

Plant Disease Information Generating System Based On Android Application Technology

Jha Ashish K.¹, Bababe Adam B.², Dr. Ashok K. Sahoo³

¹Department of Computer Science and Engineering, Sharda University, Greater Noida, India

²Department of Computer Science and Engineering, Sharda University, Greater Noida, India

³Department of Computer Science and Engineering, Sharda University, Greater Noida, India

Abstract: Agriculture is the bedrock of any society and adequate information with regards to it is very vital at any given time. "Android Based Agricultural Info Generation Application" is an android application designed for supported mobiles that majorly focuses on Agriculture sector. It is a client-server architecture in which android mobile acts as a client, Server processes the client's query and relay the related information to the client. Application is installed in mobile device and information updates are sent to it through the server. This work basically identifies plants leaf diseases on the basis of image captured i.e. send via mobile, the server processes the image based on the diseases detected and gives advices if necessary. Different disease information and its remedial/preventive measures are provided in the local server. Users can browse and download information about these agricultural information at any time.

Keywords: Plant, Mobile, Android, Disease, Agriculture

1. Introduction

Mobile phones today have moved beyond their fundamental role of communications and have graduated to become an extension of the persona of the user. The capabilities of mobile networks are increasing rapidly, extending the scope of applications that can be supported to include high-resolution images, video and file sharing, which are utilized for effective purposes.

Android is an open source software platform created by Google and the open handset. It is primarily used to power mobile phones and has the capability to make inroads in many other embedded application markets. It comes with a much wider range of features, including Sensor Rich Application (All Purpose Tool) location based services using GPS, Social Network features, more powerful tools to support access to the Internet as well as video and audio recording and sharing. Google's Goggles is one of such powerful application of android that uses pictures to prompt a search, not just text or vocal input. When a photo is taken through Goggles, the free app scans the image, matching your picture against in the database and returning a result. To this end, things like plants still give Goggles trouble and results are not always consistent, this develops the interest to have a research that will find solution for dynamic things like plants whose nature and appearance change every now and then.

2. Literature Survey

A. E Governance

Several dimension and related factors influence the definition of eGovernance. The word "electronic" in the term e-Governance implies technology driven governance. E-Governance is the application of Information and communication Technology (ICT) for delivering government services, exchange of information communication transactions, integration various stand-one systems and services between Government-to-citizens (G2C), Government-to-business(G2B), Government-to-

Government(G2G) as well as back office processes and interactions within the entire government frame work.[1]. Through the e-Governance, the government services will be made available to the citizens in a convenient, efficient and transparent manner. The three main target groups that can be distinguished in governance concepts are Government, citizens and businesses/interest groups. In e-Governance there are no distinct boundaries.

B. M-App:

Mobile application (m-app) refers to a software program that has been developed for small low-power handheld devices such as mobile phones or tablets. These applications are either pre-installed on devices during manufacture, or downloaded by customers from app stores and other mobile software.[2].

C. M-Agriculture:

mAgriculture (also written as m-Agriculture), also known as e-Agriculture and eAgriculture, describes the improvement of agricultural, farming and rural development through enhanced communication information processes. Particularly, mAgriculture involves the conceptualization, design, development, monitoring, assessment and application of inventive methods to use Information and Communication Technologies (ICTs) in the rural and developing areas of the world, with a primary focus on agriculture.[3]

D. Android

Android is a mobile operating system running on the Linux Kernel. It was initially developed by Android Inc., a firm later purchased by Google, and lately by the Open Handset Alliance. It allows developers to write managed code in the Java language, controlling the device via Google-developed Java Libraries. The unveiling of the Android distribution on 5 November 2007 was announced with the founding of the Open Handset Alliance, a consortium of 47 hardware, software and telecom

companies devoted to advancing open standards for mobile devices. Google released most of the Android code under the Apache License, a free software and open source license.

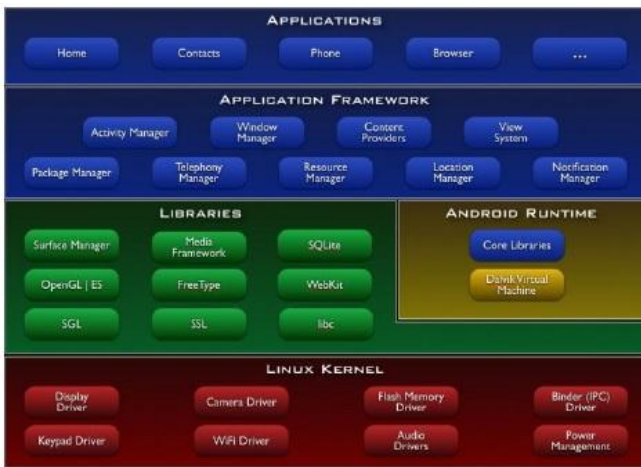


Figure 1: Android Architecture

Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack.[4]

E. Android SDK

The Android SDK includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator (based on QEMU), documentation, sample code, and tutorials. Currently supported development platforms include x86-architecture computers running Linux (any modern desktop Linux distribution), Mac OS X 10.4.8 or later, Windows XP or Vista. Requirements also include java Development Kit, Apache Ant, and Python 2.2 or later.

The officially supported integrated development environment (IDE) is Eclipse (3.2 or later) using the Android Development Tool. (ADT) Plugin, though developers may use any text editor to edit Java and XML files then use command line tools to create, build and debug Android applications. New platform technologies include; Optimized hardware speed, Support for more screen sizes and resolution, Revamped UI, New browser UI and HTML5 support, New contact lists, Better white/black ratio for backgrounds, Improved Google Maps 3.1.2, Microsoft Exchange support, Built in flash support for camera, Digital zoom, Improved virtual Keyboard. Android Applications are packaged in .apk format and stored under /data/app folder on the Android OS. The user can run the command adb root to access this folder as only the root has permissions to access this folder.

F. Php

PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic

web page content. It can also be used for command-line scripting and client-side GUI applications. PHP can be deployed on most web servers, many operating systems and platforms, and can be used with many relational database management systems (RDBMS). It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.

PHP primarily acts as a filter, taking input from a file or stream containing text and/or PHP instructions and outputs another stream of data; most commonly the output will be HTML. Since PHP 4, the PHP parser compiles input to produce bytecode for processing by the Zend Engine, giving improved performance over its interpreter predecessor.

Originally designed to create dynamic web pages, PHP now focuses mainly on server-side scripting, and it is similar to other server-side scripting languages that provide dynamic content from a web server to a client, such as Microsoft's Asp.net, Sun Microsystems' Java Server Pages, and mod_perl. PHP has also attracted the development of many frameworks that provide building blocks and a design structure to promote rapid application development (RAD). Some of these include CakePHP, Symfony, CodeIgniter, and Zend Framework, offering features similar to other web application frameworks.

G. Eclipse

Eclipse is an open source community, whose projects are focused on building an extensible development platform, run times and application frameworks for building, deploying and managing software across the entire software life cycle. Many people know us, and hopefully love us, as a Java IDE but Eclipse is much more than a Java IDE and, by means of various plug-ins, other programming languages including Ada, C, C++, COBOL, Perl, PHP, Python, Ruby (including Ruby on Rails framework), Scala, Clojure, and Scheme. The IDE is often called Eclipse ADT for Ada, Eclipse CDT for C/C++, Eclipse JDT for Java, and Eclipse PDT for PHP.

The initial code base originated from Visual Age. In its default form it is meant for Java developers, consisting of the Java Development Tools (JDT). Users can extend its abilities by installing plug-ins written for the Eclipse software framework, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules. Eclipse is free and open source software. Eclipse provides the Eclipse Rich Client Platform (RCP) for developing general purpose applications.[5]. The following components constitute the rich client platform:

- Equinox OSGi – a standard bundling framework
- Core platform – boot Eclipse, run plug-ins
- Standard Widget Toolkit (SWT) – a portable widget toolkit
- JFace – viewer classes to bring model view controller programming to SWT, file buffers, text handling, and text editors

- Eclipse Workbench – views, editors, perspectives, wizards

H. Xml

Extensible Markup Language (XML) is a set of rules for encoding documents in machine-readable form. It is defined in the XML 1.0 Specification produced by the W3C, and several other related specifications, all gratis open standards. The design goals of XML emphasize simplicity, generality, and usability over the Internet. It is a textual data format with strong support via Unicode for the languages of the world. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures, for example in web services. Many application programming interfaces (APIs) have been developed that software developers use to process XML data, and several schema systems exist to aid in the definition of XML-based languages.[6]. As of 2009, hundreds of XML-based languages have been developed, including RSS, Atom, SOAP, and XHTML. XML-based formats have become the default for many office-productivity tools, including Microsoft Office (Office Open XML), OpenOffice.org (OpenDocument), and Apple's iWork.

I. Xml Dom Parsing

The XML DOM defines a standard way for accessing and manipulating XML documents. The XML DOM views an XML document as a tree-structure. All elements can be accessed through the DOM tree. Their content (text and attributes) can be modified or deleted, and new elements can be created. The elements, their text, and their attributes are all known as nodes. The XML DOM is: A standard object model for XML. A standard programming interface for XML Platform- and language-independent, However, before an XML document can be accessed and manipulated, it must be loaded into an XML DOM object. A W3C standard World Wide Web Consortium. The XML DOM defines the objects and properties of all XML elements, and the methods (interface) to access them. In other words: The XML DOM is a standard for how to get, change, add, or delete XML elements.

J. Neural Networks

Neural network is an artificial framework that aids in proper reasoning based on the training provided. It is an artificial neural network where connection between the units do not form directed cycle.[7]

3. Methodology

To achieve the purpose of this research, functional system requirements are given as follows;

A. System Requirements

- i. Client Hardware and Software Requirements: Android Operating System based Mobile, 800GHz Processing Speed, 10 megapixel camera, Android Operating System Version 4.0.3.

- ii. Server Hardware and Software Requirements: Application Server including Web and Database Server, 2GB RAM, Processor Dual Core 2.44 GHz, Ethernet card, 300 GB of Storage, WAMP 2.0 – (Apache Tomcat 2.2.22, MySQL 5.5.24, PHP 5.3.13), Windows Vista SP2, Windows Server 2008 SP2, Java Run Time Environment version 7

B. System Overview

- i. Frontend: This system is a distributed social service system with client-server programming, which help people, especially in rural areas to get information related to agriculture. Here the frontend application is seen in screen of an android mobile. People using this system are supposed to be the client with their mobile phone, which will be implemented in JAVA programming language. High level design methodology and detailed design methodology for designing this system were used. With the help of this application client will be able to input the text or the image for their searching purpose. A simple GUI interface that any user can easily understand will be designed, Android SDK was chosen for designing the interface.
- ii. Backend: MY-SQL is used for storing data. Large amount of data are to be maintained for this application which are to be arranged properly and modified as per needs. A database administrator has the authentication to edit, modify and update the data of the database.

C. Technologies/Algorithms Used

The overall concept that is the framework for any vision related algorithm of image classification is shown below;

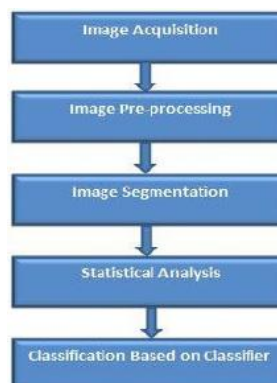


Figure 2: Disease Image Processing Block Diagram

D. Image Acquisition

Digital images are acquired on environment conditions using Digital Camera Sony Ericson 16.2 megapixels for creating test image database. Also test image dataset for mobile images are maintained using Samsung Galaxy (Android), 5mp camera. Then the image-processing techniques are applied those acquired image to extract useful features that are necessary for detection of disease in plants. Let's take following image of tomato's leaf with late blight disease for analysis.



Figure 3: Image Acquired

E. Image Pre-Processing

i. Contrast Stretching

In image processing, contrast stretching is a process that changes the range of pixel intensity values. Applications include photographs with poor contrast due to glare. Contrast Stretching is sometimes called Normalization. The purpose is to bring the image into a range that is more familiar or normal to the senses, hence the term is called normalization. Often, the motivation is to achieve consistency in dynamic range for a set of data, signals, or images to avoid mental distraction or fatigue.

The linear normalization of a grey scale digital image is performed according to the formula;

$$I_N = (I - Min) \frac{newMax - newMin}{Max - Min} + newMin \quad (1)$$

If the intensity range of the image is 50 to 180 and the desired range is 0 to 255 the process entails subtracting 50 from each of pixel intensity, making the range 0 to 130. Then each pixel intensity is multiplied by 255/130, making the range 0 to 255.[8].

F. K-Means Clustering Technique

K-Means Clustering is used for color image segmentation, which decomposes an image into meaningful partitions. Using K-Means, diseased portion of the leaf is extracted in separate cluster which help in further analysis of the type of disease contained in it.[9]. The algorithm used for fast image segmentation is as follows:

- i. Color Space Transform and Histogram Generation: HSI color space is chosen for this algorithm as HSI color space is capable of emphasizing human visual perception in hues and has an easily invertible transform from RGB. Thus, RGB color space is transformed to HSI color space.

First, convert RGB color space image to HSI space beginning with normalizing RGB values:

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B} \quad (2)$$

Each normalized H, S and I components are then obtained

by,

$$h = \cos^{-1} \left\{ \frac{0.5[(r-g)+(r-b)]}{\sqrt{[(r-g)^2+(r-b)(g-b)]^{1/2}}} \right\} \quad h \in [0, \pi] \text{ for } b \leq g \quad (3)$$

$$h = 2\pi - \cos^{-1} \left\{ \frac{0.5[(r-g)+(r-b)]}{\sqrt{[(r-g)^2+(r-b)(g-b)]^{1/2}}} \right\} \quad h \in [0, 2\pi] \text{ for } b > g \quad (4)$$

$$s = 1 - 3 \cdot \min(r, g, b); \quad s \in [0, 1] \quad (5)$$

$$i = \frac{R+G+B}{3 \times 255}; \quad i \in [0, 1] \quad (6)$$

For convenience h, s and i values are converted in the ranges of [0,360], [0,100], [0, 255], respectively, by:

$$H = \frac{h \times 180}{\pi}; \quad S = s \times 100 \quad \text{and} \quad I = i \times 255 \quad (7)$$

Since the range of tree dimension in HSI color space are not same (Hue: [0,360], saturation: [0, 1], and intensity: [0,256]), quantization process is performed to normalized the values in each dimension:

$$\begin{aligned} h' &= [\text{Hue}/h_Q], \\ s' &= [\text{Saturation}/s_Q], \\ i' &= [\text{Intensity}/i_Q] \quad (8) \end{aligned}$$

Where (h', s', and i') denotes the index and the quantization parameters $h_Q=120$, $s_Q=0.125$, $i_Q=32$ are set empirically to accentuate the importance of the hue and to save computational cost. [12].

ii. Max-min Initialization and Parameter Estimation

It consists of the three steps of Max-min algorithm, which are used in the preliminary stage for k-means.

- a. From the color histogram bins and grey histogram bins, find the bins which has the maximum number of pixels to be the first centroid.
- b. For each remaining histogram bin, calculate the min distance, which is the distance between it and its nearest centroid. Then the bin which has the maximum value of min distance is chosen as the next centroid,
- c. Repeat the process until the number of centroid equals to K_{max} or the maximum value of the distance in step (ii) is smaller than a predefined threshold Th_M . [13]

iii. K-Means Clustering in HSI color space

It includes the following steps:

- a. Estimate the parameters of k-means, including suitable value of K and the position of K initial centroids from the Max-min algorithm.
- b. Two kinds of histogram bins will be clustered together in this step. For color histogram bins, since the hue dimension is circular, the numerical boundary should be considered in the distance measurement and the process of centroid calculations. The distance measurement between a histogram bin vector B and a cluster centroid vector C in the current iteration t is defined in the form of the Euclidean distance:

$$D^2(B_i, C_j^{(t)}) = D_h^2(h_i, h_j^{(t)}) + (s_i - s_j^{(t)})^2 + (v_i - v_j^{(t)})^2$$

where

$$D_h^2(h_i, h_j^{(t)}) = \begin{cases} (\frac{360^\circ}{h_Q} - |h_i - h_j^{(t)}|)^2, & \text{if } |h_i - h_j^{(t)}| > \frac{180^\circ}{h_Q} \\ (h_i - h_j^{(t)})^2, & \text{otherwise.} \end{cases} \quad (9)$$

Next, classify each color histogram bin to its nearest cluster centroid by the distance measurement. Thus the membership function of histogram bin B_i is defined by:

$$\varnothing_{(j|B_i)}^{(t)} = \begin{cases} 1, & \text{if } j = \arg \min D^2(B_i, C_k^{(t)}) \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

c. Recalculate and update K cluster centroids. Again, since the hue dimension is circular, the indices in the hue dimension should be considered not absolutely but relatively. An efficient method is introduced to calculate the relative hue index of the original hue index h_i to the centroid C:

$$h_{i,j}^{(t)} = \begin{cases} h_i - \frac{360^\circ}{h_Q}, & \text{if } |h_i - h_j^{(t)}| > \frac{180^\circ}{h_Q} \text{ and } h_j^{(t)} < \frac{180^\circ}{h_Q} \\ h_i + \frac{360^\circ}{h_Q}, & \text{if } |h_i - h_j^{(t)}| > \frac{180^\circ}{h_Q} \text{ and } h_j^{(t)} > \frac{180^\circ}{h_Q} \\ h_i, & \text{otherwise.} \end{cases} \quad (11)$$

and the values in each dimension of all centroid vectors for the next iteration $C^{(t+1)}$ are updated according to the following equations:

$$h_j^{(t+1)} = \frac{\sum_{i=1}^{N_B} h_{i,j}^{(t)} \varnothing_{(j|B_i)}^{(t)} B(B_i)}{\sum_{i=1}^{N_B} \varnothing_{(j|B_i)}^{(t)} B(B_i)},$$

$$s_j^{(t+1)} = \frac{\sum_{i=1}^{N_B} s_i \varnothing_{(j|B_i)}^{(t)} B(B_i)}{\sum_{i=1}^{N_B} \varnothing_{(j|B_i)}^{(t)} B(B_i) + \sum_{i=1}^{N_G} \varnothing_{(j|G_i)}^{(t)} G(G_i)},$$

$$v_j^{(t+1)} = \frac{\sum_{i=1}^{N_B} v_i \varnothing_{(j|B_i)}^{(t)} B(B_i) + \sum_{i=1}^{N_G} v_i \varnothing_{(j|G_i)}^{(t)} G(G_i)}{\sum_{i=1}^{N_B} \varnothing_{(j|B_i)}^{(t)} B(B_i) + \sum_{i=1}^{N_G} \varnothing_{(j|G_i)}^{(t)} G(G_i)} \quad (12)$$

where $B(B_i)$ denotes the number of pixels in the color histogram bin with histogram bin vector B_i , and $G(G_i)$ denotes the number of pixels in the color histogram bin with histogram bin vector G_i . Note that the range of hue value of the new centroid should be normalized in the range of $[0, 360^\circ/h_Q]$. [14]

d. Image pixels are labelled with the index of nearest centroid of their corresponding histogram bins. A labelled image $l(x, y)$ is obtained in this step, and K-Means clustering is finished. [11].



Figure 4: Image Clusters 1, 2 and 3

G. Masking the Green Pixels and the Pixels on the Boundaries

This phase consists of two steps: The mostly green colored pixels are identified, and then the global image threshold using Otsu's method has been applied in order to specify the varying threshold value which chooses the threshold to minimize the interclass variance of the threshold black and white pixels. Next, the green pixels are masked as follows: if the green component of pixel intensities is less than the computed threshold value, then, the red, green and blue components of the this pixel are cleared. The next step in this phase is focused on deleting both the pixels with zero components and the pixels on the boundaries of the infected cluster(s).

H. Features Extraction

In the proposed approach, the method adopted for extracting the feature set is called the Color Co-occurrence Method or CCM method in short. It is a method, in which both the color and texture of an image are taken into account, to arrive at unique features, which represent that image.

i. Co-occurrence Methodology for Texture Analysis

The image analysis technique selected for this study was the CCM method. The use of color image features in the visible light spectrum provides additional image characteristic features over the traditional grey-scale representation. The CCM methodology established in this work consists of three major mathematical processes. First, the RGB images of leaves should be in Hue Saturation Intensity (HSI) color space representation, which has been done before in k-means clustering. Then, each pixel map is used to generate a color co-occurrence matrix, resulting in three CCM matrices, one for each of the H, S and I pixel maps. (HSI) space is also a popular color space because it is based on human color perception. Hue is generally related to the wavelength of a light and intensity shows the amplitude of a light. Lastly, saturation is a component that measures the "colourfulness" in HSI space. The color co-occurrence texture analysis method was developed through the use of Spatial Grey-level Dependence Matrices (SGDM's). The grey level co-occurrence methodology is a statistical way to describe shape by statistically sampling the way certain grey-levels occur in relation to other grey-levels. These matrices measure the probability that a pixel at one particular grey level will occur at a distinct distance and orientation from any pixel given that pixel has a second particular grey level. For a position operator p , a matrix P_{ij} was defined to counts the number of times a pixel with grey level i occurs at position p from a pixel with grey-level j . The SGDMs are represented by the

function $P(i, j, d, \Theta)$ where i represents the grey level of the location (x, y) in the image $I(x,y)$, and j represents the grey level of the pixel at a distance d from location (x, y) at an orientation angle of Θ . The reference pixel at image position (x, y) is shown as an asterix. All the neighbours from 1 to 8 are numbered in a clockwise direction. Neighbours 1 and 5 are located on the same plane at a distance of 1 and an orientation of 0 degrees. An example image matrix and its SGDM are already given in the three equations above. Here, a one and two pixel offset distance and a $0^\circ, 45^\circ, 90^\circ, 135^\circ$ orientation angle is used. After the transformation processes, calculation for the feature sets H and S were done while (I) was dropped since it does not give extra information. [15].

ii. Normalizing the CCM matrices

The CCM matrices are then normalized using Equation 13

$$P(i, j) = \frac{p(i, j, 1, 0)}{\sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} p(i, j, 1, 0)} \quad (13)$$

$P(i,j)$ is the image attribute matrix, $p(i,j,1,0)$ represents the intensity co-occurrence matrix, and Ng represents the total number of intensity levels.

The marginal probability matrix (P_x) can be defined as shown in Equation 14

$$P_x(i) = \sum_{j=0}^{Ng-1} P(i, j) \quad (14)$$

Sum and difference matrices (P_{x+y}, P_{x-y}) are defined as shown in Equations 15 and 16, respectively.

$$P_{x+y}(k) = \sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} P(i, j) \quad (15)$$

Where $k=|i-j|$; for $k=0,1,2,\dots,2(Ng-1)$.

$$P_{x-y}(k) = \sum_{i=0}^{Ng-1} \sum_{j=0}^{Ng-1} P(i, j) \quad (16)$$

iii. Color Moments

Color moments are very effective for Color based image analysis. The information of Color distribution in an image can be captured by the low order moments. The first& second order moment has been proved to be efficient and effective in representing Color distribution of image. [16].

Let P_{ij} is the i^{th} Color channel at the j^{th} image pixel. Then three Color moments can be defined as:

Moment 1: It is called Mean. It provides average Color value in the image. It is calculated using following statics: $Mean(\mu) = \frac{1}{N} \sum_{i,j} p_{ij}$ Where, N is the total number of pixels in the image.

Moment 2: It is called Standard Deviation. The standard deviation is the square root of the variance of the distribution. It is calculated using following statics:

Moment 3: It is called Skewness. It gives measure of the degree of asymmetry in the distribution. It is calculated using following statics:

$$Mean(\mu) = \sum_N^{j=1} \frac{1}{N} P_{ij} \quad (17)$$

The system was tested according to best practices with the functional requirements specified in system analysis phase. During this testing, special focus was on the betterment of the system's output and its targeted accuracy. Formulation of internal requirements of Neural Network was done for the most optimum result and finally it was successful in obtaining the targeted output with targeted accuracy. Stress testing was conducted to evaluate the system performance under stress. To test stress, the system was fed with more than 200 images or cases at once where the system reported error due to the memory limits which brought the system into low performance state or nearly to halt state.

iv. Neural Network

For classification purpose of disease, Multilayer perceptron neural network with Back propagation algorithm as a learning rule with linear momentum 0.8 and learning rule 0.7 was used. For training and testing the neural network, about 200 leaf images had been used. Out of these images about 135 images will be pre-processed, segmented and the corresponding texture and color features will be extracted. By the help of these extracted features, the Neural Network (NN) will be trained. The remaining images will be used validation purpose of the neural network. For the purpose of training and testing the neural network with extracted features are fed to multilayer feed forward neural network.

4. Results and Discussions

A. Feature Dimensionality Reduction

Using CCM matrices for feature extraction, total of 64 features texture features were computed for an image. It can be seen more that there is more redundancy in texture features as features are computed using CCM on 4 different directions (viz. 0, 45, 90, 135) with taking 1-unit distance and 2- unit distance. Therefore, texture features based on CCM are subjected to PCA algorithm for its dimensionality reduction.

Table 1: Features from HUE

Angular Moment	Entropy Feature	Inverse Difference	Sum Entropies	Difference Entropies	Information Measure of correlation	Correlation	Product Moment
0.921193	0.4264531	0.9698942	1701.1465	949.084609	0.48321106	104.930517	85.1951
0.909162	0.4788676	0.9589003	1620.7080	913.357839	0.39133856	80.4353708	62.7259
0.917608	0.4456006	0.9659167	1643.2900	949.506650	0.45254250	97.2604113	78.0737

0.914270	0.4596767	0.9628978	1643.0042	944.095174	0.43015549	92.2169339	73.2645
0.917439	0.4464428	0.9658454	1642.9076	949.283117	0.45287210	97.4997422	78.2300
0.908509	0.4820400	0.9585979	1619.3885	912.566258	0.39233328	81.1438559	63.1624
0.921474	0.4255063	0.9706512	1702.9396	935.021703	0.48592112	104.959510	85.2633
0.9141949	0.45950876	0.96301343	1664.17642	939.985983	0.43047354	90.20740317	71.58199

Table 2: Features from Saturation

Angular Moment	Entropy Feature	Inverse Difference	Sum Entropies	Difference Entropies	Information Measure of correlation	Correlation	Product Moment
0.9176075	0.46587450	0.96452391	10.5521373	9.04796849	0.45380505	61.6219849	44.58744
0.9211924	0.44725686	0.96812744	10.2352376	9.04953240	0.48346049	66.7518225	48.23946
0.9091610	0.49814098	0.95801103	11.5293268	8.96425948	0.39520199	48.3358261	35.18898
0.9142691	0.47876611	0.96206775	11.2416415	8.84361510	0.43388573	55.1247384	40.1279
0.9174388	0.46675909	0.96444969	10.5500474	9.04587838	0.45413588	61.7736300	44.67725
0.9085085	0.50145529	0.95770201	11.5219942	8.95692594	0.39621161	48.7616187	35.43538
0.9214729	0.44740154	0.96819529	9.85896405	9.41264754	0.48450078	68.7935244	49.95733
0.9141939	0.47945032	0.96195394	10.8361698	9.07248618	0.43260289	57.6376377	41.92914

Table 3: Feature For HUE After PCA

PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
0.30029	-0.93511	0.12556	0.14009	0.00091114	-9.6505e-006	-3.5703e-005	-2.4295e-005
0.16646	-0.11375	-0.94065	-0.27296	-0.0034915	4.4892e-005	0.0001507	0.00012752
0.93888	0.32179	0.11883	0.028939	0.00063337	-7.3098e-006	-2.7835e-005	-6.0232e-006
2.0077e-006	-5.3889e-006	3.7526e-007	-2.0024e-005	-0.045111	-0.16718	-0.98292	-0.062239
-1.0641e-006	-4.0621e-007	-8.6609e-006	-3.0085e-005	0.0084505	0.98511	-0.16492	-0.047785
-0.025019	0.095274	-0.29203	0.9513	0.0066515	-8.1961e-005	-0.00031261	1.9264e-005
-5.8911e-006	-2.6067e-006	0.00015286	0.00013153	-0.017741	0.03702	-0.0686	0.9968
-0.00012082	-0.00038342	-0.0015307	-0.0074341	0.99876	-0.015228	-0.044217	0.015299
2.3063e-013	1.5537e-008	4.6844e-007	1.1894e-006	1.4235	3436.7	6164.4	9.8681e+007

Table 4: Features for Saturation after PCA

PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
-0.041017	-0.32031	0.9395	0.11411	-0.0038143	0.0014813	-0.0031577	0.00051972
-0.12997	0.52026	0.26877	-0.79609	0.079242	-0.0058982	0.011383	-0.0021584
-0.99006	-0.027534	-0.067167	0.11917	-0.017641	0.0013548	-0.0022498	0.00042419
0.0015223	-0.065207	-0.01211	-0.12681	-0.69099	0.40903	-0.57502	-0.063995
0.0012024	-0.042596	-0.0045277	-0.081579	-0.63416	-0.022975	0.76725	0.012414
0.034788	0.78709	0.20102	0.56085	-0.15357	0.015285	-0.022072	0.0036365
0.00041058	-0.015332	-0.0024212	-0.032349	-0.20247	-0.50878	-0.20002	0.8117
-0.00037302	0.012473	0.0010955	0.026339	0.2218	0.75699	0.2001	0.58042
-4.7029e-021	3.3151e-009	3.4599e-008	7.6403e-006	0.0001387	0.0064431	0.023194	87.75

Table 5: Overall Accuracy

S.N	Classes	Total Number of Samples	Total Number of Training Samples	Total Number of Testing Samples	False Result	Overall Accuracy
1.	Normal	70	50	20	5	73.648%
	Disease	100	70	30	7	
2.	Septoria	50	35	20	4	
	Blight	50	35	20	3	

In the above case, the last three principal components cover more than 95 of the variability. That means the observations made in 8-dimensional space are converted to 2-dimensional space with this representation.

B. Accuracy of Neural Network

The proposed neural network was validated by both training and testing with samples collected. The neural network was first validated by using training samples with

100% accurate. Later the neural network was validated by using testing samples. The accuracy of the neural network in each of the stages for testing samples is shown in table below. The accuracy is 73.648% considering Hue and Saturation dimension.

5. Conclusion

The proper utilization of mobile's in today era can yield a better result for mankind. This project not only enhances the use of mobile phones but also makes an efficient and proper use of web source. In this research, the main target was on the development of the agriculture sector in rural areas as far as with the help of Android mobile. To great extent, this project will help develop the status of the rural areas by creating awareness among farmers about various sorts of diseases that can cause in plants and timely remedy of the caused disease. The main purpose of this project is to detect the plant diseases and to give proper advice to the user. The applications of Image processing algorithms and Neural Networks (NNs) have been formulated for clustering and classification of diseases that effect on plant leaves. Further it gives the appropriate solution to the user. Also, this system gives the information about the plants, insects and fertilizer. Local database is maintained to store to data regarding plants, its diseases, insects, fertilizers, human diseases and medicines. The information given to the user will be generated from the database. Future Enhancements to this work shall be any of the following components; Automatic Image segmentation from background, Disease detection of other plants, User's Interaction and implementation of the model in other aspects of human life like, health and education.

References

- [1] Bhattacharya, Kamal, and Tavneet Suri. "The Curious Case of e-Governance." *IEEE Internet Computing* 21, no. 1 (2017): 62-67.
- [2] Tutorial on Using Android for Image Processing Projects, EE368 Digital Image processing, Spring 2010, Linux Version
- [3] Gichamba, Amos, and Ismail A. Lukandu. "A model for designing M-agriculture applications for dairy farming." *The African Journal of Information Systems* 4, no. 4 (2012): 1.
- [4] Gandhewar, Nisarg, and Rahila Sheikh. "Google Android: An emerging software platform for mobile devices." *International Journal on Computer Science and Engineering* 1, no. 1 (2010): 12-17.
- [5] Henttonen, Katja. "Stylebase for Eclipse." An open source tool to support the modelling of quality driven software architecture (2007).
- [6] McGlone, Jonathan. "Preserving and Publishing Digital Content Using XML Workflows." *IDS Project Press*, 2013.
- [7] Ramírez-Quintana, Juan A., Mario I. Chacon-Murguia, and Jose F. Chacon-Hinojos. "Artificial neural image processing applications: A survey." *Engineering Letters* 20, no. 1 (2012): 68.
- [8] Jundare Manisha. A. Jundare Pallavi T. Jundare Pragati V. Prof. C.S.Aryan, "Plant Disease Detection

and its Treatment using Image Processing," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, Vol. 5, Issue 3, March 2016.

- [9] Tse-Wei Chen, Yi-Ling Chen, Shao-Yi Chien "Fast Image Segmentation Based on K-Means Clustering with Histograms in HSV Color Space" *Institute of Electronics Engineering and Department of Electrical Engineering National Taiwan University*, No. 1, Sec. 4
- [10] Hartigan, J. A.; Wong, M. A. (1979). "Algorithm AS 136: A K-Means Clustering Algorithm". *Journal of the Royal Statistical Society, Series C (Applied Statistics)* 28 (1):100– 108
- [11] Jayamala K. Patil, "Color Feature Extraction of Tomato Leaf Diseases", *International Journal of Engineering Trends and Technology*, Volume 2 issue 2- 2011
- [12] Sachin D. Khirade, A. B. Patil, "Plant Disease Detection using Image Processing," *International Conference on Computing Communication Control and Automation*, 2015
- [13] Fritz Albreghsen "Statistical Texture Measures Computed from Grey Level Concurrence Matrices", *Image Processing Laboratory, Department of Informatics, University of Oslo*, November 5, 2008
- [14] Amanpreet Kaur "Color Image Segmentation Based on Color and Texture properties" (*IJAEST*) *International Journal of Advanced Engineering Sciences and Technologies* Vol No. 8, Issue No. 2