International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-9 Issue-3 September 2020

Leaf Disease Detection using Labview Imaq Vision

Anusha Nellutla, Gnana Sai Ganesh Chittajallu, Shaik Feroz

Abstract - The intension of our project is to design a system which can identify the good leaves from the diseased ones. Image processing is a powerful tool capable of many applications. Image processing combined with Machine Vision can simulate and execute real time projects. In this project we have used LabVIEW along with IMAQ Vision to acquire real time images and process them. LabVIEW IMAQ Vision is potentially useful for agricultural products since it combines the merits of both LabVIEW and IMAQ Vision, which have graphical programming environment and rich image processing functions. The project aims to provide a brief introduction into the IMAQ vision components like Image Acquisition, Calibration, Defect detection. Major leaf diseases' symptoms include spots or discolouration of leaves. The presence or absence of macro and micro nutrients, bug infestation and other diseases can be identified through leaves. In this project we have obtained the images through LabVIEW IMAQ vision pallet. Further on two procedures were followed – one based on colour of the leaves and other is based on spots and patterns present on the leaves. For the discolouration we first split the image into its constituent planes- RGB and CMYK, here we used Green, Cyan and Yellow planes. Then on we decided a threshold based on sample data using Linear Regression based prediction model of Machine Learning to classify the data into three states – safe, risk and high risk.The second method was detecting spots. First, we split the images into its constituent planes to convert the RGB image to Greyscale and increase the contrast using the Colour Plane Extraction tool then use the Look up table tool to further enhance the contrast. Then on locate the bright objects and then using dilation from the Morphology tool box we increase the size of the spots to increase detection rate. Using Advanced Morphology tool box we removed the boundary objects to isolate the spots. Then using the shape detection or circle detection algorithm we can detect the spots. Several samples were obtained and are successfully classified. Finally, current limitations and likely future development trends are discussed. Combining LabVIEW along with different programming algorithms can help in raising the accuracy of the system.

Keywords—Image acquisition, colour plane extraction, Gray morphological operation, Edge detection, Real time Colour matching.

I. INTRODUCTION

Image processing is a method of performing some basic operations on an image, in order to get an enhanced image or to extract some useful information from the blurred or noisy pictures.

Revised Manuscript received on August 05, 2020.
Revised Manuscript received on August 05, 2020.
Manuscript published on September 30, 2020.
***** Correspondence Author

Anusha Nellutla, Assistant Professor Institute of Aeronautical Engineering Dept. of ECE E-mail.anushareddy.nellutla@gmail.com
Gnana Sai Ganesh Chittajallu, Institute of Aeronautical Engineering Dept. of ECE E-mail.ganesh.gnana49@gmail.com
Shaik Feroz, Institute of Aeronautical Engineering Dept. of ECE E-mail. ferozshaik025@gmail.com

It is type of signal processing in which input is an image and output maybe characteristics/features or image associated with that image

. Image processing form core research area within computer science and engineering disciplines too. Nowadays, image processing is among rapidly growing technologies.

There are majorly two types of methods used for image processing namely, analogue and digital image processing. Analogue image processing is used for hard copies like printouts and photographs.

Image analysts use different fundamentals of interpretation based on these visual techniques. Digital image processing techniques help us in the manipulation of various digital images by using various other algorithms and computers.

The basic definition of image processing can be said as processing of digital image, i.e. removing the noise and any kind of irregularities like blur, grain amount, etc., present in an image using the digital computer. The noise or irregularity may present into the image either during its formation or during transformation etc.

By using mathematical analysis, an image can be defined as a two-dimensional function like f (x, y), where x and y are spatial coordinates and the amplitude off. At any pair of coordinates present in (x, y) is referred as the gray level or intensity of the image at that particular point. When the (x, y) values and the intensity of f are all finite and discrete quantities, we can call the image as a digital image. A digital image is composed of a finite number of elements which is very important, each of part of the image has a particular value and location. These elements are known as picture elements, pixels and image elements. The most widely used term to denote the elements of a digital image is Pixel. When we are dealing with image processing there are some basic operations involve to enhance the image quality to identify the input image, which can be done by removing the blur amount, grain amount, etc., from the acquired image. For example, the image of space received by the space stations were always contains more noise, blur, gain, etc. By using these image processing techniques, the image is enhanced and make it as clear as we could recognize it. The basic steps involved in image processing can be discussed later. The main types of Image Processing Techniques are as follows,

- ➢ Image Acquisition
- ➢ Image Enhancement,
- ➢ Image Segmentation,
- ➢ Feature Extraction,
- ➢ Image Classification.

In the Indian economy, Agriculture is mainstay which comprises 18.5% of GDP (gross domestic product). In recent years in generating tremendous market values due to advance of cultivation technology the total cultivation areas and yield for agriculture product have increased rapidly.

The fruits and vegetables production is registering year on year growth and touching a new high in India according to Anon. It produces 198 million tons of fruits and vegetables. About 70% of India economy depends on agriculture, but due to environmental changes the crops get heavily affected. Even though India is having vast potential to emerge as major export of agriculture produce, but due to very high post-harvest losses in handling and processing, mismanagement of trades and procurement, lack of knowledge of preservation and quick quality evaluation technique the share in global market is very low of our country.

Due to increasing in population, there is need for growth of accurate fast and objective quality determination of agriculture product as there is increased expectation of food products of high quality & safety standards. The quality has been assessed traditionally by hand, inspecting the products individually or sampling large batches which is very time consuming and in consistent for the inspection application in agriculture. Leaf spot, dryness, color change and defoliation are characteristic symptoms of diseased leaf. Personal trainer is there to detect the defects of size, color, or strange feature and then classify in terms of color, texture or different types of defects hinder their classification.

Every action that is needed to control a process efficiency as controlled by system is Automotive system which are more precise. The automotive qualitative inspection on agriculture produces that have been reflected the progress of machine vision application and have much interest. Shape classification, defects Detection, quality grading, etc... is the application of the technique used.

At the initial or early stages image processing helps to support farmers in the identification of disease in leaf. For the plant disease detection color analysis texture, feature extraction, image segmentation, image processing is **c**arried out. The different algorithms are used for detection of disease. Threshold methods were used as feature extraction to identify the disease of leaf using image processing technique

Identification of diseased leaf using image processing technique which involves morphological processing, color clustering, LABVIEW IMAQ vision leaf disease detection. It applied border segmentation and pattern classification techniques for detection of diseased leaf.

II. LITERATURE REVIEW

1. LabVIEW

Laboratory Virtual Instrument Engineering Workbench (LabVIEW) is the abbreviation of LABVIEW and it is basically used for development environment and system-design platform for a visual programming language from National Instruments. The initial release of LabVIEW was 1986. LabVIEW is basically a graphical language and it is commonly used for data acquisition, instrument control and industrial automation. LabVIEW mainly consists of two windows called front panel and block diagram.

Front panel is used for obtaining results like graphs, output values etc. It is the user interface of a VI. Generally, we design the front panel first and then we design the block diagram to perform tasks on the inputs and outputs which we created on the front panel.

Block diagram is used for designing the program. The block diagram generally contains the graphical source code of a LabVIEW program. The concept of the block diagram is to separate the graphical source code from the user interface in a simple and logical manner.

i. Data Flow Programming

The programming paradigm used in LabVIEW is called as "G". which is based on data availability. When there is enough data availability to a sub VI, that sub VI will execute. Execution flow is determined by the structure of a graphical block diagram in which the programmer connects different functional nodes by drawing wires between the components. These connected wires propagate variables and data which executes as soon as all the input data becomes available. LabVIEW can execute codes inherently in parallel.

ii. Graphical Programming

LabVIEW integrates the creation of user interfaces into the development cycle. LabVIEW programs-subroutines are termed virtual instruments (VIs). Each VI has three components namely a block diagram, a front panel, and a connector pane. The connector pane is used to represent the designed VI in the block diagrams of other. The front panel is built using indicators and controls. Controls are inputs which allows a user to supply information to the VI. Indicators are outputs while they indicate, or display the results based on the inputs given to the VI.

The Functions palette contains the VIs, constants and functions is used to create the block diagram. To access the Function palette from the block diagram we need to select View » Functions Palette. The Functions palette is divided into various categories; you can show and hide categories to suit your needs. **Figure 2.1** shows a function palette with all of the categories labelled and the programming category expanded.

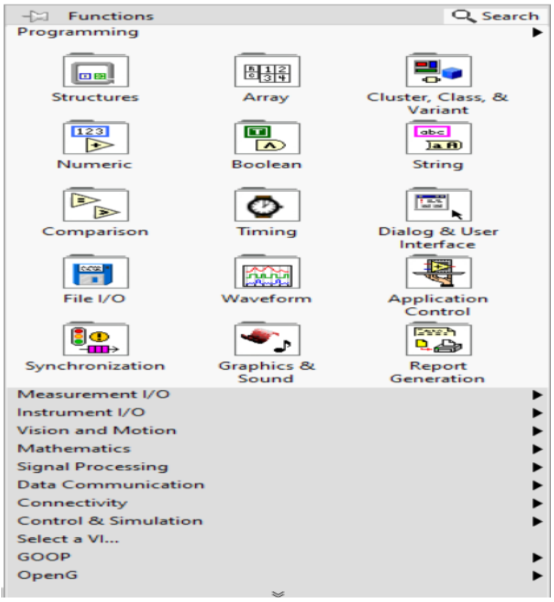


Figure 2.1. LabVIEW function palettes

2. IMAQ Vision

NI-IMAQ (Image Acquisition) is a add-on for LabVIEW to perform advanced image processing techniques. There is a separate palette for IMAQ Vision in Block diagram named as vision and motion to implement various image processing techniques to an image of any type as shown in fig. 2.2.

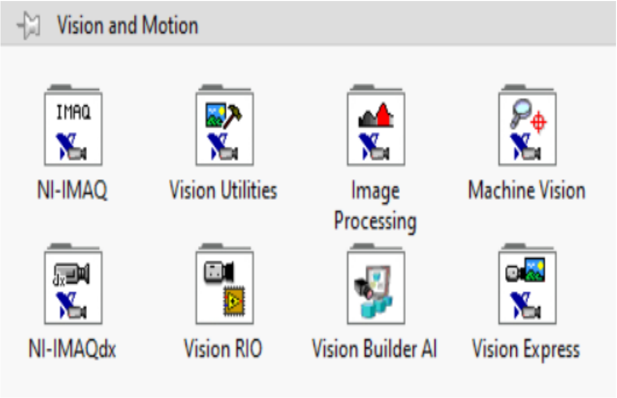


Figure 2.2. Vision and motion palette

Types of Images:

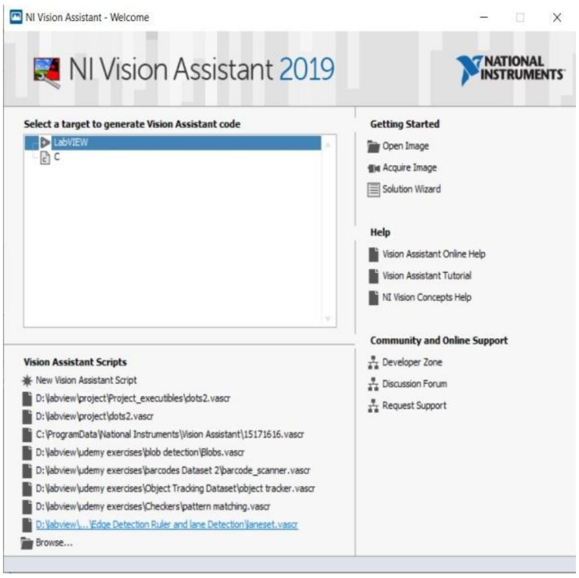
The NI-IMAQ vision libraries can manipulate three types of images like grayscale, color, and complex images. An image is a two-dimensional array of values representing light intensity. An image's resolution can be represented as the number of rows and columns of pixels. A typical image consists of *m* columns and *n* rows which has a resolution of m × n. The different types of images and its properties is shown in the below table 2.1.

Table - 2.1: Bytes per pixel of an image

RGB Color (4 bytes or 32-bit)	8-bit for the alpha value (not used)	8-bit for the Red intensity	8-bit for the Green intensity	8-bit for the Blue intensity
HSL Color (4 bytes or 32-bit)	8 bits not used	8-bit for the Hue	8-bit for the Saturation	8-bit for the Luminance
Complex (8 bytes or 64-bit)	32-bit floating for the Real part		32-bit floating for the Imaginary part	

3. NI Vision Assistant:

NI Vision Assistant is a tool for testing and prototyping image processing applications. To prototype an image processing application, we can build custom algorithms with the help of Vision Assistant scripting feature. The scripting feature records each and every step of the processing algorithm.



NI Vision Assistant - Welcome

NI Vision Assistant 2019

NATIONAL INSTRUMENTS

Select a target to generate Vision Assistant code

- LabVIEW
- C

Getting Started

- Open Image
- Acquire Image
- Solution Wizard

Help

- Vision Assistant Online Help
- Vision Assistant Tutorial
- NI Vision Concepts Help

Community and Online Support

- Developer Zone
- Discussion Forum
- Request Support

Vision Assistant Scripts

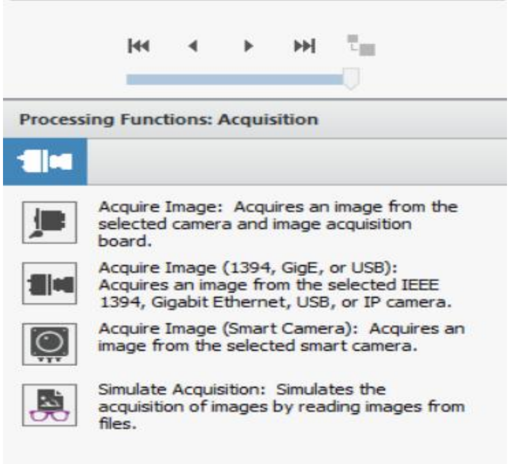
- New Vision Assistant Script
- D:\labview\project\Project_executibles\dots2.vascr
- D:\labview\project\dots2.vascr
- C:\ProgramData\National Instruments\Vision Assistant\15171616.vascr
- D:\labview\udemy exercises\blob detection\Blobs.vascr
- D:\labview\udemy exercises\barcodes Dataset 2\barcode_scanner.vascr
- D:\labview\udemy exercises\Object Tracking Dataset\object tracker.vascr
- D:\labview\udemy exercises\Checkers\pattern matching.vascr
- D:\labview\...\Edge Detection Ruler and lane Detection\laneset.vascr
- Browse...

Figure 2.3. Linking Vision Assistant to LabVIEW

i. Introduction to Image Processing with Vision Assistant:

Vision assistant is used to load images and to perform various processing images on the image. To acquire image into Vision Assistant software we can either select acquire image from a camera connected to the computer or we can browse from the images present in a specific folder. The following figure shows how to acquire an image.

After acquiring the image into vision assistant, we can perform various image processing operations on it using the built-in options in as shown in fig. 2.5.



Processing Functions: Acquisition

Acquire Image: Acquires an image from the selected camera and image acquisition board.

Acquire Image (1394, GigE, or USB): Acquires an image from the selected IEEE 1394, Gigabit Ethernet, USB, or IP camera.

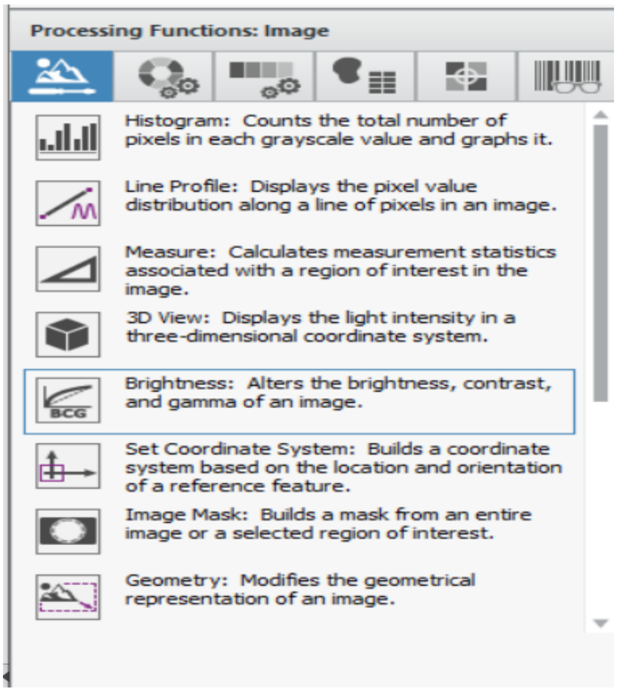
Acquire Image (Smart Camera): Acquires an image from the selected smart camera.

Simulate Acquisition: Simulates the acquisition of images by reading images from files.

Figure 2.4. To acquire image from a camera

Retrieval Number: 100.1/ijrte.C4574099320
DOI:10.35940/ijrte.C4574.099320
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication



Processing Functions: Image

Histogram: Counts the total number of pixels in each grayscale value and graphs it.

Line Profile: Displays the pixel value distribution along a line of pixels in an image.

Measure: Calculates measurement statistics associated with a region of interest in the image.

3D View: Displays the light intensity in a three-dimensional coordinate system.

Brightness: Alters the brightness, contrast, and gamma of an image.

Set Coordinate System: Builds a coordinate system based on the location and orientation of a reference feature.

Image Mask: Builds a mask from an entire image or a selected region of interest.

Geometry: Modifies the geometrical representation of an image.

Figure 2.5. Processing functions of an image

4. Basic Operations on image:

i. Colour transformation:

Colour of an image can be described using its red (R), green (G) and blue (B) coordinates, called RGB system. Similarly, it also can be represented as CMY (Cyan, Magenta and Yellow), HSL (Hue, Saturation, Luminance). These parameters are used to enhance the images to make clearly visible. Color transformation generally converts an image that is represented in one color space to another color space, whose final goal is to make the translated image looks as similar as possible to the original one.

By using the LabVIEW's built-in functions, we can easily perform these types of operations with the help of NI-IMAQ Vision. The below Block diagram is used to perform color transformation.

In the above block diagram, first we give an input image of type either RGB (U64), or HSL (U32).

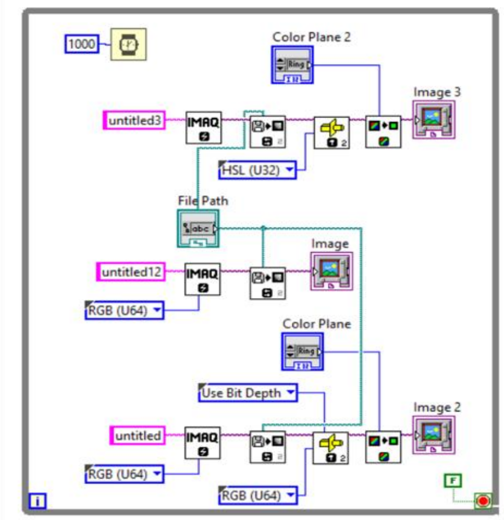
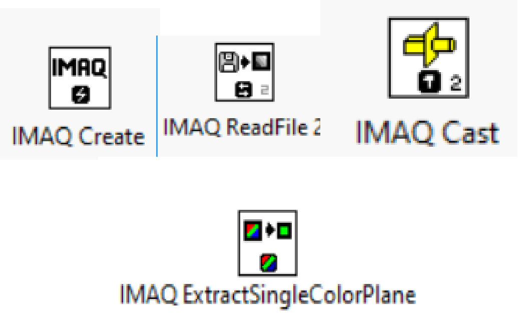


Figure 2.6. color transformation Block diagram



The IMAQ create palette grabs the image and creates a temporary memory location for the input image. Now the IMAQ Read palette reads an image of any type like JPEG, PNG, BMP, etc., The type casting is used to convert an incompatible image type of one form to another. After this IMAQ extract single color plane is used to extract a desired single color from the pixels of an image as shown in figure 2.7. The resultant output is shown in the front panel. Using this extract single color plane palette we can extract any color easily.

ii. Morphological operations

Morphology operations is a set of image processing operations that process the images based on shapes. During the morphological operations each pixel in the image is adjusted based on the value of other pixels in its neighbourhood. The basic morphological techniques are dilation, erosion, reconstruction. The below figures 2.8 & 2.9 explains well how an image is dilates and erodes.

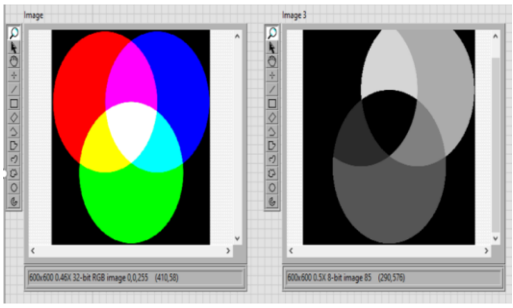


Figure 2.7. Results of color transformation in front panel

a) Dilation:

Morphological Dilation adds pixels to the boundaries of objects to an image. It adds pixels to the neighbour pixels by comparing input image with structuring element. The below figure states how dilation of pixels is performed in image processing.

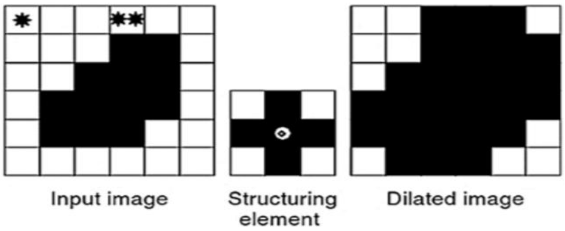


Figure 2.8. Dilation

b) Erosion:

Morphological Erosion removes pixels on the boundaries of objects in an image. It removes pixels to the neighbour pixels by comparing input image with structuring element. The below figure states how erosion of pixels is performed in image processing.

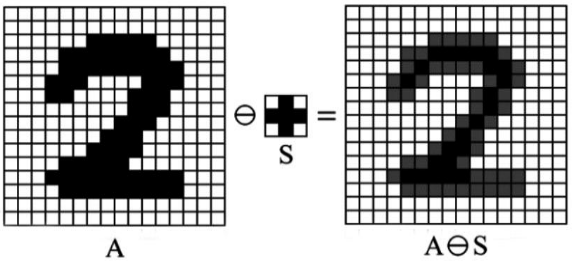


Figure 2.9. Erosion

iii. Dilation and erosion using LabVIEW:

The Dilation and Erosion of an image is performed using LabVIEW IMAQ vision as follows.

From the fig2.10 block diagram we created a temporary memory location using IMAQ create and read that image file into block diagram.

IMAQ consists of a built-in palette which performs morphological operations easily. After all the pixels of an image dilates and erodes, we get the output in the front panel as shown in figure 2.11. The image gray morphology performs the gray scale morphological transformation, dilation and erosion to each and every pixel in an image.[6]

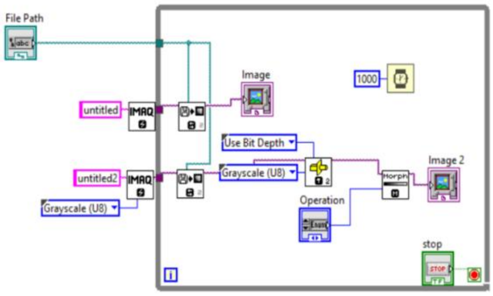


figure 2.10. Dilation and erosion using LabVIEW

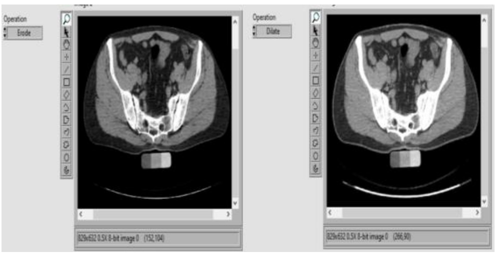


figure 2.11. output of dilation and erosion in front panel

III. METHODOLOGY

1. Image Acquisition using NI Vision Assistant:

In Image processing, Image acquisition is the first method and it uses a digital camera to capture the image. The image is stored in digital media or storage to perform further LabVIEW operations. The captured healthy and unhealthy images of plant leaves can be acquired using the NI Vision Assistant as shown in fig 3.1.

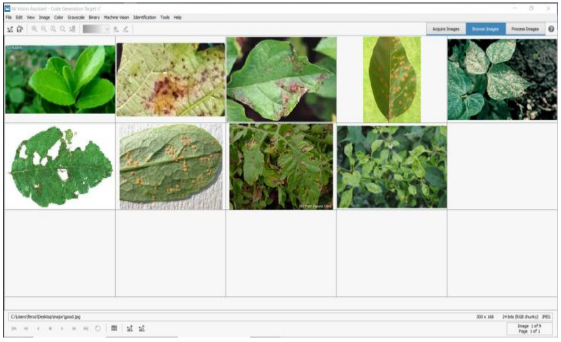


figure 3.1. Acquiring captured leaf images using NI Vision Assistant.

Vision assistant is tool which will be alongside the LabVIEW IMAQ tools. The Vision assistant is compatible with both LabVIEW and C codes. It can export to any one of those codes depending on the application. The vision assistant algorithm can be used in LabVIEW in two ways, one by directly by calling the script file using the Ni Vision Assistant tool from the Vision Express palette as shown in figure 3.2 and the other is to directly export the script file into the LabVIEW file.

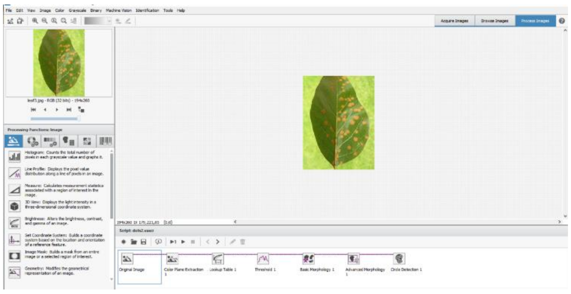


Figure 3.2. Acquiring images into NI Vision Assistant.

Our script file aims to isolate the spots on the leaf. The first step in vision assistant is to acquire the images. We can acquire the images either from the camera as datasets or browse from the already acquired ones. We have used the already acquired ones. The first step after acquiring images is to extract the colour planes and converting the original image to the 8-bit unsigned type. We have done that using the Extract single colour planes option as shown in figure 3.3 and extracted the red plane where in the contrast between spots and the rest of the leaf be highlighted.

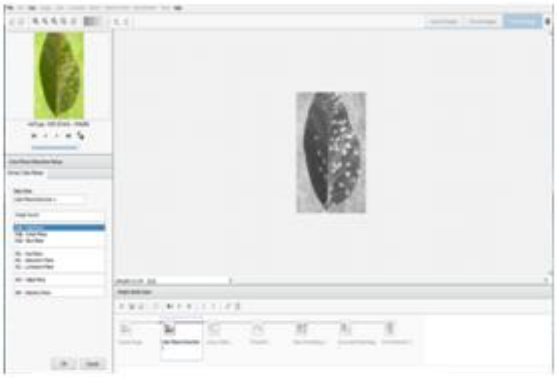


Figure 3.3. To extract single color plane

The contrast can be further increased by using an LUT. LUT stands for look up table and it is used for binarizing the image. Further on we used the thresholding operator to look for the bright objects and adjusted the threshold such that the spots of the leaf would be highlighted. The basic morphological operation of dilate is used which is clearly shown in figure 3.4.

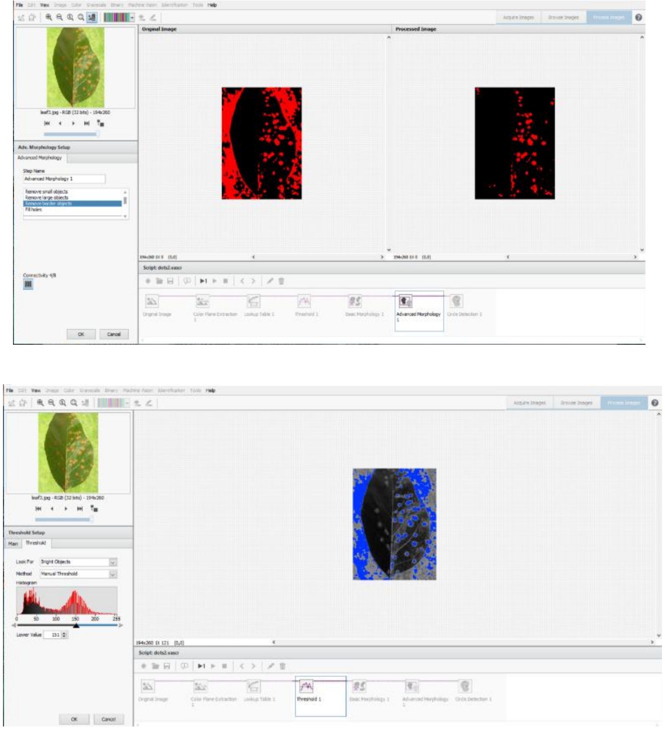
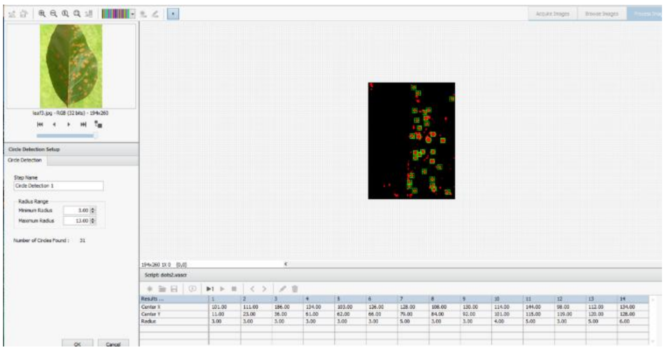


Figure 3.4 : To highlight the spots on the leaf

As said in the paper [1], To improve the image data contained unwanted distortions or to enhance some image features, image pre-processing is used. Various techniques used in pre-processing method such as change in size and shape of image, noise filtering, conversion of image, image enhancement and morphological operations. To resize image, to enhance contrast and RGB to grey scale conversion, various LabVIEW codes (VI's) are used.

In this step we perform some basic morphological operations and color plane extraction to enhance the image and we create overlays which is placed on the detected circles where the plant is damaged as shown in fig 3.2. NI Vision Assistant provides flexibility to create the VI's easily by just selecting the option create LabVIEW VI.

To enlarge the spots. The neighbourhood is selected as 3*3 with adjacency as N4(p) and a single iteration. Advance morphology is used to remove the border objects such that only the spots remain. Then on we can use a shape detector or a circle detector to obtain the data of these spots. We here used a circle detector. The circle detector gives the following data: the number of circles, the coordinates of the centre, the radius and the core area as shown in figure 3.5. The centre is given in terms of 2 integers x and y. The radius is also an integer and the core area being the number of pixels being enclosed within the boundary of the circle, also an integer.



Results ...	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Center X	101.00	111.00	186.00	134.00	103.00	126.00	128.00	108.00	130.00	114.00	144.00	98.00	112.00	134.00
Center Y	11.00	23.00	38.00	61.00	62.00	66.00	79.00	84.00	92.00	101.00	115.00	119.00	120.00	128.00
Radius	3.00	3.00	3.00	3.00	3.00	3.00	5.00	3.00	3.00	4.00	5.00	3.00	5.00	6.00

Figure 3.5. Data of the spotted circles in damaged leaf

The output of the script file here is a cluster with the above parameters which is further called to provide an overlay to the circles. The overlay is created using Create oval overlay tool in from the Overlay palette. The oval overlay takes up an image, a colour of our choice to create the overlay and a cluster. The cluster defines the bounding rectangles of the oval which calculated by adding and subtracting the radius to the centre coordinates as shown in figure3.6. .

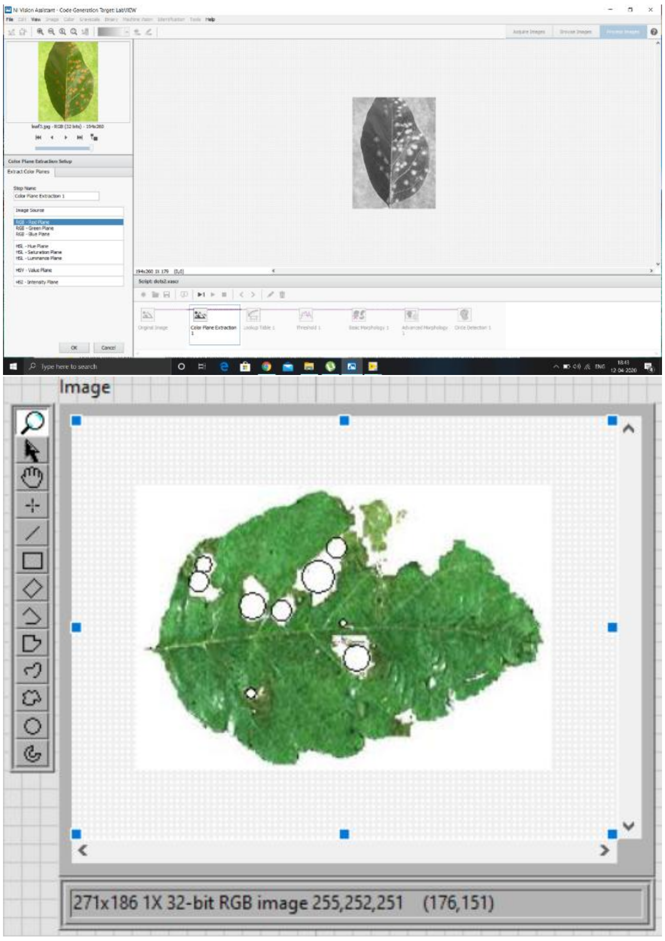

NI Vision Assistant - Code Generation Target: LabVIEW
Color Plane Extraction Setup
Extract Color Planes
Step Name: Color Plane Extraction 1
Image
271x186 1X 32-bit RGB image 255,252,251 (176,151)

figure 3.6. Circle overlay on damaged leaves

2. Image Segmentation:

Image segmentation is process, where the conversion of captured image into several segments and rendering of an image into something for easier analysis. The LabVIEW block diagram for performing image segmentation is given in the fig. 3.7

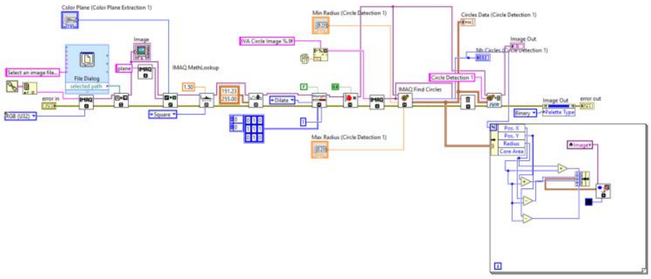

Color Plane (Color Plane Extraction 1)
IMAQ MathLookup
Min Radius (Circle Detection 1)
Circles Data (Circle Detection 1)
IMAQ Find Circles
Max Radius (Circle Detection 1)
Image Out

Figure 3.7. segmentation of an image

After creating memory location for the acquired image, we read it using IMAQ read palette. Now the single color plane extraction extracts a single color (red) and sends the pixel values to the math lookup table palette.

IMAQ MathLookup

IMAQ math Lookup is used to convert the image by replacing them with values from a defined lookup table. This VI modifies the dynamic range of either a specific part of an image or the complete image according to the image we chose.[6]

IMAQ Threshold

IMAQ Threshold is used to applies a threshold value to an image. The default lower threshold pixel value is 128 and the default lower threshold pixel value is 255.[6]

IMAQ Morphology

IMAQ Morphology Performs primary morphological transformations. The connected input image for a morphological transformation must be created with a border capable of supporting the size of the structuring element. For example, if we create a structuring element of size 3 × 3 it requires a minimum of border 1. Similarly, we need minimum of 2 borders if we create 5 × 5 structuring element.[6]

IMAQ RejectBorder

IMAQ Reject Border is used to eliminate the pixels which touches the border of an image on which we created the overlays.[6]

IMAQ Find Circles

IMAQ Find Circles finds the radius and centre of the circular particles even when multiple circular particles overlap. It detects and Separates overlapping circular particles based on the classification of their radius, surface area and perimeter.[6]

IVA Store Circles Results

IVA store Circle Results stores the matches found by pattern matching function in the point array.[6]

IMAQ Overlay Oval

IMAQ overlay oval is used to overlays an oval on an image.[6] As the image is read from the memory basic morphological operations is done on every pixel of an image and performs basic pre-processing tasks like extracting color planes and creating overlays on the spots where the leaf is damaged and this data is forwarded to store the overlays on the leaf IVA store circle results palette. The positions of the overlays which is present in the form of X & Y coordinates will be stored and displayed on the output screen in the front panel as shown in the figure 3.8.

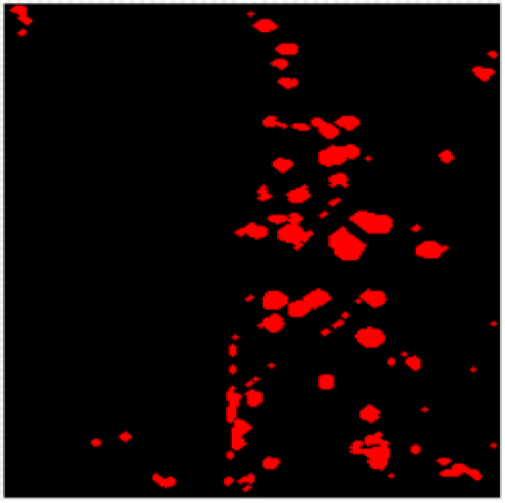


figure 3.8. Image segmentation and overlay to a damaged leaf

3. Extraction of RGB and CMYK planes:

Another approach to classify the disease is by the discolouration of the leaves. In the reference [2], the author specifies the algorithm to classify the chilli plant diseases based on their colour. The author provides insight on the different colour classifications RGB and CMYK. RGB stands for Red, Blue and Green whereas CMYK is the complementary counter part of RGB comprising of Cyan, Magenta, Yellow and Black. We have opened the images in LabVIEW and casted it into the RGB (U32 format). After the casting we have used the Extract colour planes VI present in the Colour processing sub-palette of the Vision and Mission palette. The colour plane extraction vi splits the input colour image into the respective planes of Red, Blue and Green. Each of the planes is a standard Unsigned integer of 8 bits. The intensity of the pixels ranging from0-255. A pixel containing more than 80% of the maximum value can be said to represent the original colours. The method to convert RGB to CMYK is by taking the compliment of RGB pixels. We have done that in LabVIEW by creating a white image with the same size as that of the original input image using the Set Image Size and Fill Colour VIs present in the Pixel Manipulation Sub palette.

IMAQ create session is used for every new image to be displayed. Further on by using the IMAQ Subtract vi and subtracting the RGB planes obtained from Extract Planes vi and the white image we obtained the CMYK planes. Further on using the CMYK and the RGB planes obtained, we used two for loops and a Get Pixel Value to count the number of active pixels per plane and calculated the percentage of each colour. The total area of the image is obtained by using the property node as shown if fig 3.6 and getting the image size from it. The same image size parameters ae used as the controls in the for loops and the fill image vi.

After calculating the percentages as per the author's data we are considering Yellow, Cyan and Green percentage of the image.

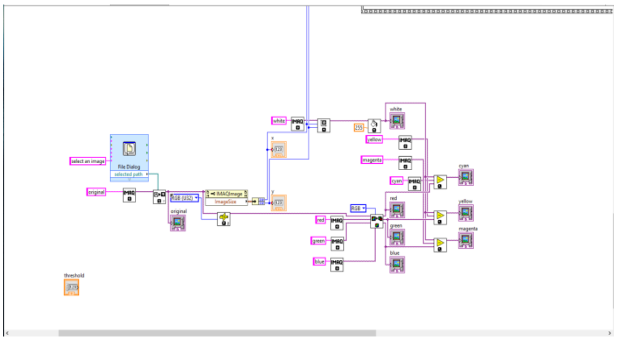


Figure 3.9 : Creating memory locations for image and reading it to get RGB and CMYK images.

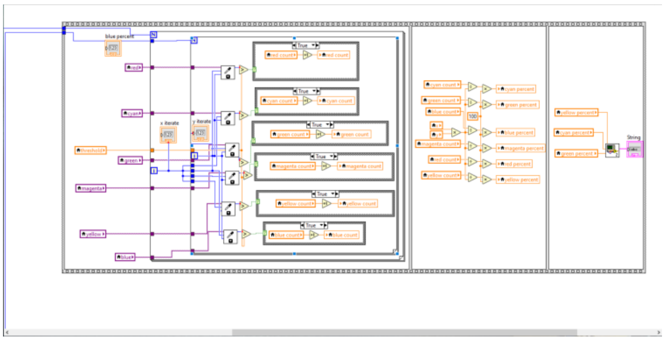


Figure3.10: Flat sequence to process the images.

We have taken the data from the author's paper [2] and used a Linear Regression algorithm to a build a system of classification to classify the images into 3 stages based on the Cyan, Yellow and Green percentages. We have created a sub vi to integrate this python code as shown in figure 3.12 with LabVIEW.

4. LabVIEW integration with Python

In LabVIEW there is a specific palette for the integration of python codes into LabVIEW as shown in figure 3.11

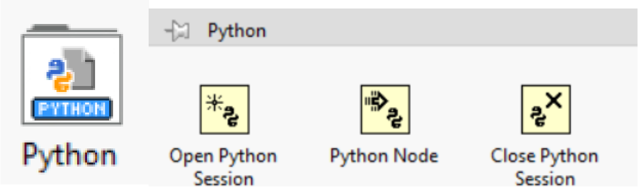


Figure 3.11 : Python palette in LabVIEW

The python integration toolkit comprising of 3 blocks has been used. The first being Open python session which takes up the python version as the input parameter and creates a python session ID. The next VI is Python node vi which takes up the previous session ID and the path to the python code and the method name to be invoked and the input and output parameters of the python file. The last being Close python session which takes up a session id and closes it. In our case the python code takes in 3 float parameters that is the percentages of the Yellow, Cyan and Green and gives a number output. The number output classifies the input percentages as being Safe, Risky and High risk. The output of the python code is converted into string type using The Case Structure from the Structures palette.

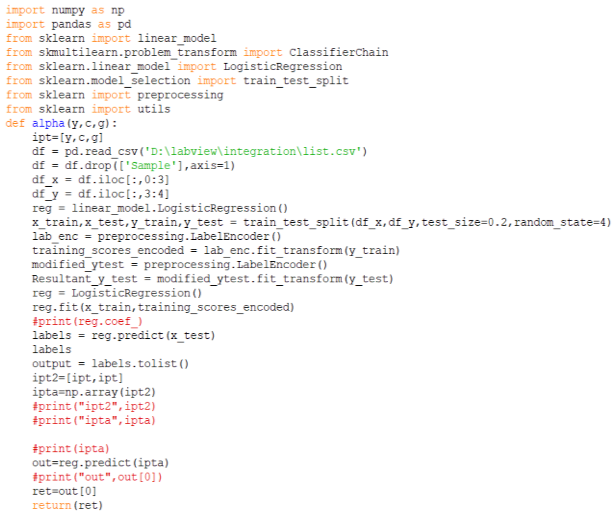
As the overlay images are processed for the damaged plants, now we have to predict the level of damage caused to these leaves to determine which diseases is caused to the leaves using the below table 3.1, which show the threshold values for a particular color of a leaf. By using these values reading from the python code provided in figure 3.8, it takes 3 input values and process them to give a output which says the given leaf is healthy or not.

figure 3.1: The sample values of plants healthy and risky result [2]

Sample	Green	Cyan	Yellow	Disease
1	0	0	5.034	1
2	0	0	2.597	0.5
3	0	10.347	0.035	0
4	0	13.511	2.917	0.5
5	0	14.474	0.413	0
6	0	18.116	0	0
7	0.05	18.519	0.19	0
8	0.06	24.711	0	0
9	0.293	11.199	0.003	0
10	0.35	12.911	4.336	0.5
11	0.608	28.316	0	0
12	0.639	17.073	0	0
13	0.79	3.234	4.022	0.5
14	0.793	2.102	0	0
15	0.835	1.454	0.04	0
16	0.858	0	8.356	1
17	0.936	0	0.139	0
18	1.006	0	6.928	1
19	1.088	9.507	2.104	0.5
20	1.137	0.417	3.698	0.5
21	1.266	0	3.151	0.5
22	1.646	3.719	1.802	0.5
23	1.668	0	1.141	0.5
24	2.031	0	3.696	0.5
25	2.246	10.842	2.715	0.5
26	2.246	0	2.009	0.5
27	2.349	15.543	1.067	0.5
28	2.524	7.758	1.737	0.5
29	2.595	0	3.786	0.5
30	2.848	0	3.601	0.5
31	3.367	7.614	2.743	0.5
32	3.706	0	1.694	0.5
33	3.772	0.023	0.03	0
34	3.789	0	2.539	0.5

35	3.917	0	1.192	0.5
36	3.922	0	3.946	0.5
37	4.008	0	6.074	1
38	4.161	2.5	4.454	0.5
39	4.166	0	8.966	1
40	4.259	0	0	0
41	4.595	0	4.066	0.5
42	4.733	0	2.55	0.5
43	5.405	0	1.154	0.5
44	5.455	19.67	0	0
45	5.57	0	3.661	0.5
46	5.594	0	1.146	0.5
47	6.128	0.462	0	0
48	6.338	0.25	1.044	0.5
49	6.556	1.927	4.558	0.5
50	6.675	0	1.405	0.5
51	6.748	0	1.308	0.5
52	7.029	0	4.323	0.5
53	7.302	0	7.115	1
54	7.415	0.002	9.809	1
55	7.439	0	2.198	0.5
56	7.828	0	6.351	1
57	7.919	0	2.017	0.5
58	7.926	0.443	0.226	0
59	8.305	0	4.117	0.5
60	8.33	0	4.225	0.5
61	8.46	0	4.523	0.5
62	8.886	0.536	5.173	0.5
63	9.139	0	7.743	1
64	9.266	1.831	1.309	0.5
65	9.351	12.605	0.067	0
66	9.456	0	3.542	0.5
67	9.689	0	12.621	1
68	9.814	1.377	3.337	0.5
69	9.974	0	1.515	0.5
70	10.29	0	3.379	0.5
71	10.385	0	1.453	0.5
72	10.491	0.44	3.351	0.5
73	11.057	0	1.038	0.5
74	11.15	28.41	3.121	0.5
75	11.181	0	16.973	1
76	11.402	0	6.739	1
77	11.557	0	13.251	1
78	12.1	0	3.606	0.5
79	12.209	0.521	7.044	1
80	12.555	0	3.774	0.5
81	12.717	0.357	1.386	0.5
82	12.857	6.084	1.212	0.5
83	13.402	0	5.362	0.5
84	14.049	0.279	0.216	0
85	14.129	24.883	1.107	0.5
86	14.232	0	2.488	0.5
87	14.458	0	1.085	0.5
88	14.689	0	3.97	0.5
89	15.077	0	1.828	0.5
90	15.92	0.138	2.451	0.5
91	16	3.059	1.126	0.5
92	16.418	0.264	2.375	0.5
93	16.837	0	12.183	1
94	17.16	0.001	1.825	0.5
95	17.21	0	1.269	0.5
96	17.293	0	3.775	0.5
97	17.315	0	1.317	0.5
98	17.391	0	1.289	0.5
99	17.61	0.171	2.373	0.5
100	17.863	0	2.656	0.5
101	18.351	0.09	0.519	0
102	18.581	0.52	0.731	0
103	18.839	0	2.356	0.5
104	19.905	0	2.932	0.5
105	20.417	0.06	0.828	0
106	20.417	0.06	0.828	0
107	20.432	0.067	0.914	0
108	20.93	0.029	0.83	0
109	21.218	0.163	0.739	0
110	21.426	0.067	0.835	0
111	21.441	14.954	1.248	0.5
112	22.916	0.007	0.014	0
113	23.127	0.16	0.624	0
114	26.58	0.934	5.312	0.5
115	26.718	0	1.688	0.5
116	27.854	0.011	4.078	0.5
117	27.91	2.382	9.653	1
118	28.157	0	12.981	1
119	29.625	0	8.397	1
120	31.742	0	0.521	0

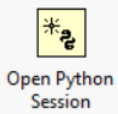
As said in [2], this technique is to be used for early detection of plant disease through leaf inspection. Leaves images captured are processed to determine the healthiness of each plant. By using leaf samples in RGB color model technique, it will identify the potential problems to the plants before its goes seriously damage for all the other leaves in the field. The RGB values extracted using the python code using from figure 3.8, will search for the closest match with the above table 3.1 using Predictive Linear Regression model of Machine Learning [7] and the resultant values decides the healthiness of the leaves.



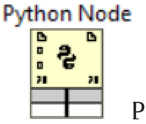
```
import numpy as np
import pandas as pd
from sklearn import linear_model
from skmultilearn.problem_transform import ClassifierChain
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn import utils
def alpha(y,c,g):
    ipt=[y,c,g]
    df = pd.read_csv('D:\labview\integration\list.csv')
    df = df.drop(['Sample'],axis=1)
    df_x = df.iloc[:,0:3]
    df_y = df.iloc[:,3:4]
    reg = linear_model.LogisticRegression()
    x_train,x_test,y_train,y_test = train_test_split(df_x,df_y,test_size=0.2,random_state=4)
    lab_enc = preprocessing.LabelEncoder()
    training_scores_encoded = lab_enc.fit_transform(y_train)
    modified_ytest = preprocessing.LabelEncoder()
    Resultant_y_test = modified_ytest.fit_transform(y_test)
    reg = LogisticRegression()
    reg.fit(x_train,training_scores_encoded)
    #print(reg.coef_)
    labels = reg.predict(x_test)
    labels
    output = labels.tolist()
    ipt2=[ipt,ipt]
    ipta=np.array(ipt2)
    #print("ipt2",ipt2)
    #print("ipta",ipta)

    #print(ipta)
    out=reg.predict(ipta)
    #print("out",out[0])
    ret=out[0]
    return(ret)
```

figure 3.12: Python code snippet for integration with LabVIEW.

To determine the healthiness of a plant we gave the values of table 3.1 as input for the predefined RGB values and stored into a Excel file. When we run the python code it calls the .csv table and check for the RGB values with the detected RGB values on the leaf. We created a block diagram in LabVIEW using the built-in python palette as described below [8], and called the python code to run within as shown in the figure 3.8. This block diagram contains three VI's as describe below,



This Open python session is used to configuring a multiple Python Nodes to run in specific versions of python like 2.7, 3.6 This block opens a python script saved in a specific directory and connects to the LabVIEW.[5]



Python Node VI is expandable and it shows the data types for the wired inputs and outputs values provided in the python code snippet. We can also configure the python node to specify the python session, module path and even function names.[5]

The Close Python Session is generally used to close all the opened python sessions previously by the user.[5]

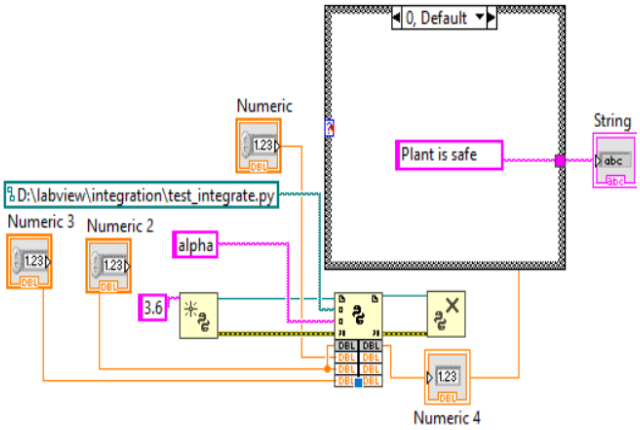
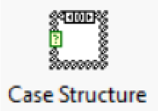
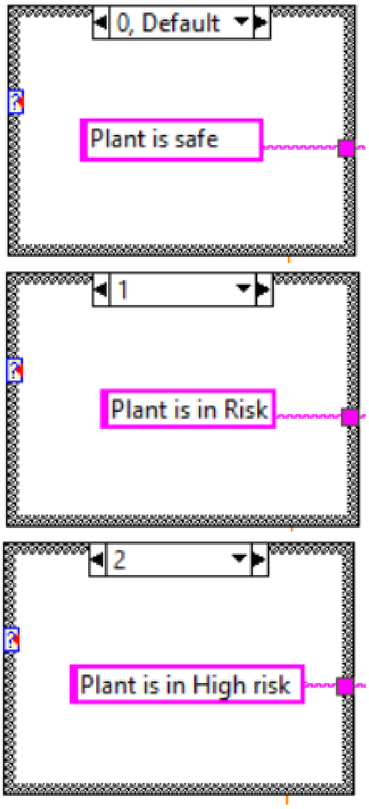


figure 3.11: Block diagram for Python integration with Lab VIEW.

In the above block diagram, we provide three input values which is pre-defined RGB values from the table 3.1 as Numeric1, Numeric2, Numeric3. Now the Open Python Session opens the python script to execute. This script takes the three input RGB values which we got after the segmentation of the image and matches with the table and provides a output value which says that the healthiness of the leaf. Python node manages all the inputs and outputs and the file path and displays the output. At the output section we placed which displays a string saying the plant is safe or not safe as shown in figure 3.9.[5]

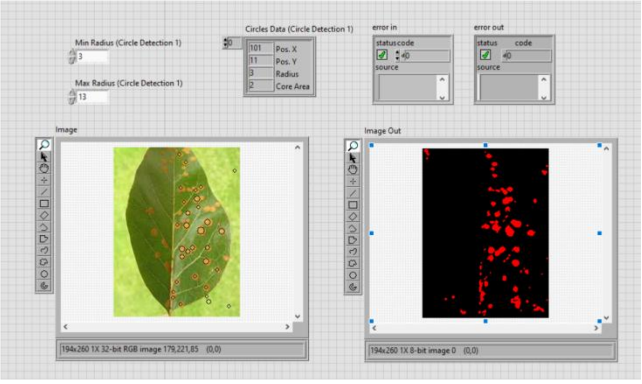




IV. RESULTS AND DISCUSSIONS

We designed an algorithm in LabVIEW using IMAQ VISION which detects the damaged plants and leaves by capturing them with a camera. This algorithm is based on the image processing techniques like image segmentation, color plane extraction feature extraction.

First the captured images are processed and an overlay is created on the damaged spot present on the leaf as shown in figure 4.1


Min Radius (Circle Detection 1) 3
Max Radius (Circle Detection 1) 13
Circles Data (Circle Detection 1) 0

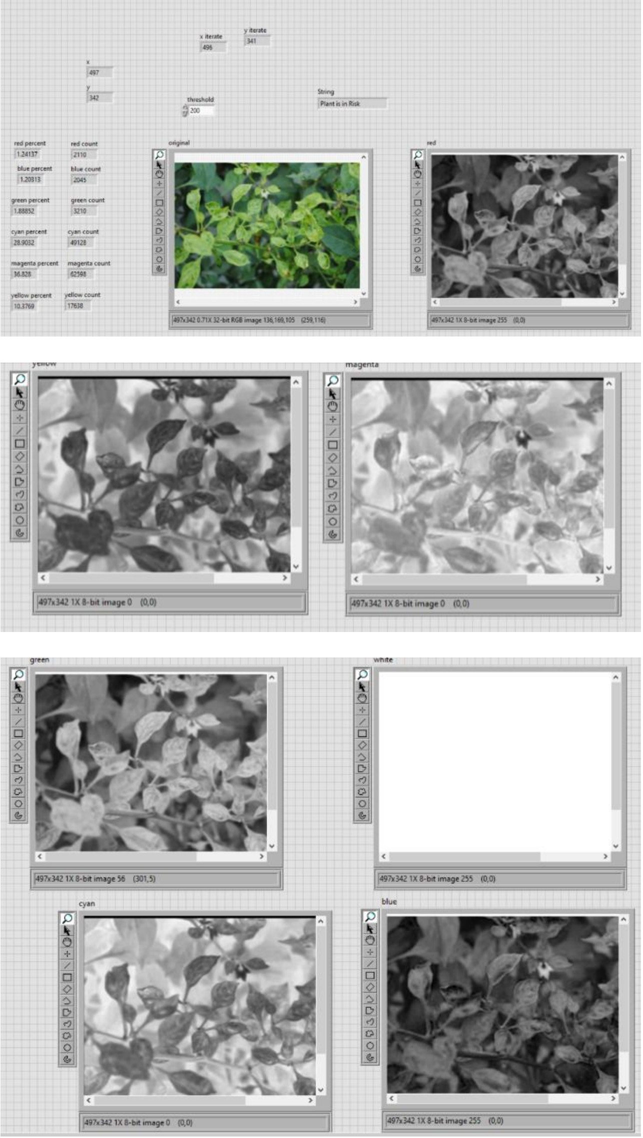
101 Pos. X
11 Pos. Y
3 Radius
2 Core Area

error in: status ✓ code 0 source
error out: status ✓ code 0 source
Image: 194x260 1X 32-bit RGB image 179,221,85 (0,0)
Image Out: 194x260 1X 8-bit image 0 (0,0)

figure 4.1: Image feature detection.

In the above front panel at the image output screen we can see the red spots which refers to the damaged area on the plant. The data of the damaged area including with area, radius, etc. will be stored.

In the second step as said in the reference [2], we used a Linear regression algorithm which is used to classify the given images into 3 stages based on CMY color percentages and the RGB and CMYK image outputs are shown below.



x iterate 496

y iterate 341

x 497

y 342

threshold 200

String: Plant is in Risk

red percent 1.24137, red count 2110

blue percent 1.20313, blue count 2045

green percent 1.88852, green count 3210

cyan percent 28.9032, cyan count 49128

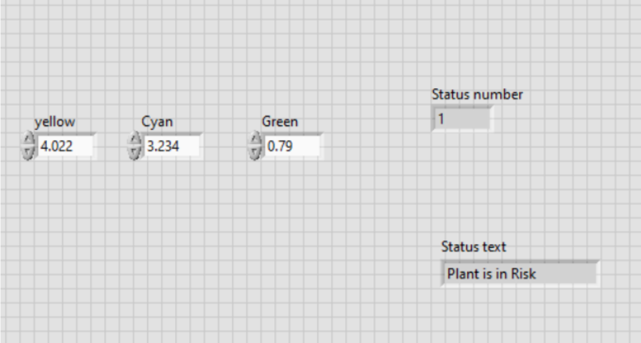
magenta percent 36.828, magenta count 62598

yellow percent 10.3769, yellow count 17638

original
497x342 0.71X 32-bit RGB image 136,169,105 (259,116)
red
497x342 1X 8-bit image 255 (0,0)
yellow
497x342 1X 8-bit image 0 (0,0)
magenta
497x342 1X 8-bit image 0 (0,0)
green
497x342 1X 8-bit image 56 (301,5)
white
497x342 1X 8-bit image 255 (0,0)
cyan
497x342 1X 8-bit image 0 (0,0)
blue
497x342 1X 8-bit image 255 (0,0)

Figure 4.2: Color plane extraction and the percentage of RGB and CMY present in the leaves

Now as we got the RGB values of the plants, we can find the health of the plants using these RGB values by comparing it with the table using the LabVIEW block diagram as specified in figure3.8 which integrates the python code. The RGB values will be given as 3 inputs and the output values determines the health of the plant as shown in figure 4.3.



yellow 4.022

Cyan 3.234

Green 0.79

Status number 1

Status text Plant is in Risk

Figure 4.3: Values to find the healthiness of plant

We took some leaves to test with, and stored the CMY values of those samples leaves into a table as shown in table 4.1. The linear regression algorithm says that the healthiness of the leaves. We got some healthy leaves and some are highly risky as provided in the table 4.1.

Table 4.1: Sample images output values for healthiness of the leaf

S.no	Yellow	Cyan	Green	Status
1	10.3769	28.9032	1.8852	Risk
2	19.9717	27.4708	0.0437	High Risk
3	2.67669	0	22.6199	Risk
4	12.9582	12.6234	40.311	High Risk
5	5.92513	0.52532	41.1212	Risk
6	30.4143	48.9498	6.65077	High Risk
7	17.4919	20.0704	11.9315	High Risk
8	3.18857	2.49722	24.4259	Risk
9	1.4515	5.52814	7.48263	Risk
10	37.3741	38.5408	1.43253	High Risk

V. CONCLUSION

We have worked on two different algorithms which classify based on the colour profile and the patterns present on the leaves. The spots and the colour profile of the image have been extracted by our analysis. The results of the algorithm using colour profiles are tabulated as shown in Table 4.1. The algorithm working with patterns on the leaf was used to detect diseases like mildew, blight and pest infestation. These profiles can be further studied and classified using different algorithms as in [4] to be precisely classified and could be provided with a precaution or a cure. The algorithm could further be developed into an app reaching large number of crowds.

Also, with the loss of dexterity in the coming years and with the non-availability of labour we could equip drones with these algorithms to increase the yield of the crop by effective identification, also the treatment for the diseases can be done without human intervention. In this way we can reduce the man power and can increase efficiency. For now, our research has been concentrated only on the leaves of the plants, this can be further extended to other parts of the plant too. Which may help in quicker diagnosis and suggesting quicker and more accurate solutions to the plant.

REFERENCES:

1. Malti K. Singh, Subrat Chetia2, "Detection and Classification of Plant Leaf Diseases in Image Processing using MATLAB", International Journal of Life Sciences Research, Vol. 5, Issue 4, pp: (120-124), Month: October - December 2017.
2. Z. Husin, A.Y.M. Shakaff, A.H.A. Aziz, R.S.M. Farook, "Plant Chili Disease Detection using the RGB Color Model",
3. https://www.google.com/search?q=diseased+leaves&rlz=1C1CHBF_enIN883IN883&oq=diseased+leaves&aqs=chrome..69i57j0l7.9136j1j7&sourceid=chrome&ie=UTF-8

4. Sandesh Raut, Kartik Ingole, "Review On Leaf Disease Detection Using Image Processing Techniques", International Research Journal of Engineering and Technology, Volume: 04 Issue: 04 | Apr -2017.
5. https://www.ni.com/pdf/manuals/320999c.pdf
6. https://www.ni.com/pdf/manuals/371077a.pdf
7. https://www.geeksforgeeks.org/linear-regression-python-implementation/
8. https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z0000019UFmSAM&l=en-IN

AUTHOR'S PROFILE:

Anusha Nellutla was born in 1986, at present is working as Assistant Professor in the department of ECE, Institute of Aeronautical Engineering, Hyderabad, Telangana, India. She received B. Tech, degree in Electronics and Communication Engineering from Jawaharlal Nehru Technological University, Hyderabad, Telangana, India, M.Tech., (Embedded Systems) from J.N.T. University, Hyderabad, Telangana, India. Her research interests include Image processing and Embedded Systems.

Gnana Sai Ganesh Chittajallu was born in 1999, at present pursuing his final year of Bachelor of Technology in Electronics and Communication Engineering from Institute of Aeronautical Engineering. His research interests include Image Processing and Computer Vision.

Shaik Feroz was born in 1998, at present pursuing his final year of Bachelor of Technology in Electronics and Communication Engineering from Institute of Aeronautical Engineering. His research interests include Image Processing and Computer Vision.