A Progress Report

on

Quantification of infection patterns on tomato leaves due to various pathogens using image processing and machine learning techniques.

carried out as part of the course CS1634 Submitted by

Sarthak Sharma
189302091

&

Aahan Singh Charak
189301024

VI-CSE

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

In

Computer Science & Engineering

MANIPAL UNIVERSITY
JAIPUR

INSPIRED BY LIFE

Binha
14/06/21

Department of Computer Science & Engineering,
School of Computing and IT,
Manipal University Jaipur,
June 2021

Abstract

This project aims at quantification of infection patterns on tomato leaves due to various pathogens using image processing and machine learning techniques.Raters are prone to tiredness , which might eventually lead to inaccurate measurements.A lot of resources are required to constantly train the raters.They often require some reference to quantify disease severity e.g Standard Area Diagram. Raters can’t cover a large area and as such many plants are left with no inspection at all. Some plant diseases show symptoms after a long time. Till the time we are able to see the symptoms, most of the harm is done.So, the best alternative to visual estimation is using images to carry out the quantification. Images can be analysed in visual or non-visual spectrum. Visual spectrum is that part of the light spectrum in which humans can see objects. Other techniques include capturing images in the hyperspectral or multispectral forms. But it requires sophisticated sensor technology which is currently out of our scope and reach. So we will work with images captured using a camera in the visual range. By using image processing techniques we will segment out the diseased portion of the tomato-leaves and quantify the disease severity in them either using the nominal or percentage scale. We will mainly be using ROI segmentation and binary thresholding to achieve this. After thresholding quantification will be carried out, which involves calculating the fraction of the diseased pixels to the healthy pixels.This fraction can then be converted to a percentage value, which can be used to estimate the amount of disease present on the leaf. The real challenge in this research is the availability of accurate measurements of the diseased pixels in the image which we can refer to in order to validate our results. Ground truth validation is thus a challenge for us. Though our results might be correct, validation is still needed to confirm them.

Table of contents Page Number

1. Introduction

Agriculture plays a very important role in our daily lives. From providing food to everyone, livelihood to the farmers and reducing the pollution levels , agriculture plays an important role in the survival of human beings.

Plants play a very vital role in nature. They are one of the forces keeping global-warming away from us. But plants aren't resistant to diseases. So, it becomes necessary for us to study the diseases occuring in plants, in order to develop measures to prevent and cure them. Let's take a look at the chart given below, showing the total area of tomato harvests vs the year of harvests in India.

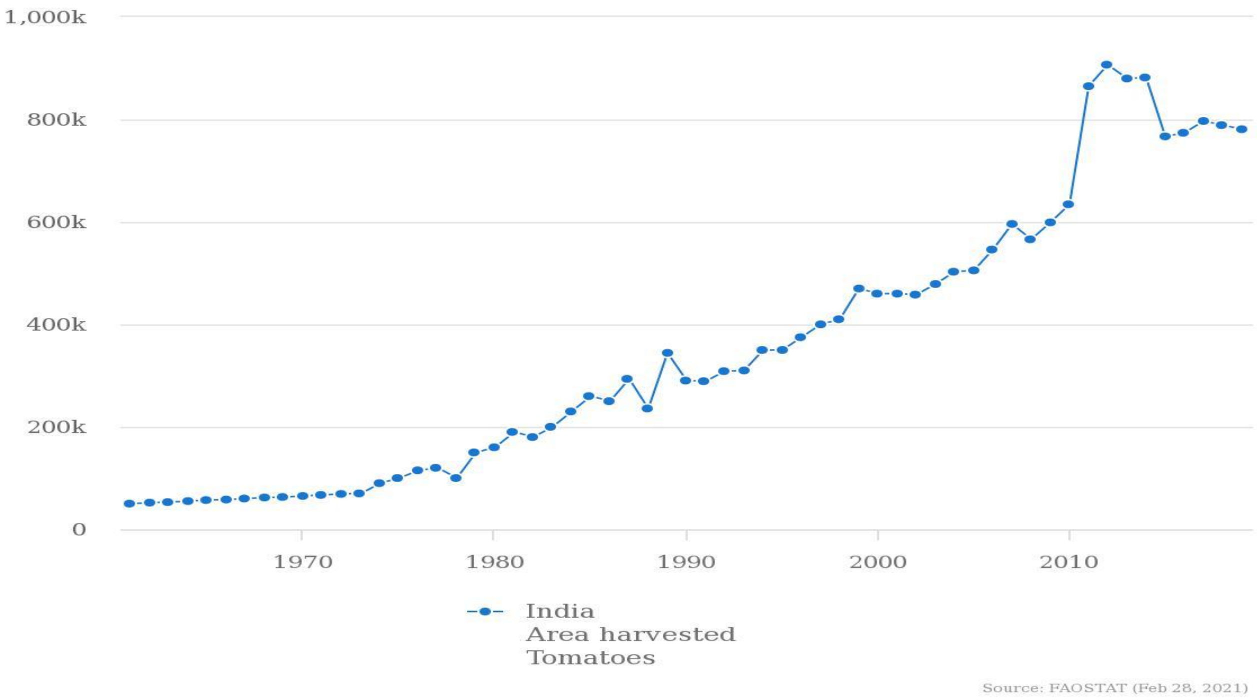


Fig 1. Graph showing area of harvested tomato vs the respective year of harvesting in India.

From the above given trend we can clearly see that production takes a dip starting from around late 2014. One of the main factors behind this is diseases happening in plants, but if we are able to correctly identify them and calculate the severity of infection, we will surely be able to boost productivity. Being aware of the infection patterns and severity of infection well in advance can help to reduce the risk of disease spreading over a large area of crops. Diseases in plants are quite ubiquitous, so identifying them well in advance will be a great help to us.

Visual estimation is one of the ways to achieve this but requires a lot of hard work and human resources which isn't cost efficient. Visual analysis of images for quantification can help estimate the severity of infection over a large area without much effort and is also cheap. Moreover, the market is filled with many softwares which make our jobs easier as we don't have to develop our own algorithms to achieve this task and they also help us save a lot of precious time.

1.1 Motivation

One of the main motivating factors behind carrying out this research is that there hasn't been extensive research on the quantification aspect of plant leaf diseases, especially the tomato ones. Most software available in the market is for detecting and classifying leaf diseases. Though this might be helpful in the agriculture department, knowing how much the plant is diseased will be even more helpful to the farmers, as they can take appropriate actions accordingly, as severe diseases require more attention as opposed to the mild ones.

In most of the cases, estimation about the severity of plant diseases is done by humans visually. These individuals who excel in the field of visual estimation of plant disease severity are called Raters. If they are trained effectively, they can make accurate estimations about the plant disease severity. But there can be a lot of disadvantages in quantifying plant diseases visually. According to Bock et al.(2013) there are many disadvantages of visual estimation. Some of them are discussed below :

Raters are prone to tiredness , which might eventually lead to inaccurate measurements.
A lot of resources are required to constantly train the raters.
They often require some reference to quantify disease severity e.g Standard Area Diagram.
Raters can't cover a large area and as such many plants are left with no inspection at all.
Some plant diseases show symptoms after a long time. Till the time we are able to see the symptoms, most of the harm is done.

So, the best alternative to visual estimation is using images to carry out the quantification. Images can be analysed in visual or non-visual spectrum. Visual spectrum is that part of the light spectrum in which humans can see objects. Other techniques include capturing images in the hyperspectral or multispectral forms. But it requires sophisticated sensor technology which is currently out of our scope and reach. So we will work with images captured using a camera in the visual range.

By using image processing techniques we will segment out the diseased portion of the tomato-leaves and quantify the disease severity in them, either using the nominal or the percentage scale. Example of a tomato leaf image in the visible spectrum is given below.

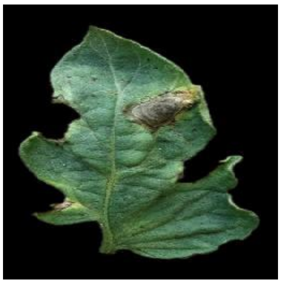


Fig 2.. Image showing bacterial spot on an infected tomato leaf
Image can be found at Plant Village image dataset at github.

2 Literature Review

2.1 Quantification:

Quantification is the process of finding out how much portion of a plant is diseased. It is also called the severity of disease in a plant. It basically tells us about the percentage of the diseased tissue in a plant.

In this project we are measuring the severity of infection on tomato leaves. There are many techniques to measure this but we are using visual spectrum image analysis techniques. Image will be in the form of pixels (rows and columns) and each pixel will contain a specific portion of the leaf. The diseased pixels will be found using image analysis. We will call them bad pixels. We will then calculate the percentage of bad pixels on the leaf segment. This percentage scale can now be converted to nominal in order to find the severity of infection on the leaf.

According to Barbedo (2013), the severity of infection on a leaf can be easily identified by the color pattern on the leaf.Most of the quantification algorithms require segmentation step.Every quantification analysis that we do by ourselves is not 100 percent accurate. All measurements diverge from the "true value" or "ground truth". The measurements that we make are not absolute and are bound to differ from the "ground truth". It's just an estimate that we make.

2.2 Segmentation:

Segmentation is one of the most important steps for disease quantification. Segmentation is an image processing concept which involves dividing a particular image into regions of interest. We can then study those regions of interest without worrying about unwanted pixels interfering with our measurements.

Thresholding is one of the most commonly used techniques used to segment the diseased portion of a leaf from the healthy portion. According to Barbedo (2013), the most simple thresholding requires separating the diseased portion of the leaf and then applying a correction factor. The correct- ion factor makes sure that the healthy pixels which were counted as diseased by mistake are counted out of the final value. Colored CCD(charged couple device) showed better results than black CCD in coffee leaves.

Quantification via image processing (using segmentation) is found to be a better and efficient choice as compared to visual estimation.In two-stage thresholding we first segment the image from the back- ground. And in the second stage we segment diseased pixels from healthy. Then we calculate the portion of bad pixels divided by the whole leaf area pixels.

2.3 Visible Spectrum Image Analysis:

There exist many color spaces which can be used to analyze images. HSV,RGB ,La*b* are some of the color spaces which can be used to study and quantify leaf diseases in plants.

According to Barbedo(2013), visible spectrum image analysis has its own demerits. Some diseases are not visible to the naked eye, also some diseases begin to show symptoms at a much later stage, which is quite late to carry out analysis. Here we will deal with quantification of diseases using HSV and La*b* color spaces.

The good thing about visible spectrum image analysis is that it produces efficient results, if carried out under controlled conditions.

2.4 Datasets:

In order to carry out research properly datasets are a must.An algorithm can be made better and more useful by testing and evaluating results on a well defined collection of data (Data Set) that is compatible with our case study.Results after using the datasets will tell about how efficient our algorithm really is. Plant village data set is used which consists of 54303 healthy and unhealthy leaf images divided into 38 categories by species and disease.

2.5 Nominal Scale and Percentage Scale:

During experiments/research, results are recorded using predefined statistical scales. Nominal scale is one of those scales. Nominal data consists of non-numeric information. Nominal scale is used to record such kinds of data. Data like gender,hair-color etc.. are usually measured using the nominal scale.With the help of nominal scale we can not only classify different objects , we can also allot different numbers based on characteristics.For example we can allot numbers 1- 5 for disease severity , 1 indicating lowest concern and 5 indicating very severe disease .

Percentage scale on the other hand is simply the representation of diseased pixels in the form of percentage values ranging between 0 and 100.

2.6 Standard Area Diagrams:

Standard Area Diagram generally known as SAD is an important tool which can be used to visually estimate the severity of disease on a plant/leaf.

According to Bock et al. (2020) SAD'S are extensive tools used by raters to correctly quantify disease severity in plants.

The most typical SAD comprises five to eight black-and-white drawings of leaves, with severity increasing in a non-linear fashion. In the case of a two step SAD validation approach , linear regression is the preferred method.

2.7 Color Spaces:

RGB is mostly used for image capturing but is rarely used to segment images. HSV is used for image thresholding segmentation. In this we use a histogram of intensities which can be used to segment diseased portions from the leaf. RGB is not preferred as it just gives us the idea how human beings perceive the captured object, HSV on the other hand gives us an idea about the true color of the captured object, which at times is also called as pure color.

2.8 Outcomes

1. HSV and La*b* Color spaces are the most useful when it comes to studying the extensive features of the plants.

2. Nominal or percentage scale can be used to represent severity of disease in numerical or alphabetical form.

3. Though standard area diagrams are effective to visually rate the severity of disease in the plants they are pretty hard to come by.

4. Raters are most efficient in visually estimating severity of plant disease but are costly and require a large amount of capital to train and keep up to date.

5. Thresholding is one of the most commonly used segmentation techniques which can be used to efficiently segment the diseased portion of a leaf/plant from the healthy portion.

6. One of the simplest quantification techniques is to find the fraction of diseased pixels to the healthy pixels and then accordingly convert the value to nominal or percentage scales.

7. There are some cases in which visual spectrum analysis can't be used like in the case of hidden disease patterns. In such cases we use hyperspectral or multispectral image analysis techniques which are currently out of our scope.

8. Raters are prone to errors like lighting,illusions etc.. For such cases visible image spectrum techniques provide an alternative.

9. Shadows and darkness can cause errors while performing visual image spectrum image analysis.

10. Ground truth validation is also required to validate our results. Such sources of validation are very hard to come by and are mostly done by raters using standard area diagrams.

2.9 Problem Statement

Objective: Quantification of infection patterns on tomato leaves due to various pathogens using image processing and machine learning techniques.

Description:

Plant diseases affect agricultural related activities in all economies. It is necessary to identify plant disease at its earliest in order to avoid enormous amounts of damage. Visually estimating the severity of disease in leaves is easy but requires a lot of capital. Disease quantification using image processing is quite effective in getting the job done. Image processing techniques like segmentation,pixel quantification and color space analysis are productive tools which help us in estimating disease severity in plants. In this project we will mainly use these techniques on tomato leaves to quantify diseases on a vast dataset of images. Many tomato diseases like early-blight,late-blight,mites etc.. will be covered.We will use image processing using python and OpenCv to get this task done. This task can be achieved by calculating the bad pixels to good pixels ratio in the images. Bad pixel basically refers to the diseased portion of the leaf image and good pixel ratio refers to the healthy portion of the leaf.

2.10 Research Objectives

- To analyze the infection pattern on tomato leaves using various color spaces like HSV, La*b* etc..
- To segment the diseased portion of the leaves from the healthy portion using image processing and machine learning techniques.
- To quantify the disease portions using appropriate algorithms.
- Validating our results using ground truth validation.

3. Methodology and Framework

3.1 Flow Chart

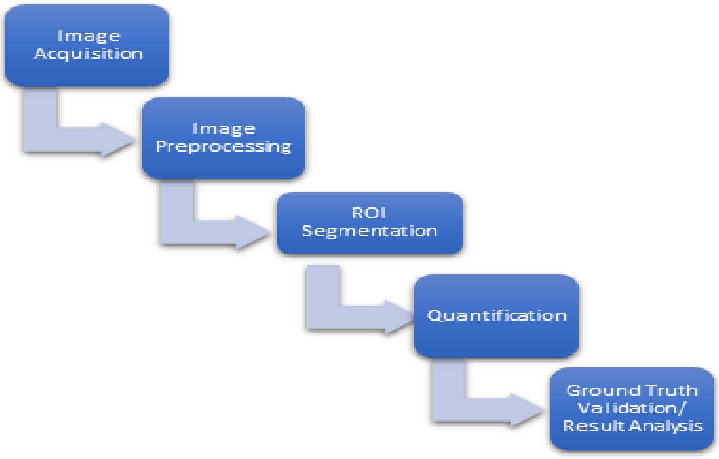
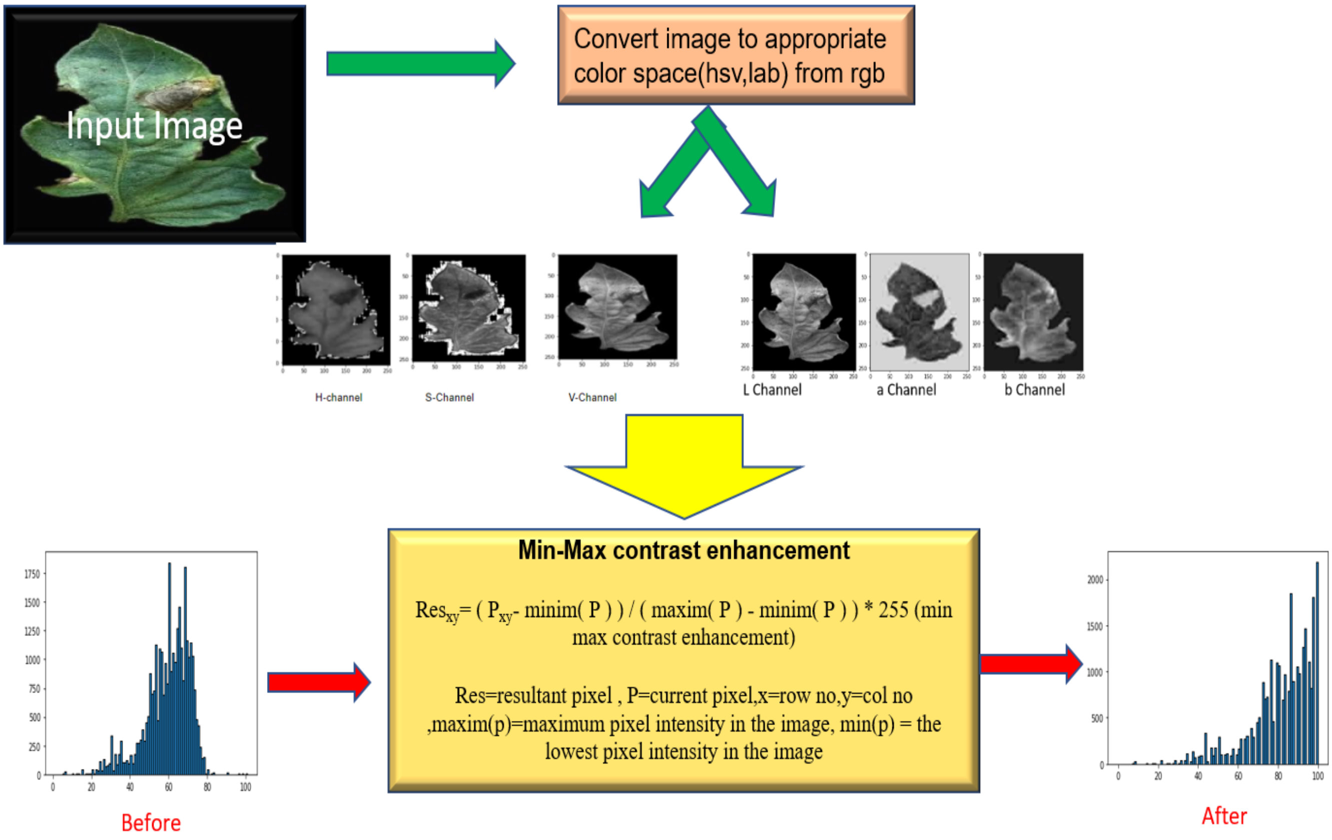


Fig 3 Flow Diagram For Methodology

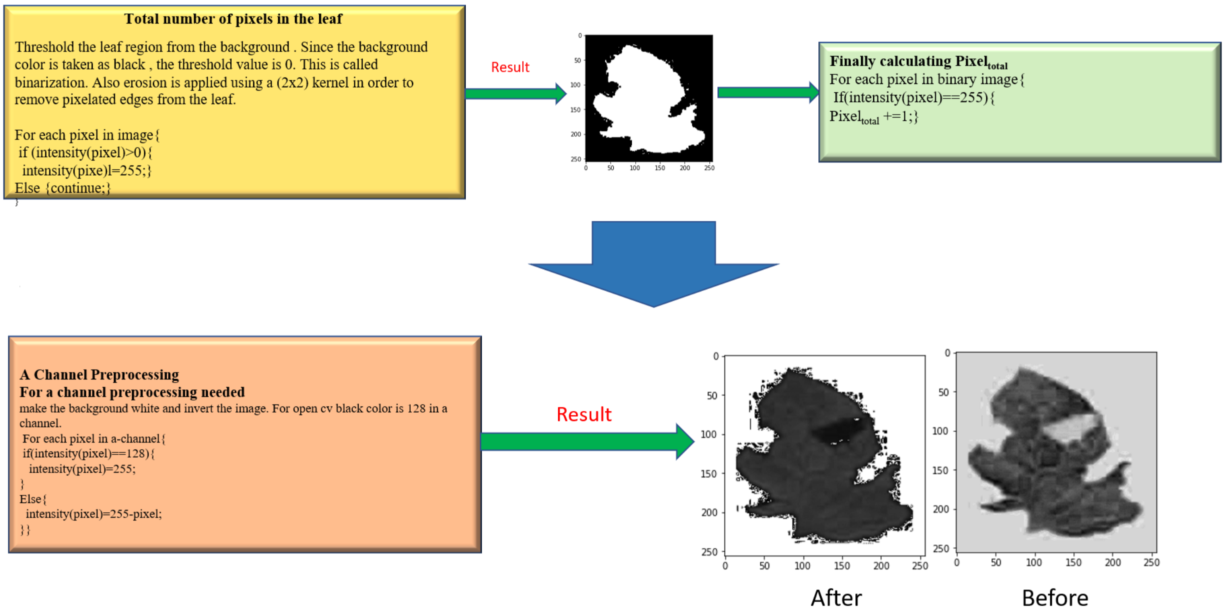


Min-Max contrast enhancement

Res_{xy}= (P_{xy}- minim(P)) / (maxim(P) - minim(P)) * 255 (min max contrast enhancement)

Res=resultant pixel , P=current pixel,x=row no,y=col no ,maxim(p)=maximum pixel intensity in the image, min(p) = the lowest pixel intensity in the image

Fig 4. Detailed Flow Diagram For Methodology 1 of 3



Total number of pixels in the leaf

Threshold the leaf region from the background . Since the background color is taken as black , the threshold value is 0. This is called binarization. Also erosion is applied using a (2x2) kernel in order to remove pixelated edges from the leaf.

```
For each pixel in image{
 if (intensity(pixel)>0){
  intensity(pixe)l=255;}
Else {continue;}
}
```

Result

Finally calculating Pixel$_{total}$

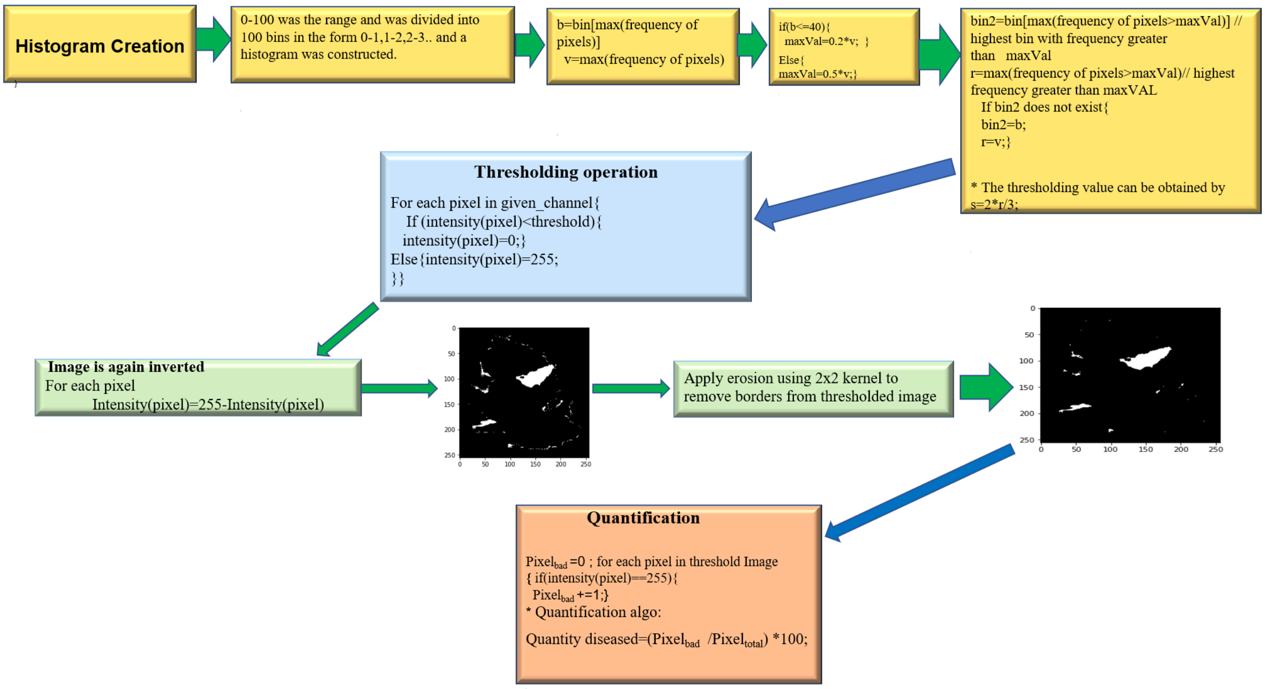
For each pixel in binary image{
 If(intensity(pixel)==255){
Pixel$_{total}$ +=1;}

A Channel Preprocessing
For a channel preprocessing needed
make the background white and invert the image. For open cv black color is 128 in a channel.

```
 For each pixel in a-channel{
 if(intensity(pixel)==128){
   intensity(pixel)=255;
}
Else{
  intensity(pixel)=255-pixel;
}}
```

Result

Fig 5. Detailed Flow Diagram For Methodology 2 of 3



Histogram Creation

0-100 was the range and was divided into 100 bins in the form 0-1,1-2,2-3.. and a histogram was constructed.

```
b=bin[max(frequency of pixels)]
  v=max(frequency of pixels)
```

```
if(b<=40){
  maxVal=0.2*v;  }
Else{
maxVal=0.5*v;}
```

```
bin2=bin[max(frequency of pixels>maxVal)] // highest bin with frequency greater than   maxVal
r=max(frequency of pixels>maxVal)// highest frequency greater than maxVAL
  If bin2 does not exist{
  bin2=b;
  r=v;}
```

* The thresholding value can be obtained by s=2*r/3;

Thresholding operation

```
For each pixel in given_channel{
    If (intensity(pixel)<threshold){
   intensity(pixel)=0;}
Else{intensity(pixel)=255;
}}
```

Image is again inverted
For each pixel
            Intensity(pixel)=255-Intensity(pixel)

Apply erosion using 2x2 kernel to remove borders from thresholded image

Quantification

Pixel$_{bad}$ =0 ; for each pixel in threshold Image
{ if(intensity(pixel)==255){
  Pixel$_{bad}$ +=1;}
* Quantification algo:

Quantity diseased=(Pixel$_{bad}$ /Pixel$_{total}$) *100;

Fig 6. Detailed Flow Diagram For Methodology 3 of 3

3.2 Dataset Acquisition

Dataset acquisition is the first and foremost step in order to carry out research work. Plant village dataset from github was used to collect images for various kinds of tomato leaf diseases. Two different kinds of tomato blight diseases like early blight, late blight were considered for our research. 50 different sample images for each disease were considered as of now.

3.3 Color Space Analysis

As our next step, we analysed the image samples using two different color space models, HSV and La*b* .

- **HSV Color Space Analysis**

  HSV stands for hue,saturation and value. Hue is the angle, the color makes with the color wheel (red is 0 degrees). Saturation basically tells the amount of greyness in a color. As we keep on mixing white light with the color, the greyness level keeps on increasing and eventually the color turns to white.It's value ranges from 0 to 100 but since we used opencv to carry out our research work, the saturation value had to be converted to a range between 0 and 255. Value is the amount of brightness in an image. 100 value means 100 percent brightness, 0 value means 0 brightness or completely black color. Again for opencv the value had to be converted into a range of 0 to 255.

  Next, we visualized the images in h,s and v channels separately, using grayscale. H channel gave us the best visualization of the diseased portion so it was considered to segment the image.

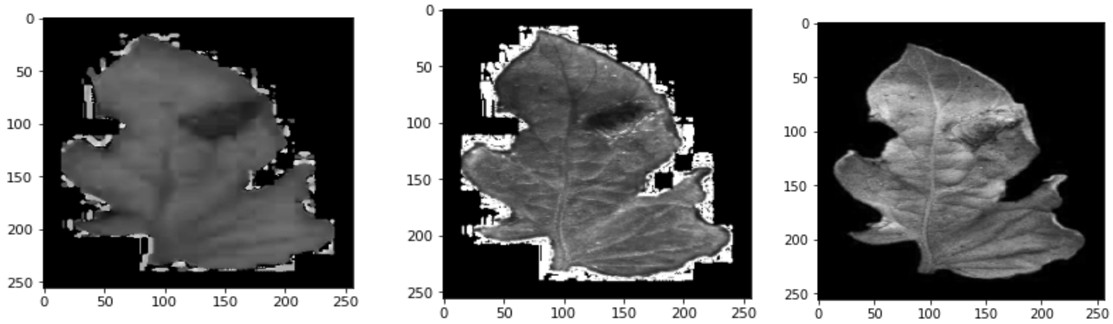


Fig7. Tomato Early Blight Image Visualized in H,S and V channels respectively.

- **La*b* Color Space Analysis**

  For La*b* color space, we chose the L channel for segmentation.

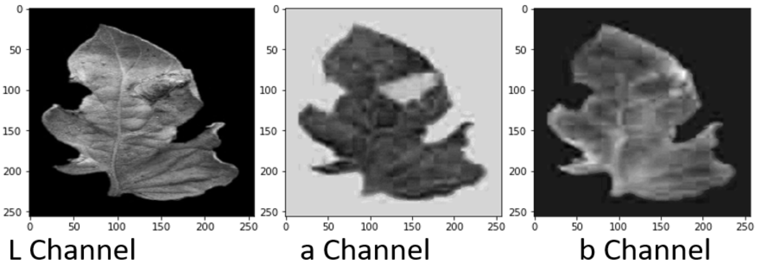


Fig 8. Tomato Early Blight Image Visualized in L,a and b channels respectively.

3.4 Contrast Enhancement

For contrast enhancement we use min-max contrast enhancement. The formula is stated as:

$$\mathbf{Res_{xy} = (P_{xy} - minim(P)) / (maxim(P) - minim(P)) * 255}$$

Res=Resultant Pixel Value
P_{xy}= Current Pixel on which operation is being performed
maxim(P)= Maximum Pixel Intensity
minim(P)=Minimum Pixel Intensity

x and y are the indices of the current pixel and 255 is a factor multiplied to convert image's scale to (0-255)

3.5 ROI Segmentation

For ROI Segmentation we used H channel and A channel for HSV and La*b* respectively. We used the same algorithm as described by Barbedo (2016).

First we constructed an intensity histogram for both the channels. 100 bins were taken from 1 to 100 .We restricted the range upto 100 because after that most leaves didn't have any significant amount of pixel intensities. Histogram construction was important because the healthier plant tissues tend to generate a peak at the right side of the histogram. The diseased tissues tend to generate peaks towards the left side of the histogram.

In grayscale images healthier plant tissues are represented as light shades of gray whilst on the other hand diseased tissues tend to be on the blacker side. First, peak intensity (global maxima) was identified as the bin corresponding to it was noted down as let's say **BIN.**
The intensity value of the maxima was noted down as **maxInt.**

```
if ( BIN <= 40 )
        REF = BIN value where Intensity > 0.2 * maxInt and Intensity ! = maxInt
else
```

REF = BIN value where Intensity > 0.5 * maxInt and Intensity ! = maxInt

if no bin found for reference then REF = BIN

If a peak is found at lower ranges, then that means that the plant is severely diseased. Also,if the peak is found at higher ranges, it means that the leaf is healthy as green and healthy pixels have the highest number in the image.

The best bin which separates diseased and healthy pixels can be given by

TBIN = 2* REF / 3

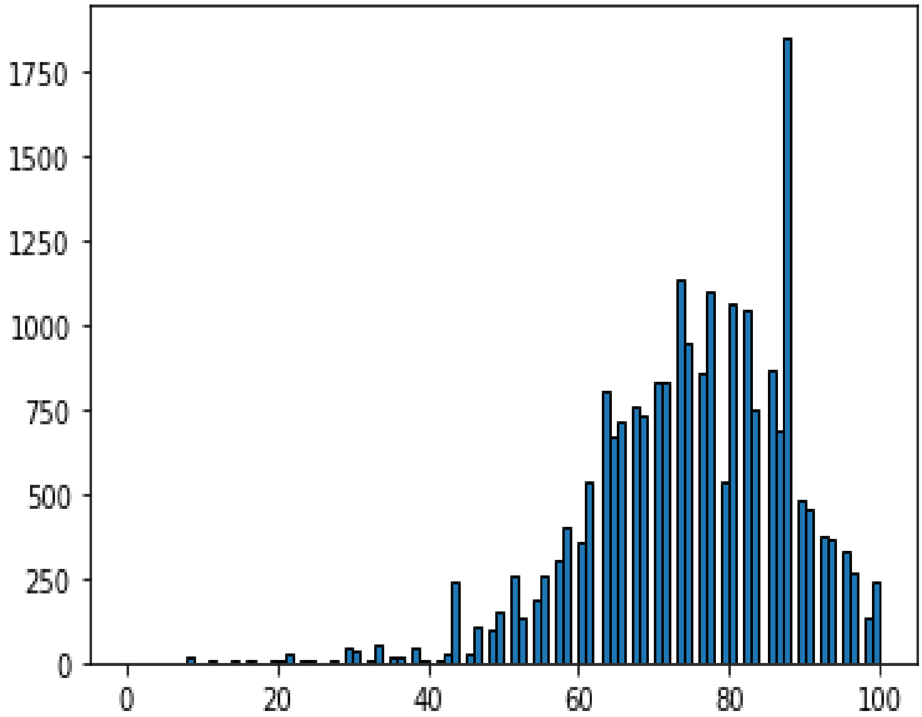


Fig 9. An Intensity Histogram Example

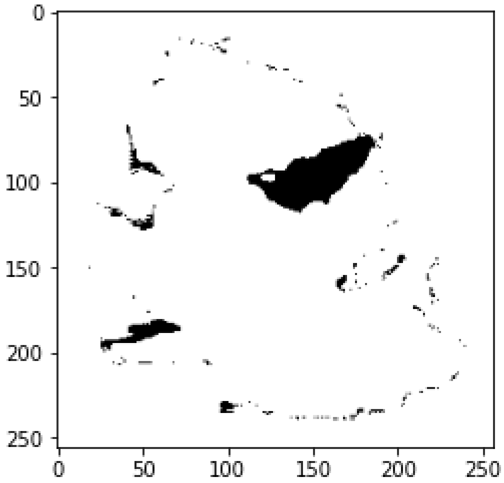


Fig 10. Segmented Image of the leaf Example

3.6 Quantification

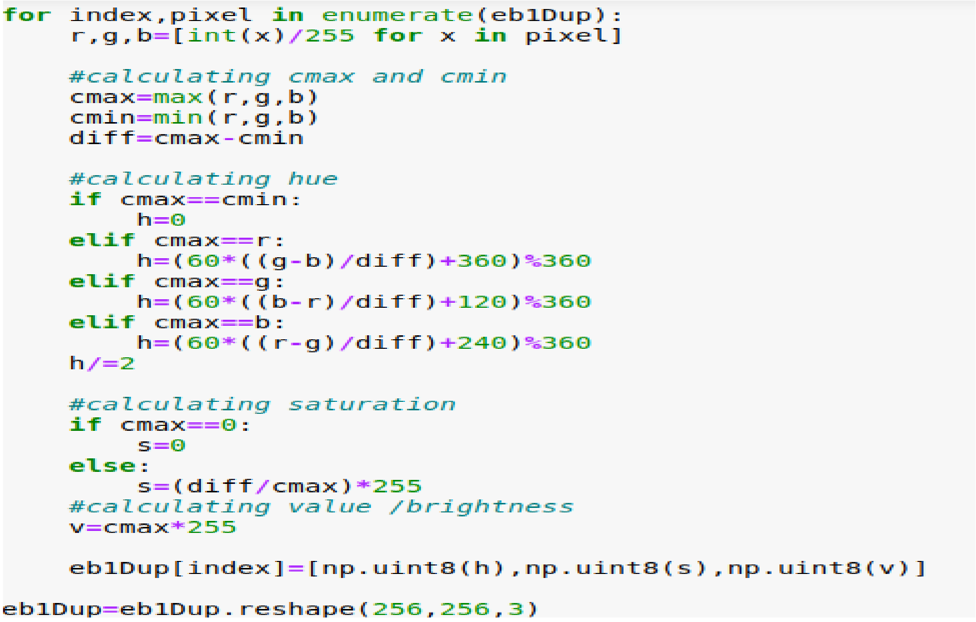
The quantification process was fairly easy. All we had to do was to calculate the total pixel area occupied by the leaf and the total pixel area occupied by the diseased pixels. Next the diseased pixels were divided by the healthy pixels to generate a percentage value. This process was repeated for different kinds of tomato leaf diseases and these values were noted down.

Severity of Disease (%) = N_{bad} / N_{total} *100

N_{bad} = Number of diseased pixels
N_{total} = Total number of pixels occupied by the leaf

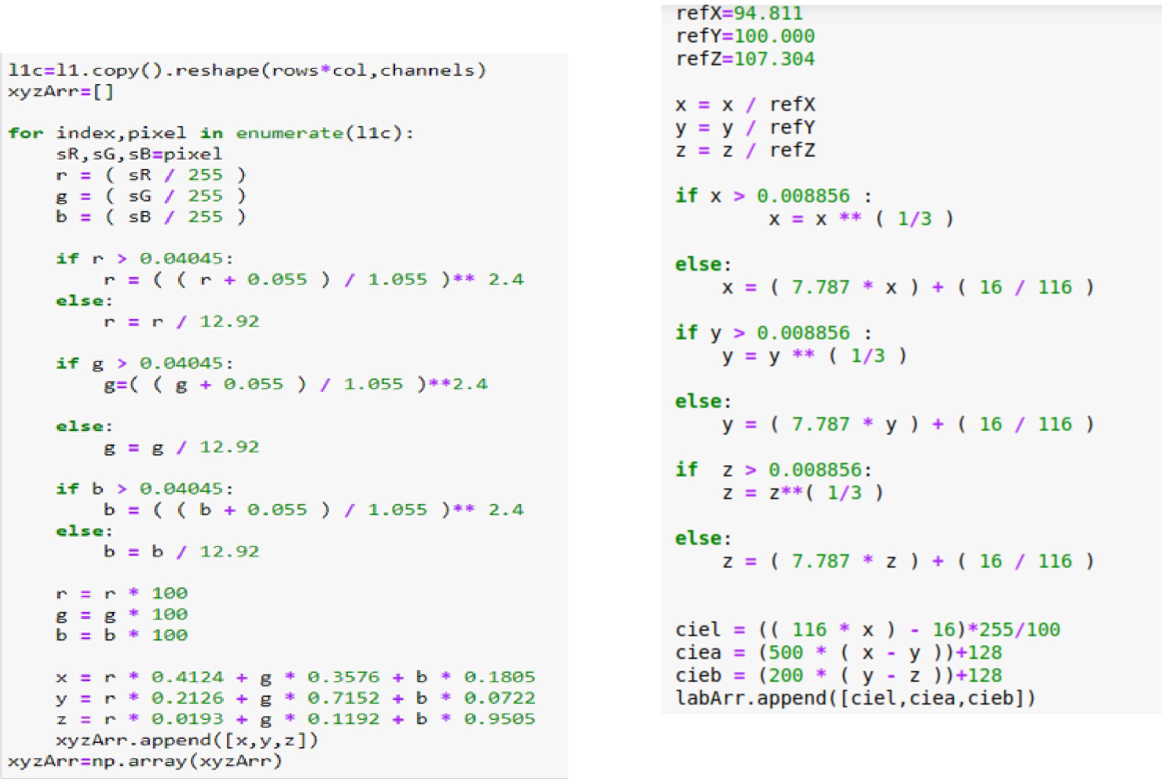
4. Work Done

4.1 Code Snippet



```
for index,pixel in enumerate(eb1Dup):
    r,g,b=[int(x)/255 for x in pixel]

    #calculating cmax and cmin
    cmax=max(r,g,b)
    cmin=min(r,g,b)
    diff=cmax-cmin

    #calculating hue
    if cmax==cmin:
        h=0
    elif cmax==r:
        h=(60*((g-b)/diff)+360)%360
    elif cmax==g:
        h=(60*((b-r)/diff)+120)%360
    elif cmax==b:
        h=(60*((r-g)/diff)+240)%360
    h/=2

    #calculating saturation
    if cmax==0:
        s=0
    else:
        s=(diff/cmax)*255
    #calculating value /brightness
    v=cmax*255

    eb1Dup[index]=[np.uint8(h),np.uint8(s),np.uint8(v)]

eb1Dup=eb1Dup.reshape(256,256,3)
```

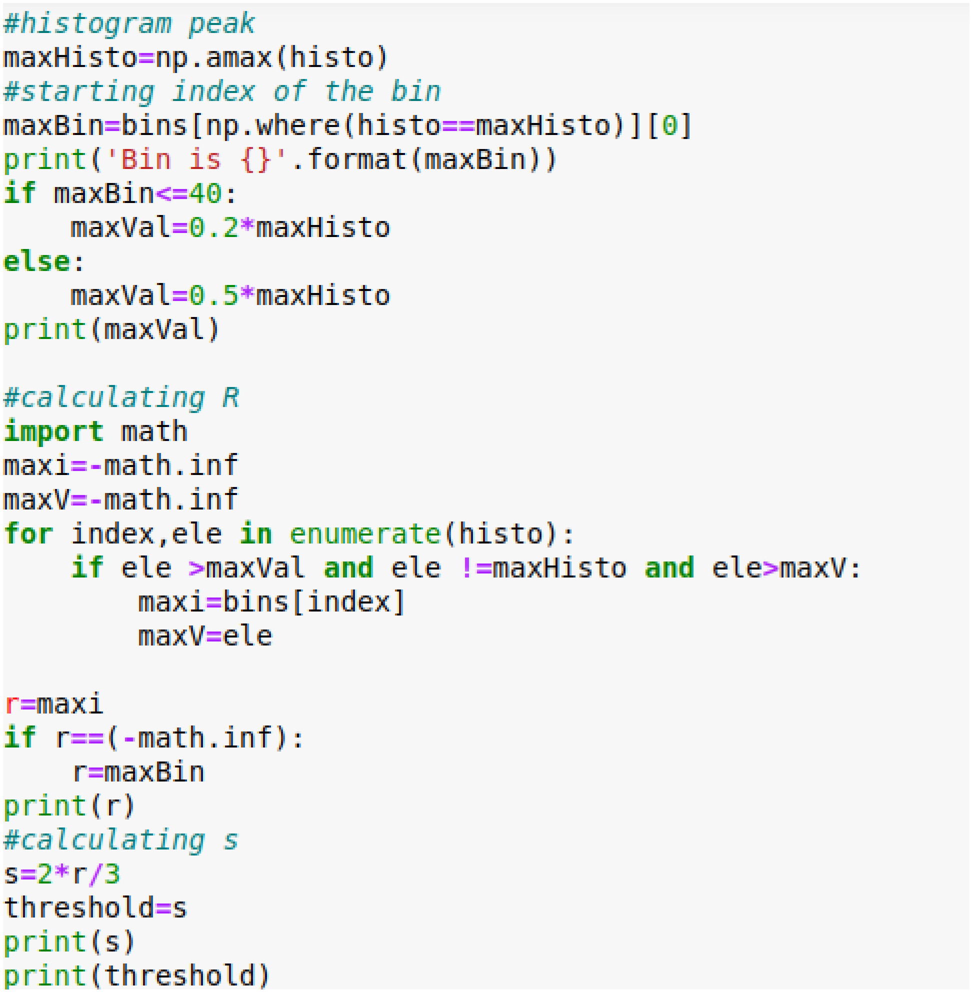
Fig 11. Algorithm to convert RGB to HSV



```
l1c=l1.copy().reshape(rows*col,channels)
xyzArr=[]

for index,pixel in enumerate(l1c):
    sR,sG,sB=pixel
    r = ( sR / 255 )
    g = ( sG / 255 )
    b = ( sB / 255 )

    if r > 0.04045:
        r = ( ( r + 0.055 ) / 1.055 )** 2.4
    else:
        r = r / 12.92

    if g > 0.04045:
        g=( ( g + 0.055 ) / 1.055 )**2.4

    else:
        g = g / 12.92

    if b > 0.04045:
        b = ( ( b + 0.055 ) / 1.055 )** 2.4
    else:
        b = b / 12.92

    r = r * 100
    g = g * 100
    b = b * 100

    x = r * 0.4124 + g * 0.3576 + b * 0.1805
    y = r * 0.2126 + g * 0.7152 + b * 0.0722
    z = r * 0.0193 + g * 0.1192 + b * 0.9505
    xyzArr.append([x,y,z])
xyzArr=np.array(xyzArr)
```

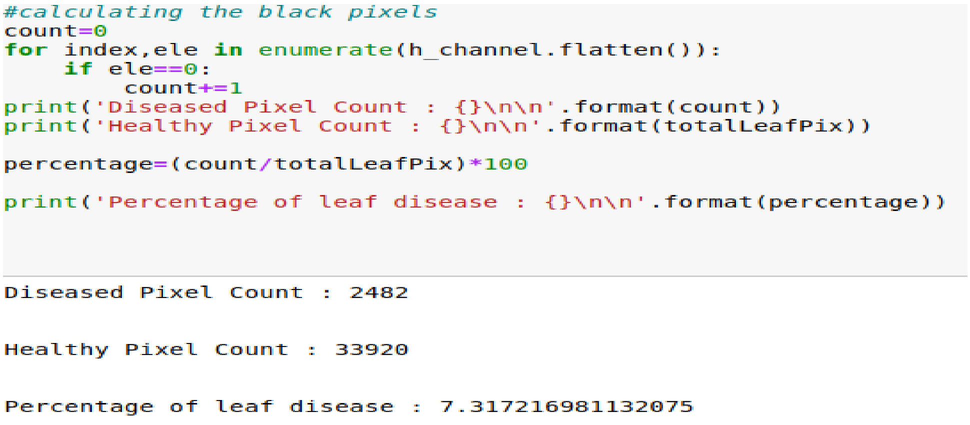
```
refX=94.811
refY=100.000
refZ=107.304

x = x / refX
y = y / refY
z = z / refZ

if x > 0.008856 :
        x = x ** ( 1/3 )

else:
    x = ( 7.787 * x ) + ( 16 / 116 )

if y > 0.008856 :
    y = y ** ( 1/3 )

else:
    y = ( 7.787 * y ) + ( 16 / 116 )

if  z > 0.008856:
    z = z**( 1/3 )

else:
    z = ( 7.787 * z ) + ( 16 / 116 )


ciel = (( 116 * x ) - 16)*255/100
ciea = (500 * ( x - y ))+128
cieb = (200 * ( y - z ))+128
labArr.append([ciel,ciea,cieb])
```

Fig. 12 Algorithm to convert RGB TO La*b* color space



```
#histogram peak
maxHisto=np.amax(histo)
#starting index of the bin
maxBin=bins[np.where(histo==maxHisto)][0]
print('Bin is {}'.format(maxBin))
if maxBin<=40:
    maxVal=0.2*maxHisto
else:
    maxVal=0.5*maxHisto
print(maxVal)

#calculating R
import math
maxi=-math.inf
maxV=-math.inf
for index,ele in enumerate(histo):
    if ele >maxVal and ele !=maxHisto and ele>maxV:
        maxi=bins[index]
        maxV=ele

r=maxi
if r==(-math.inf):
    r=maxBin
print(r)
#calculating s
s=2*r/3
threshold=s
print(s)
print(threshold)
```

Fig 13 Thresholding Algorithm



```
#calculating the black pixels
count=0
for index,ele in enumerate(h_channel.flatten()):
    if ele==0:
        count+=1
print('Diseased Pixel Count : {}\n\n'.format(count))
print('Healthy Pixel Count : {}\n\n'.format(totalLeafPix))

percentage=(count/totalLeafPix)*100

print('Percentage of leaf disease : {}\n\n'.format(percentage))
```

```
Diseased Pixel Count : 2482


Healthy Pixel Count : 33920


Percentage of leaf disease : 7.317216981132075
```

Fig14 Quantification Algorithm

4.2 Results and Discussion

The quantification process was carried out on 100 different sample images of tomato leaves with 50 images belonging to the early blight category and 50 images belonging to the late blight category respectively. Then the results of the h and a color channels were recorded in an excel sheet. This was done for both early and late blight sample images. The sample early blight images are shown in Fig. 18 and the sample late blight images are shown in Fig. 19 respectively.

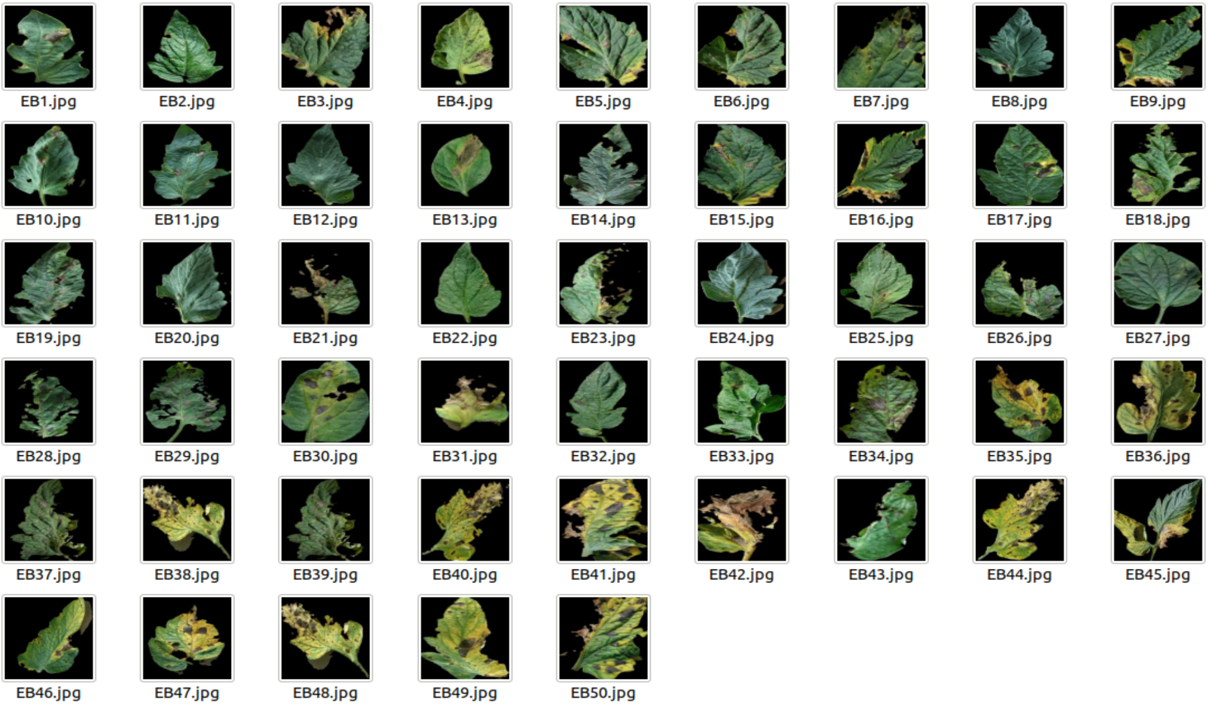


Fig 18. Early Blight Sample Images

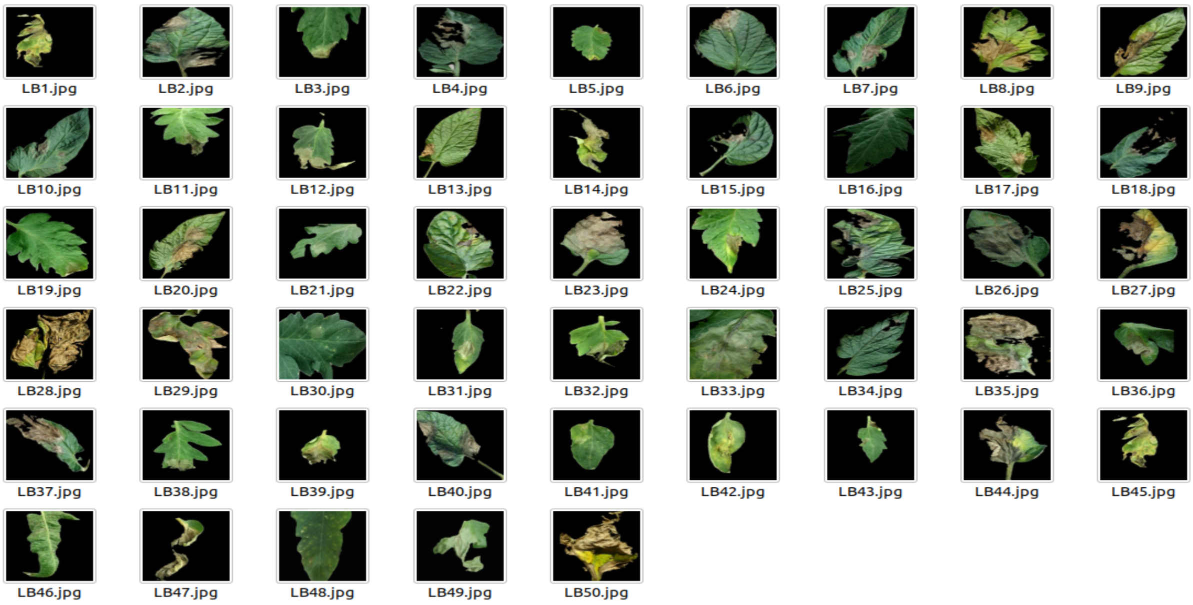


Fig 19. Late Blight Sample Images

Now, there was no ground truth validation source from where we could validate our quantification results, so we decided to validate our results by checking whether the diseased portions of the leaves were correctly segmented from the healthy portions. The segmentation results for early blight image samples are given in Fig 20.Similarly Fig 21. shows segmentation results for late blight images.

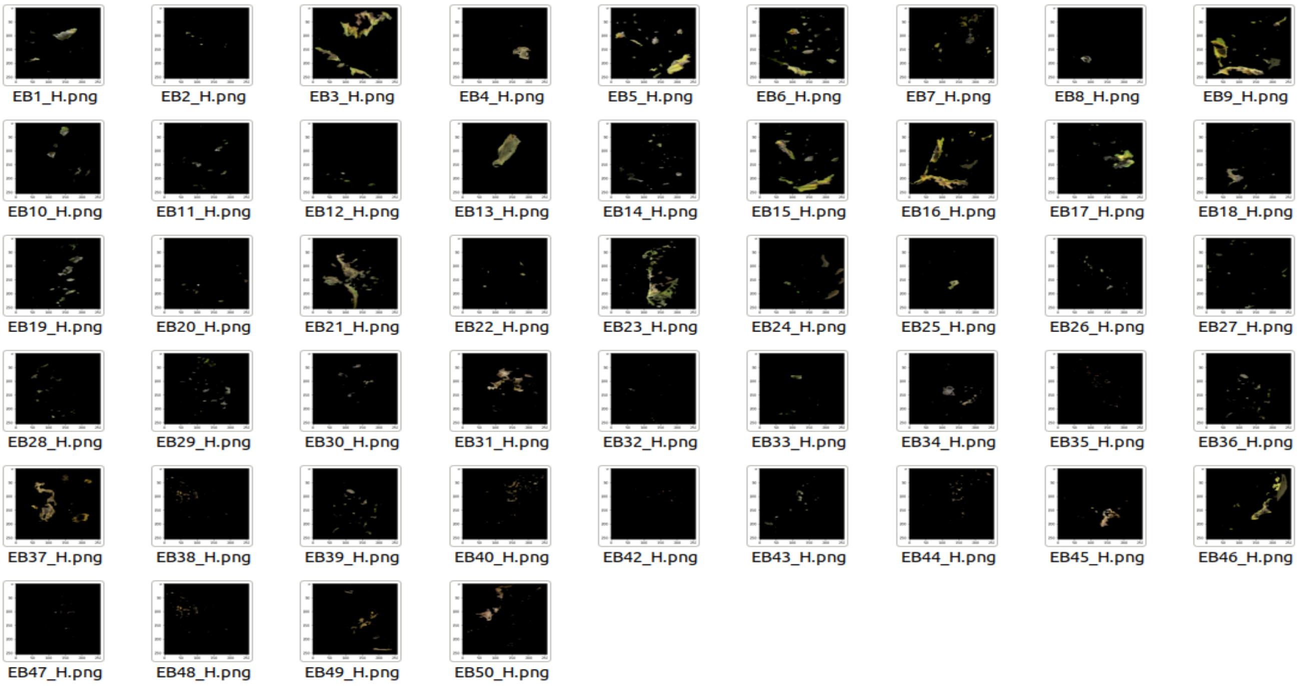


Fig 20 (a) Segmentation results for early blight in h channel

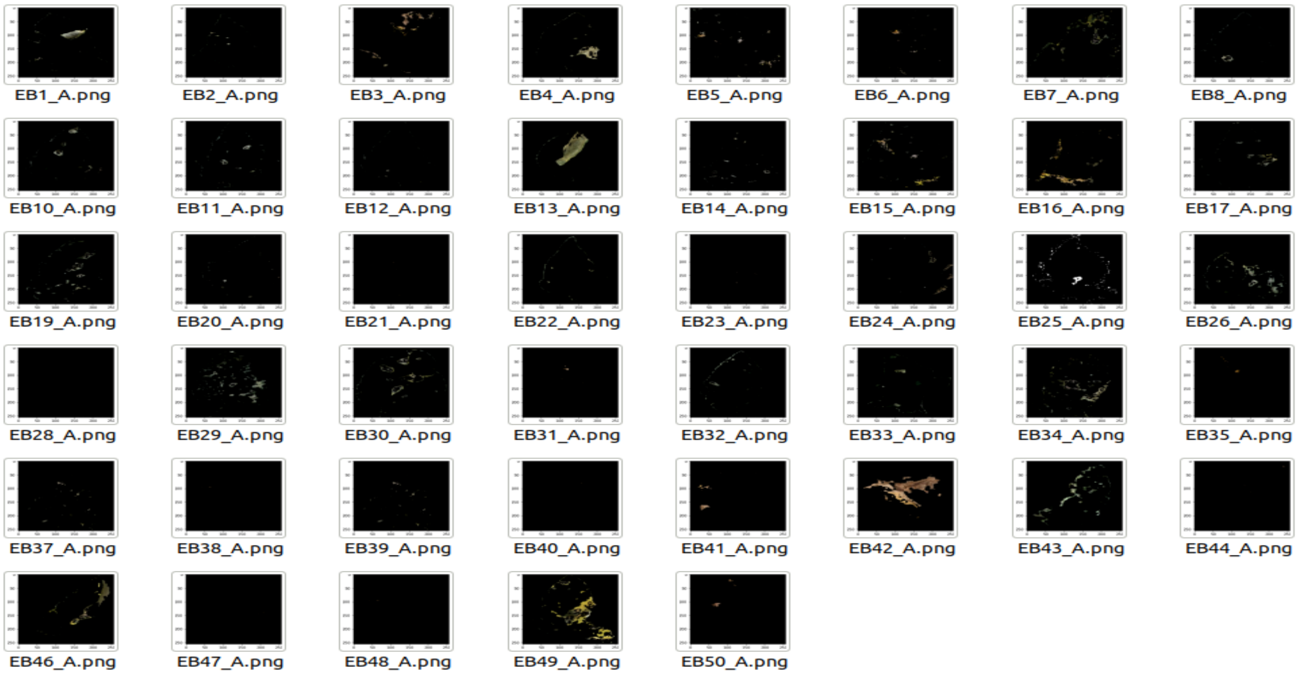


Fig. 20(b) Segmentation results for early blight in a channel

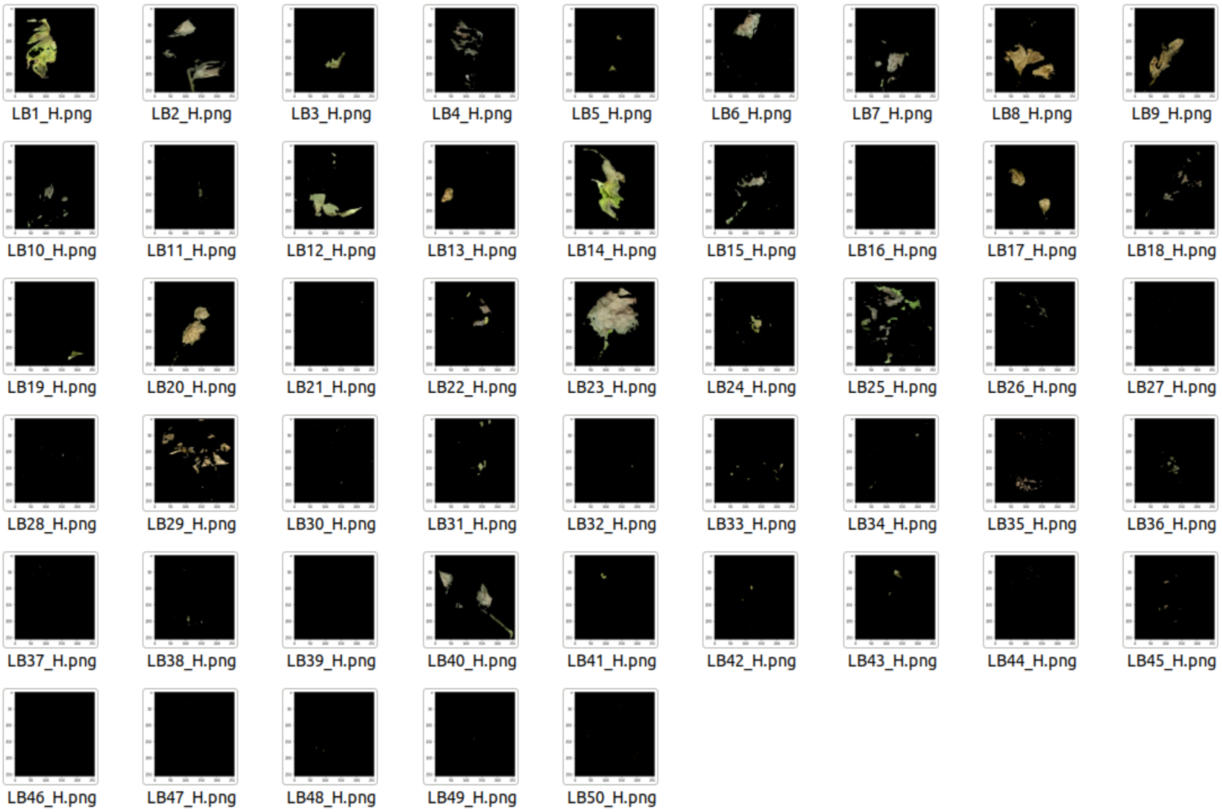


Fig 21(a) Segmentation results for late blight in h channel.

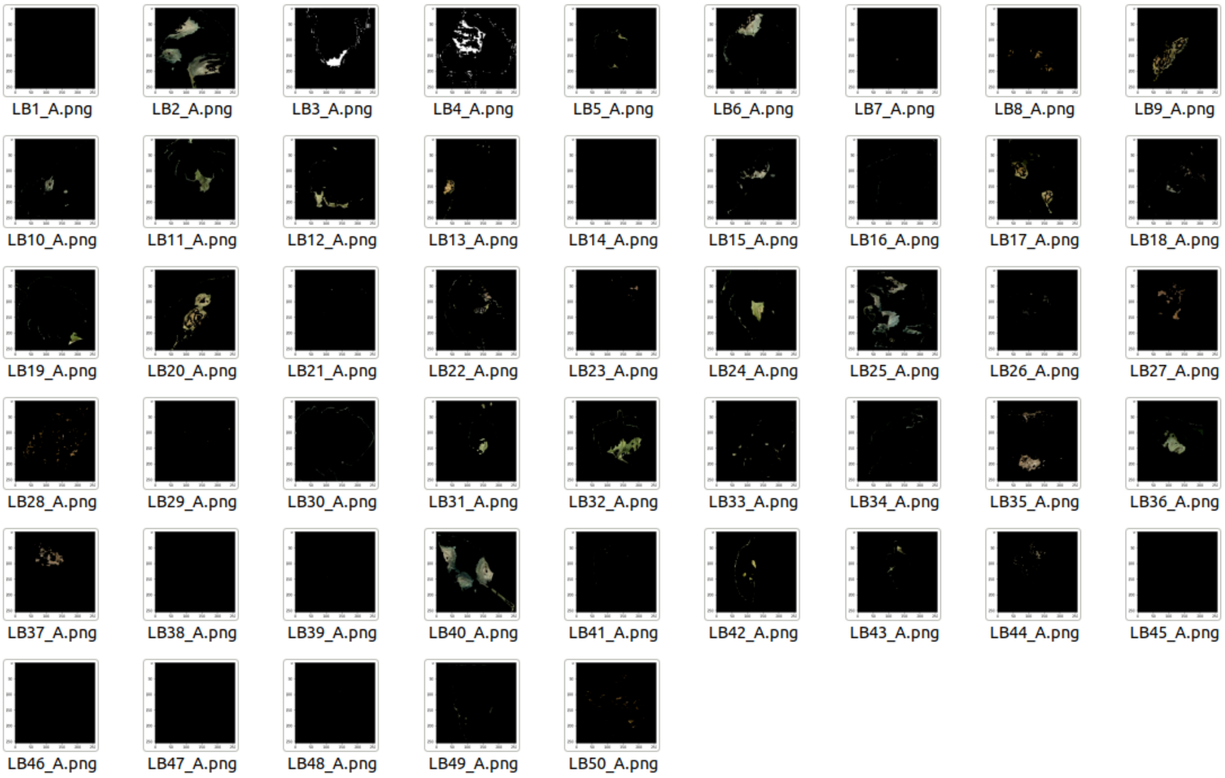
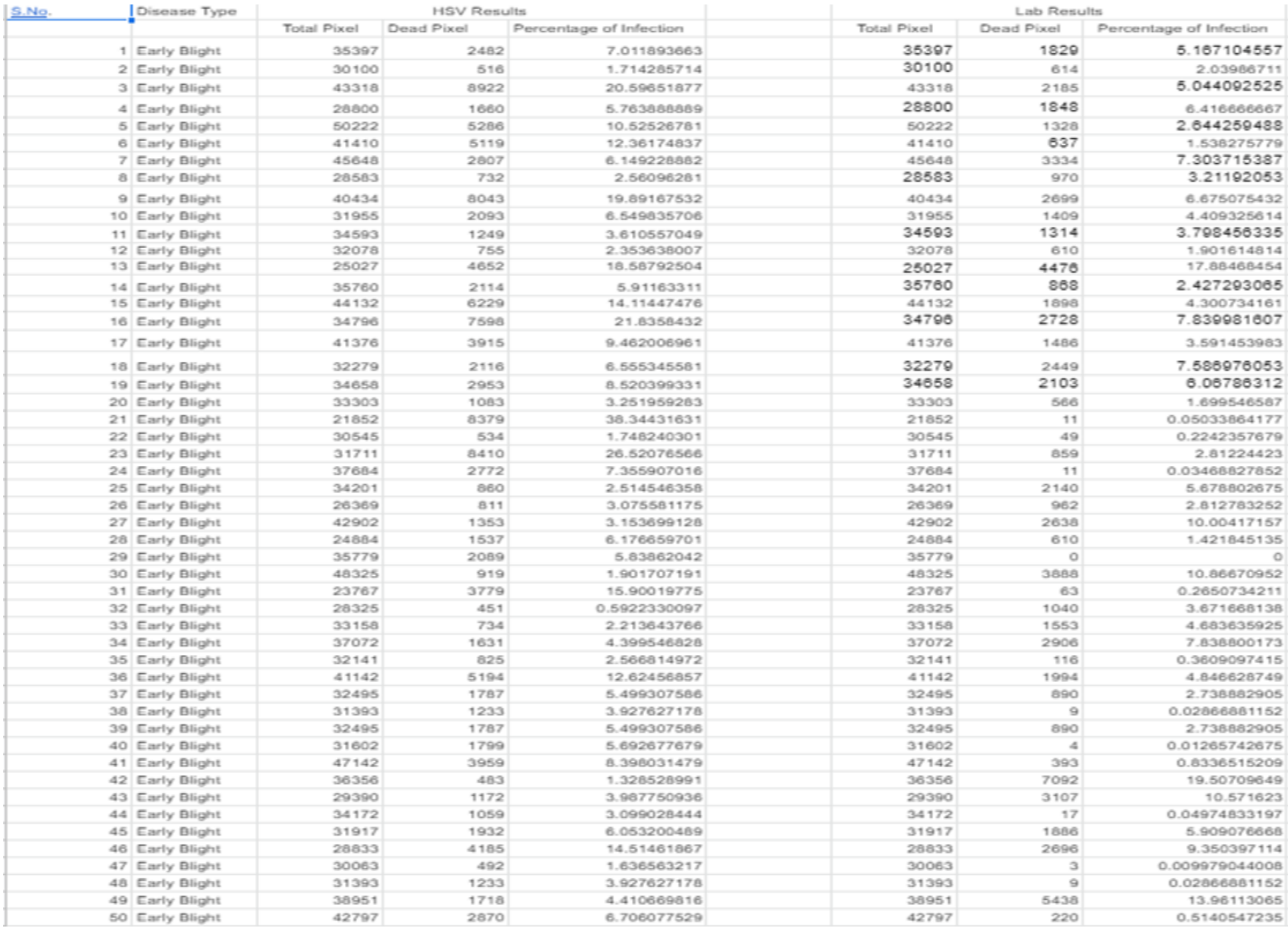
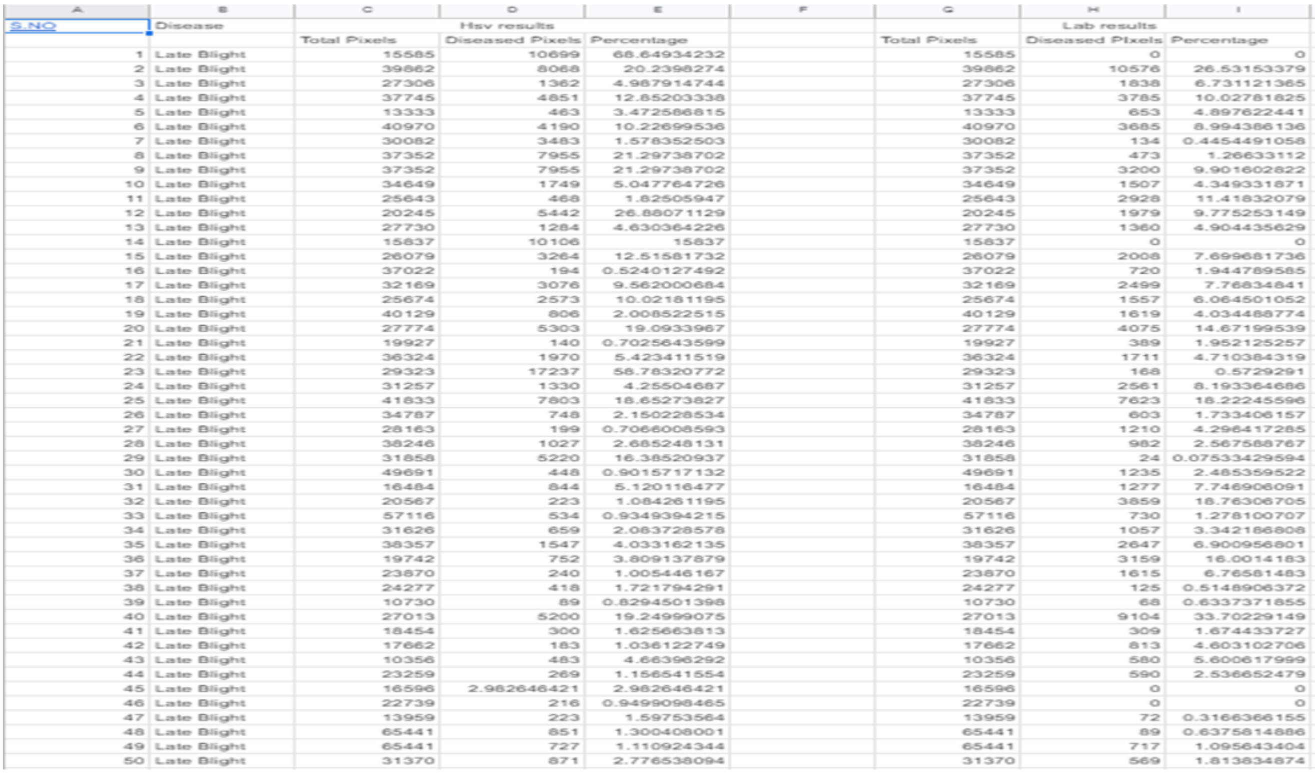


Fig. 21(b) Segmentation results for late blight in a channel



S.No.	Disease Type	HSV Results				Lab Results		
		Total Pixel	Dead Pixel	Percentage of Infection		Total Pixel	Dead Pixel	Percentage of Infection
1	Early Blight	35397	2482	7.011893663		35397	1829	5.167104557
2	Early Blight	30100	516	1.714285714		30100	614	2.03986711
3	Early Blight	43318	8922	20.59651877		43318	2185	5.044092525
4	Early Blight	28800	1660	5.763888889		28800	1848	6.416666667
5	Early Blight	50222	5286	10.52526781		50222	1328	2.644259488
6	Early Blight	41410	5119	12.36174837		41410	637	1.538275779
7	Early Blight	45648	2807	6.149228882		45648	3334	7.303715387
8	Early Blight	28583	732	2.56096281		28583	970	3.21192053
9	Early Blight	40434	8043	19.89167532		40434	2699	6.675075432
10	Early Blight	31955	2093	6.549835706		31955	1409	4.409325614
11	Early Blight	34593	1249	3.610557049		34593	1314	3.798456335
12	Early Blight	32078	755	2.353638007		32078	610	1.901614814
13	Early Blight	25027	4652	18.58792504		25027	4476	17.88468454
14	Early Blight	35760	2114	5.91163311		35760	868	2.427293065
15	Early Blight	44132	6229	14.11447476		44132	1898	4.300734161
16	Early Blight	34796	7598	21.8358432		34796	2728	7.839981607
17	Early Blight	41376	3915	9.462006961		41376	1486	3.591453983
18	Early Blight	32279	2116	6.555345581		32279	2449	7.586976053
19	Early Blight	34658	2953	8.520399331		34658	2103	6.06786312
20	Early Blight	33303	1083	3.251959283		33303	566	1.699546587
21	Early Blight	21852	8379	38.34431631		21852	11	0.05033864177
22	Early Blight	30545	534	1.748240301		30545	49	0.2242357679
23	Early Blight	31711	8410	26.52076566		31711	859	2.81224423
24	Early Blight	37684	2772	7.355907016		37684	11	0.03468827852
25	Early Blight	34201	860	2.514546358		34201	2140	5.678802675
26	Early Blight	26369	811	3.075581175		26369	962	2.812783252
27	Early Blight	42902	1353	3.153699128		42902	2638	10.00417157
28	Early Blight	24884	1537	6.176659701		24884	610	1.421845135
29	Early Blight	35779	2089	5.83862042		35779	0	0
30	Early Blight	48325	919	1.901707191		48325	3888	10.86670952
31	Early Blight	23767	3779	15.90019775		23767	63	0.2650734211
32	Early Blight	28325	451	0.5922330097		28325	1040	3.671668138
33	Early Blight	33158	734	2.213643766		33158	1553	4.683635925
34	Early Blight	37072	1631	4.399546828		37072	2906	7.838800173
35	Early Blight	32141	825	2.566814972		32141	116	0.3609097415
36	Early Blight	41142	5194	12.62456857		41142	1994	4.846628749
37	Early Blight	32495	1787	5.499307586		32495	890	2.738882905
38	Early Blight	31393	1233	3.927627178		31393	9	0.02866881152
39	Early Blight	32495	1787	5.499307586		32495	890	2.738882905
40	Early Blight	31602	1799	5.692677679		31602	4	0.01265742675
41	Early Blight	47142	3959	8.398031479		47142	393	0.8336515209
42	Early Blight	36356	483	1.328528991		36356	7092	19.50709649
43	Early Blight	29390	1172	3.987750936		29390	3107	10.571623
44	Early Blight	34172	1059	3.099028444		34172	17	0.04974833197
45	Early Blight	31917	1932	6.053200489		31917	1886	5.909076668
46	Early Blight	28833	4185	14.51461867		28833	2696	9.350397114
47	Early Blight	30063	492	1.636563217		30063	3	0.009979044008
48	Early Blight	31393	1233	3.927627178		31393	9	0.02866881152
49	Early Blight	38951	1718	4.410669816		38951	5438	13.96113065
50	Early Blight	42797	2870	6.706077529		42797	220	0.5140547235

Fig. 22(a) Quantification results for early blight



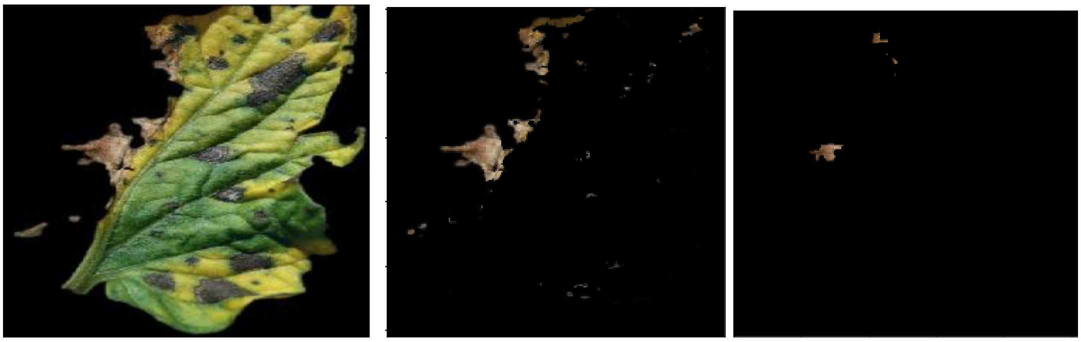
A	B	C	D	E	F	G	H	I
S.NO	Disease	Hsv results				Lab results		
		Total Pixels	Diseased Pixels	Percentage		Total Pixels	Diseased Pixels	Percentage
1	Late Blight	15585	10699	68.64934232		15585	0	0
2	Late Blight	39862	8068	20.2398274		39862	10576	26.53153379
3	Late Blight	27306	1362	4.987914744		27306	1838	6.731121365
4	Late Blight	37745	4851	12.85203338		37745	3785	10.02781825
5	Late Blight	13333	463	3.472586815		13333	653	4.897622441
6	Late Blight	40970	4190	10.22699536		40970	3685	8.994386136
7	Late Blight	30082	3483	1.578352503		30082	134	0.4454491058
8	Late Blight	37352	7955	21.29738702		37352	473	1.26633112
9	Late Blight	37352	7955	21.29738702		37352	3200	9.901602822
10	Late Blight	34649	1749	5.047764726		34649	1507	4.349331871
11	Late Blight	25643	468	1.82505947		25643	2928	11.41832079
12	Late Blight	20245	5442	26.88071129		20245	1979	9.775253149
13	Late Blight	27730	1284	4.630364226		27730	1360	4.904435629
14	Late Blight	15837	10106	15837		15837	0	0
15	Late Blight	26079	3264	12.51581732		26079	2008	7.699681736
16	Late Blight	37022	194	0.5240127492		37022	720	1.944789585
17	Late Blight	32169	3076	9.562000684		32169	2499	7.76834841
18	Late Blight	25674	2573	10.02181195		25674	1557	6.064501052
19	Late Blight	40129	806	2.008522515		40129	1619	4.034488774
20	Late Blight	27774	5303	19.0933967		27774	4075	14.67199539
21	Late Blight	19927	140	0.7025643599		19927	389	1.952125257
22	Late Blight	36324	1970	5.423411519		36324	1711	4.710384319
23	Late Blight	29323	17237	58.78320772		29323	168	0.5729291
24	Late Blight	31257	1330	4.25504687		31257	2561	8.193364686
25	Late Blight	41833	7803	18.65273827		41833	7623	18.22245596
26	Late Blight	34787	748	2.150228534		34787	603	1.733406157
27	Late Blight	28163	199	0.7066008593		28163	1210	4.296417285
28	Late Blight	38246	1027	2.685248131		38246	982	2.567588767
29	Late Blight	31858	5220	16.38520937		31858	24	0.07533429594
30	Late Blight	49691	448	0.9015717132		49691	1235	2.485359522
31	Late Blight	16484	844	5.120116477		16484	1277	7.746906091
32	Late Blight	20567	223	1.084261195		20567	3859	18.76306705
33	Late Blight	57116	534	0.9349394215		57116	730	1.278100707
34	Late Blight	31626	659	2.083728578		31626	1057	3.342186808
35	Late Blight	38357	1547	4.033162135		38357	2647	6.900956801
36	Late Blight	19742	752	3.809137879		19742	3159	16.0014183
37	Late Blight	23870	240	1.005446167		23870	1615	6.76581483
38	Late Blight	24277	418	1.721794291		24277	125	0.5148906372
39	Late Blight	10730	89	0.8294501398		10730	68	0.6337371855
40	Late Blight	27013	5200	19.24999075		27013	9104	33.70229149
41	Late Blight	18454	300	1.625663813		18454	309	1.674433727
42	Late Blight	17662	183	1.036122749		17662	813	4.603102706
43	Late Blight	10356	483	4.66396292		10356	580	5.600617999
44	Late Blight	23259	269	1.156541554		23259	590	2.536652479
45	Late Blight	16596	2.982646421	2.982646421		16596	0	0
46	Late Blight	22739	216	0.9499098465		22739	0	0
47	Late Blight	13959	223	1.59753564		13959	72	0.3166366155
48	Late Blight	65441	851	1.300408001		65441	89	0.6375814886
49	Late Blight	65441	727	1.110924344		65441	717	1.095643404
50	Late Blight	31370	871	2.776538094		31370	569	1.813834874

Fig. 22(b) Quantification results for late blight

4.3 Scope of Improvement

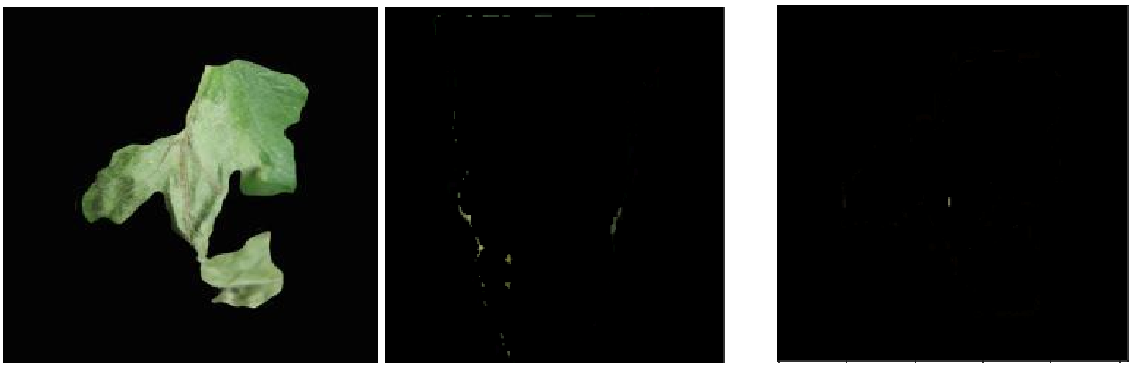
By experimental analysis, it was found out that for some images where h channel wasn't able to yield satisfactory results, a channel was able to get the correct results and vice versa. So, for some images, where one channel failed, the other channel was able to yield satisfactory results. However, this was not true for all the images. In some of the sample images both of the channels were unable to yield correct results.

The leaves which had a yellow colored diseased region and the leaves which had low contrast between the diseased and the healthy regions weren't segmented properly by the algorithm. Since yellow hue lies closer to green hue , the yellow infected regions might've been taken into the healthy group by the algorithm. Same is the case with diseases showing light green color symptoms. Fig. 23 shows the cases where the algorithm proposed by Barbedo(2016) yields incorrect results..



(i) Original Image (ii) Segmentation in the h channel (iii) Segmentation in the a channel

Fig 23(a) Figure showing results of the algorithm in the case of yellow infection pattern.



(i) Original Image (ii) Segmentation in the h channel (iii) Segmentation in the a channel

Fig 23(b) Figure showing results of the algorithm in the case of infection pattern with less contrast between healthy and diseased regions.

Fig 23. Images showing cases where the algorithm fails to yield the correct results

Proposed Solution

Since the semi-automatic segmentation algorithm as proposed by Barbedo(2016) failed to segment the diseased region in some cases,we decided to try an alternate route which could solve that problem. Our approach involves the use of a supervised machine learning algorithm known as the K-Means Clustering algorithm.

K-means Clustering

K-means Clustering is a supervised machine learning algorithm. This algorithm works in the same way classification algorithms work, the difference being that in the case of k-means clustering, the model itself determines which clusters are to be generated from the given data. In image processing K-means clustering can be used to segment the region of interest from other objects in the image. Pixels which have similar color values are placed in a similar cluster and can be easily distinguished from the other pixels in the image.

The algorithm is used in the case we have unlabeled data. The goal is to find groups having similar characteristics, and the number of groups is represented by "k" .The objective function for k-means clustering is given below :

$$\mathbf{J}=\sum_{j=1}^{k}\sum_{i=1}^{n}||x_i^{(j)} - c_j||^2$$

where,
J = objective function,
k = number of clusters,
n = number of cases,
$x_i^{(j)}$= case i,
c_j = centroid for cluster j,
$||x_i^{(j)} - c_j||^2$= distance function

In our case, we will use a and b channels from lab color space to form clusters of pixels. L channel is not considered because L represents the luminance information, on the other hand a and b channels separate the color information from the luminance information.So, in order to reduce the errors which might occur due to lighting effects, we decided to discard L channel to carry out our research.

Clusters with similar a and b values are grouped together such that the mean distance between the centroids and elements in the cluster is minimized. Each pixel in a cluster is then given the same color as that of the centroid of the cluster so that it becomes easy for us to identify different clusters and choose the diseased clusters.

The appropriate number of clusters k can be identified by the elbow method. The elbow method plots the values of cost functions generated by different values of k. As k increases, the average distortion keeps on decreasing. The improvements in average distortion decreases as k increases. So, the optimum value is that value of k where the average distortion declines the most. This will generate an elbow like curve so that's where its name comes from. A sample elbow curve is given in Fig. 24.

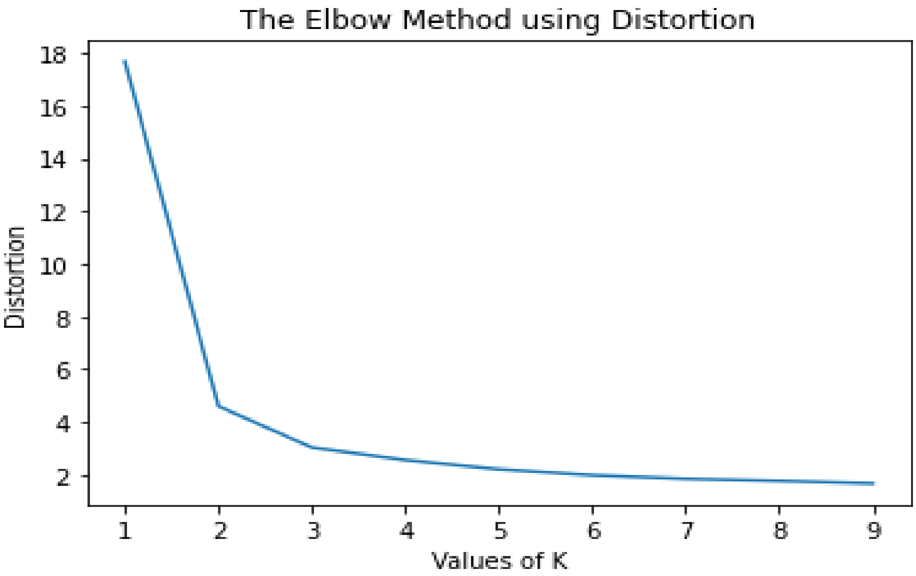
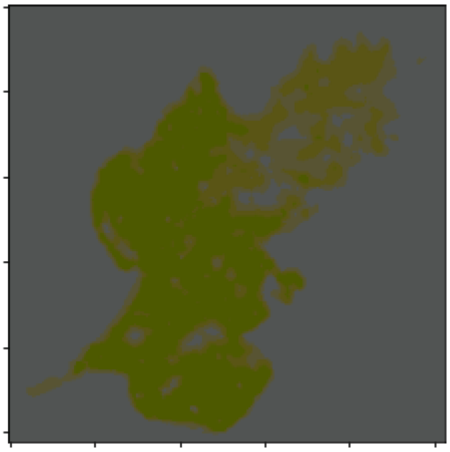


Fig 24 Elbow curve showing average distortion vs k. Here an elbow-like shape can be clearly seen. Either 2 or 3 can be chosen as the k value.

A result of successfully generated clusters can be seen in figure 25. To visualize the image in the rgb color channel, the luminance (L) value is kept around 90 and the resulting lab image was converted to rgb color image.

(a)



(b)

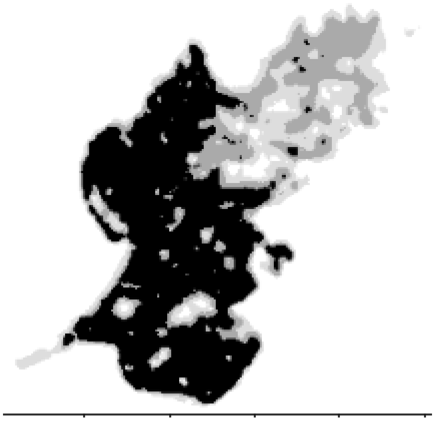
Fig 25.(a) Original Image.(b) Original image after k-means clustering with k value of 3.

Segmentation

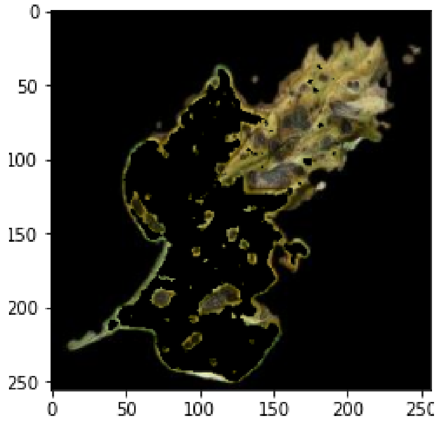
For segmentation purposes, two approaches can be followed. The first approach was to statically determine which clusters the diseased pixels belong to. But this approach is not that efficient.

The second approach involved first extracting the a-channel from the clustered image and then analysing the image in grey-scale. After experimental analysis it was found that healthy leaf regions, which are dark green in color, have the lowest a values, hence in grey-scale they represent the darkest color. Fig 26(b) shows the a-channel of a clustered leaf image in gray-scale.

(a)



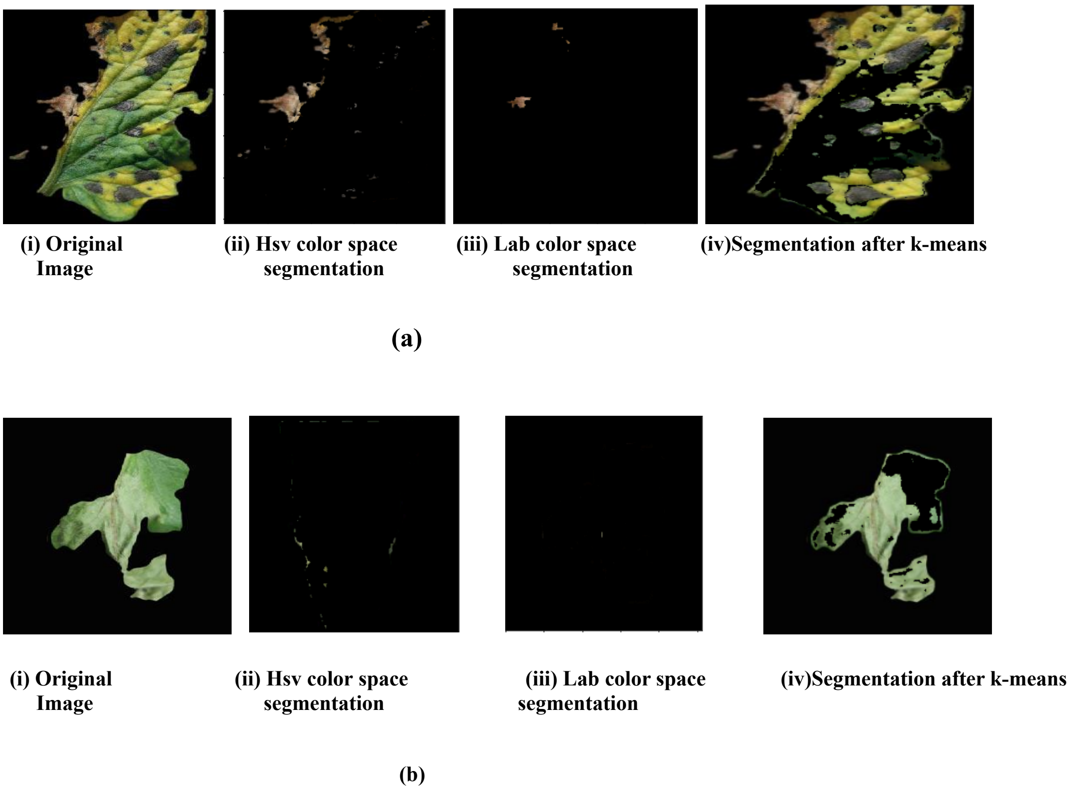
(b)



(c)

Fig 26. (a) Original leaf image. (b) A channel of the image visualized after clustering.Here healthy region can be clearly identified by dark black pixels.(C) Image showing segmented leaf image after removing the healthy region shown in figure (b).

A comparison between results given by the algorithm proposed by Barbedo(2016) and the k-means method is shown in Fig 27. Since most of the images produced a k value of 4, we chose 4 as our default k value.



(i) Original Image (ii) Hsv color space segmentation (iii) Lab color space segmentation (iv)Segmentation after k-means

(a)

(i) Original Image (ii) Hsv color space segmentation (iii) Lab color space segmentation (iv)Segmentation after k-means

(b)

Fig.27 Images showing comparison between various techniques we used for segmentation

So as we can see from the results shown in Fig27. that this approach does fit well when we want to segment the tomato leaf diseases having high color contrast.

4.4 Contribution of individual members

Aahan Singh
Aahan was responsible for quantifying diseases in HSV color space .Also researched about plant disease segmentation using k-means clustering.

Sarthak Sharma
Sarthak was responsible for quantifying diseases in the La*b* color space.Also researched about plant disease segmentation using k-means clustering.

Mr Aditya Sinha
Guided us throughout the duration of our project.

All algorithms and inferences were made with contributions from each member

5. Conclusion and Future Plans

The algorithm proposed by Barbedo(2016) produced excellent results for most of the disease variations. In some cases h color channel produced correct results while in some other cases a* color channel produced the results correctly. Since we considered images which were already segmented from the background, our next task will be taking an image in real-time using a camera and then quantifying the diseased portion. Such kind of software will be extremely helpful to the agriculture community, as it would help farmers understand how severe the disease is in real-time and shall help them take efficient decisions in a matter of seconds.

But in some cases like, in the case where there is a high contrast between different regions in the leaf, the algorithm is ineffective in producing desired results. Though sometimes correct results are produced too, but this case is rare. So,we decided to provide an alternative approach in the cases where the algorithm fails. K-means clustering can efficiently segment the diseased region of the leaf from the healthy region by dividing the area inside the leaf into clusters. After clustering, the diseased regions can be segmented by analyzing the a-channel in grayscale. But sometimes clustering can generate more regions than necessary, which might lead to inaccuracies.

Also, we are yet to apply the clustering technique on the rest of the images in the dataset. This is one of the first things which will be done in the future. If clustering does infact generate correct results for most of the images in the dataset, we will adopt it as the go-to segmentation technique in the software we are developing.

Quantification is a fairly easy procedure. It’s just a fraction of total pixels representing diseased regions in the leaf area. Even if the procedure is easy, we are yet to find sources for ground truth validation because without validation we can’t 100 percent say that our results are in fact correct. This issue will be hopefully resolved in the future.

We will also apply techniques like k-medoids clustering, fuzzy k-means clustering etc.. to segment the leaf images in order to find out the best technique to segment the images in our study. For now k-means segmentation will be the main focus.

Also one of the next things which we would like to do is to combine the process of detection and quantification, so that a hybrid software can be produced.

6. References

[1] *Bock, C. H., Poole, G. H., Parker, P. E., & Gottwald, T. R. (2010). Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging. Critical reviews in plant sciences, 29(2), 59-107.*

[2] *Barbedo, J. G. A. (2013). Digital image processing techniques for detecting, quantifying and classifying plant diseases. SpringerPlus, 2(1), 1-12*.

[3] *Yang, X., Beyenal, H., Harkin, G., & Lewandowski, Z. (2001). Evaluation of biofilm image thresholding methods. Water research, 35(5), 1149-1158*.

[4] *Ali, N. M., Rashid, N. K. A. M., & Mustafah, Y. M. (2013). Performance comparison between RGB and HSV color segmentations for road signs detection. Applied Mechanics and Materials, 393(1), 550-555.*

[5] *Bock, C. H., Barbedo, J. G., Del Ponte, E. M., Bohnenkamp, D., & Mahlein, A. K. (2020). From visual estimates to fully automated sensor-based measurements of plant disease severity: status and challenges for improving accuracy. Phytopathology Research, 2, 1-30.*

[6] *Barbedo, J. G. A. (2016). A novel algorithm for semi-automatic segmentation of plant leaf disease symptoms using digital image processing. Tropical Plant Pathology, 41(4), 210-224.*

[7] FAOSTAT. (2021). Retrieved 1 March 2021, from http://www.fao.org/faostat/en/#data

[8] Gonzalez, R. C., & Woods, R. E. (2018). *Digital image processing* (4th ed.). Uttar Pradesh: Pearson India.