. Along these lines, it is important to be more mindful and lessen its utilization to ensure water bodies and human wellbeing. Later on, we can improve the constant execution of this calculation on the homestead to persistently screen and identify plant illnesses. In the ongoing framework, we can screen and give explicit answers for forestall different infections in cotton plants.

FUTURE WORK

In future, the model can be implemented on other plant diseases detection like wheat disease detection, Paddy disease detection, turmeric disease detection with large dataset using CNN.

REFERENCES

- [1] Sasaki Y, Okamoto T. The Automatic diagnosis of plant disease to recognize between healthy and unhealthy leaves. *Journal of the Society of Agricultural Machinery*. 61(2): 119-126.
- [2] Serna JrDG, Pinto FAC, Queiroz DM, Viana PA. Fall Army worm damaged maize plant identification using digital images. *Biosystems Engineering*. 2003; 85: 449- 454.
- [3] Chen Jiajuan, Ji Shouwen, Li Juan, Zhao Xuedu. The Automatic Measurements of the Danger Degree of Cotton Insects Pesticides Using Computer Vision(CV). *Transactions of the Chinese Society of Agricultural Engineering*. 2001; 02: 157-160.
- [4] Jun Sun, Yan Wang, Xiaohong Wu, Xiaodong Zhang, Hongyan Gao. A New Image Segmentation Algorithm and Its Application in Lettuce Object Segmentation. *TELKOMNIKA*. 2012, 10(3): 557-563.
- [5] Hanshan Li, Zhiyong Lei. Research on Infrared Special Facula View Measurement Method Based on Image Processing Technic. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(6): 1422-1429.
- [6] Gonzalez RC, Wood RE. *Digital Image Processing*. Third Edition. BeiJing: Publishing House of Electronics Industry. 2010.
- [7] Doutre C, Nasiopoulos P, Konstantinos N Platanioti. A Fast Demosaicking Method Directly Producing YCbCr 4:2:0 Output *IEEE Transactions Electronics*. 2007; 53(2). on Consumer
- [8] Sanjay Kr, Singh DS. Chauhan, Mayank Vatsa, Richa Singh. The Robast Colour Based Face Detection Algorithm. *Tamkang Journal of Science and Engineering*. 2003; 6(4): 227-234.

[9] NabiyeV V. Günay A. Problems of Cybemetics and Informatics. In V. NabiyeV (Ed.). Towards a biometric purpose image filter according to skin detection. 2008; 10-12.

[10] Sun Hui-xian, Zhang Yu-hua, Luo Fci Color edge detection based on HSI color lu. space. Optical Technique. 2009: 35(2).