PLANT LEAF DISEASE DETECTION

UNDERGRADUATE RESEARCH OF PROJECT

Submitted by

Kollipara Naga Prasanna Suparnika(AP19110010480)

Nathani Akhila(AP19110010322)

Sadineni Navya(AP19110010417)

Kavuru Manansa(AP19110010343)

SUBMITTED IN PARTIAL FULFILLMENT OF THE DEGREE

of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SRM UNIVERSITY- ANDHRA PRADESH-620 015

ACKNOWLEDGEMENT

We feel very happy for completing our project successfully associated with SRM University. We would like to convey my heartfelt gratitude to Ms. Priyanka Singh for her tremendous direction and assistance in the completion of our project. We have learned a lot from the weekly meetings that will be useful in the future process of our research project as well. It was a fantastic opportunity for both education and personal growth. As a result, we feel really fortunate to have been given the opportunity to be a part of it.

We would like to thank SRM AP for providing us an opportunity to do a Research project. I would like to take this opportunity to express my gratitude to all of my group members Akhila Nathani, Manasa Kavuru and Navya Sadineni. We would not have been able to complete this project without their help and cooperation. We will never forget this experience and we ensure that all the details provided in this report are true.

I would like to convey my heartfelt gratitude to all of my team members, for participating in beneficial choices and providing essential counsel, huge support, and encouragement to one another, and organizing all facilities to make the Research work simpler. I've chosen this opportunity to gratefully appreciate their effort. This opportunity is a significant step forward in our professional growth. In order to achieve desired professional objectives, we will try to apply acquired skills and knowledge as effectively as possible, and we will continue to work on their improvement. We want to continue working with you all in the future.

Yours Truly,

Name: K.N.P. Suparnika, Manasa Kavuru, Navya Sadineni and Akhila Nathani

Place: SRM-AP

Date: 9th May 2022.

ABSTRACT

This paper holds a study on plant leaf diseases classification utilizing image processing. India is one of the nations who depends more on the agricultural segment. Acknowledgment of the plant diseases are imperative in arrange to avoid the misfortunes inside the surrender of crops. It is more troublesome to distinguish the infections physically in our day by day lives. Advanced image handling has three basic ways: image processing, analysis and understanding. Picture preparing contains the pre-processing of the plant leaf as division, color extraction, disease particular information extraction and filtration of images.

Load the image, convert it into grayscale image, applied thresholding, finding area and perimeter of leaf, Gaussian blur of grayscale image, performed Otsu thresholding, later we applied morphological transform which includes dilation erosion and closing, Splitting the original image to b,g,r layers and find the mean and standard deviation for each layer. Bitwise AND operation between the original image and closing image, By using GLCM, contrast, dissimilarity, homogeneity,energy, correlation and angular second moment features are extracted. Here, each feature has 6 values. Therefore considered as 24 features. All the features are stored in an array along with the type of the leaf under class level column. This is performed for all the images of all types. Finally all the data is stored in a CSV file.

1. INTRODUCTION

India is one of the largest countries which produces food products like wheat, maize, pulses, fruits and numerous vegetables etc. Agriculture is the primary power house of India. Crops are affected by different types of diseases and are the major problem for food security. But the crop disease identification is very difficult to find physically due to lack of infrastructure. Detecting the disease name exactly may not lead to excess usage of pesticides and fertilizers. So, if the farmers have knowledge about various types of diseases for a variety of plants, they may reduce the unwanted usage of pesticides and they were able to identify the fertilizer which actually kills that particular disease. This may lead to good yield in agriculture that benefits the farmers as well as the environment will not get polluted by over usage of herbicides. So, detecting the diseases by doing experiments like soil analysis, biochemical methods, microscopic examination on leaves in the laboratory may consume a lot of cost and time. The combination of expanding worldwide smartphone infiltration and later propels in computer vision made conceivable by profound learning has cleared the way for smartphone-assisted disease determination. So, using machine learning algorithms will give better accuracy within a short period of time and also the cost would be low. However, food security remains threatened by a number of factors including climate change , the decline in pollinators , plant diseases, and others. Modern technologies have given human society the capacity to create sufficient nourishment to meet the request of more than 7 billion individuals.

2. MOTIVATION

Since, we are interested in machine learning and we wish to contribute something to the farmers which can help them to increase their crop yield and we found this is the best way to do it.

3. LITERATURE REVIEW

Different types of techniques and methodologies had been discussed during the research project in detecting and identifying plant diseases. Front. Plant Sci, SP Mohanty 22 September (2016) presented a technique in which a pre-processing step involved conversion of RGB to grayscale, image resizing and Background noise removal. Segmentation was robotized by the implies of a script tuned to perform well on our specific dataset. We chose a procedure based on a set of veils produced by examination of the color, softness and immersion components of diverse parts of the pictures in a few color spaces. One of the steps of that preparation moreover permitted us to effortlessly settle color casts, which happened to be exceptionally solid in a few of the subsets of the dataset, in this way expelling another potential predisposition. Lastly, the author used neural networking for classification and recognition of leaf diseases.

3.1. Pranesh Kulkarni (2021) presented a technique in which pre-processing step involved original to grayscale, smoothing using gaussian filter, Ostu's Thresholding, Morphological Transform, Bitwise AND operation with original bit frame, Grey level co-occurrence matrix and HSV color space conversion. At last Random Forest classifier is used for classification and disease detection.

3.2. G Geetha (2020) presented a technique in which Pre-processing, Segmentation, Feature Extraction and Classification of KNN are used for recognition of tomato plant disease identification.

3.3. Sandesh Raut et.al (2017) presented a technique in which pre-processing involved image resizing, noise removal and conversion of RGB images to gray level. In segmentation, centroid value is calculated using the k-means method for dividing image into object. Factual surface highlights extraction utilizing Gray-Level Co-Occurrence Network setup method. Lastly authors used multi-SVM for classification and recognition of leaf and fruit diseases.

3.4. Arti N. Rathod et.al (2014) proposed a technique for detecting diseases in leaves. This creates the color structure RGB to CIELAB transformation. After applying the K-Medoid cluster and removing the masked green pixels. The GLCM function to calculate the texture features statistics in an effective manner. Lastly author at long last to recognize the infections utilizing neural network configuration.At long last

3.5. K.Gowthami et.al (2017) proposed a technique that can be used for detecting of black spot and anthracnose leaf diseases in plants. This RGB image was converted into a gray-level image to separate the infected parts from the leaf. Different types of segmentation techniques like Otsu thresholding, k-means clustering, Boundary & Spot detection algorithms are used here. Compute the texture features using color co-occurrence methodology. By using ANN and back propagation classification, it can identify the disease of the infected plant.

3.6. Prof. Shripad S.Veling et.al (2019) used the MATLAB tool for mango disease detection on leaf, fruit and flower. FastRobust Fuzzy C-Means clustering algorithm is used for segmentation of leaf images and the Gray-Level Co-occurrence (GLCM) method is used for infected leaf texture analysis. For classification of plant diseases, support vector machines (SVM) are used.

3.7. Sharath D M, Akhilesh,et.al (2019) presented a system on pomegranate fruit disease detection.For commotion expulsion employment a Gaussian channel and get cut strategy for picture division. Identify the bacterial blight diseases using canny edge detection technique. Based on the data comparison procedure to finalize whether the fruit is infected or not. It also suggests a farmer to give proper solutions on how to overcome the infection problem.

3.8. Namrata R.Bhimte et.al(2018) discussed a disease detection method for cotton plant leaves. In this, a cotton leaf image is taken as input. The RGB Image values were converted into color space. Then by using k means clustering algorithm, cotton leaf disease segmentation is done. Gray-Level Co-Occurrence Matrix is used for feature extraction to see various statistics such as mean, standard deviation, contrast, energy, entropy, homogeneity and correlation. Finally for classification and detection of disease in the leaf used SVM algorithm.

3.9. Ms. Kiran R et.al (2014) described techniques for detecting diseases like anthracnose, citrus canker, over-watering and citrus greening in citrus plants. Image pre-processing involved YCbCr color system, color space and discrete cosine transform for color space conversion. Along with that, extraction is done with the assistance of Gray-Level Co-Occurrence Network. Radial basis kernel and polynomial support vector machines are used for classification of fruit diseases.

4. BACKGROUND DETAILS

Different Models

4.1. K-Nearest Neighbour (KNN)

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. The KNN algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories.KNN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suited category by using K- NN algorithm. The KNN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.

KNN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. The KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

4.1.1. Advantages of KNN algorithm

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

4.1.2. Disadvantages of KNN algorithm

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

4.2. Support Vector Machine

The goal of the Support Vector Machine algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine. SVM can be two types:

4.2.1. Linear SVM

Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and the classifier is used as Linear SVM classifier.

4.2.2. Non-linear SVM

Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and the classifier used is called Non-linear SVM classifier.

4.3. Decision Tree

Decision tree induction is the learning of decision trees from the class labeled training tuples. A decision tree is a flowchart like tree structure. Where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node holds a class level. The top most node in the tree is the root node.

The algorithm consists of three parameters:

D: D is the data partition, initially it is the complete set of training tuples and their class levels.

Attribute_list: It is the list of attributes that describes the tuples.

Attribute_selection_method: It specifies a heuristic procedure for selecting the attribute that best discriminates the given tuples according to class.

Attribute selection method is used to determine the splitting criterion. The splitting criterion tells us which attribute to test at node N by determining the best way to separate or partition the tuples in D into individual classes.

4.4. Random Forest

Random forest is a supervised machine learning algorithm used for classification as well as for Regression. Random forest classifier is a type of ensemble algorithm. Builds a set of decision trees from a randomly selected subset of training sets. It then combines votes from different decision trees to decide the final class of the test object. The Decision of the majority of the trees is chosen by the random forest as the final decision.

4.4.1. Application of Random Forest

- Random forest used in ETM devices to acquire images of the earth’s surface.
- Multi-class detection is done using random forest

4.4.2. Advantages of Random Forest

- Accuracy is higher and training time is less
- Providing better detection in complicated environments
- Use of multiple trees reduce the risk of over fit
- Runs efficiently on large database
- Random forest can maintain accuracy when a large portion of data is missing

4.4.3. Disadvantages of KNN algorithm

- The random forest model is used for both classification and regression but mostly applicable for regression analysis.

4.5. Logistic Regression

Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. Logistic Regression gives probabilistic values which lie between 0 and 1. In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1). In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

4.5.1. Assumptions taken for Logistic Regression

- The dependent variable must be categorical in nature.
- The independent variable should not have multi col-linearity.

4.6. The Multi Layer Perceptron (MLP)

The Multi layer Perceptron classifier is a supervised learning algorithm used for classification or regression. Multi layer perceptron is also known as MLP. These are fully connected, dense layers that convert any input dimension to the desired dimension. Layered cognition is a neural network with multiple layers. To create a neural network, combine neurons so that the output of some neurons is the input of another. A multi layered perceptron has the same structure of a single layer perceptron with one or more hidden layers.

4.6.1. Advantages of Multi layer Perception

- ➢ Ability to learn nonlinear models.
- ➢ Ability to learn models in real time (online learning)

4.6.2. Disadvantages of Multi layer Perception

- ➢ The Multi layer Perceptron (MLP) has a non-convex loss function when multiple local minima values are present. Therefore, different random weight initialization can result in different validation accuracy.
- ➢ With the MLP, we need to adjust a number of hyper parameters, such as: Number of hidden neurons, layers, iterations.
- ➢ The MLP is sensitive to functional scaling.

5. PREFERRED MODEL

5.1. Pre-Processing Steps

5.1.1. Load the image

First, we get all the paths of the folders in which images are present (since one folder contains only one leaf type). Using glob function, all the images in the folder are extracted in an array. We had used open CV2 to load the dataset. Open the image file. The format of the file is often JPEG, PNG, BMP, etc. Resize the image to match the input size for the Input layer of the model. Convert the image pixels to float data types. Normalize the image to possess pixel values scaled down between 0 and 1 from 0 to 255. Image data for models should be either a NumPy array or a tensor object.

5.1.2. Convert into grayscale image

Convert an image to Grayscale in python using the image. Convert () method of the pillow library. Convert an image to Grayscale in python using the color. rgb2 gray () method of the scikit image module. Convert an image to Grayscale in python using the cv2.imread() method of the OpenCV library. Convert the image to Grayscale in python using the conversion formula and the matplotlib library. There are various methods to convert an image to grayscale in Python.

A grayscale image is an image in which a single pixel represents the quantum of light or only contains light intensity information. It is a single-dimensional image and has a different memorial of gray color only. As the grayscale images are single-dimensional, they’re wont to decrease models' training complexity in various problems and in algorithms just like the Canny edge detection.

The mode includes 1-bit and 8-bits pixel black and white images, RGB images, HSV images, BGR images and LAB images, etc. As we want to convert our image to grayscale, we can pass 1 as mode argument for the 1-bit black and white mode, L for 8-bits black and white image, and LA for alpha mode.

5.1.3. Thresholding

We use skimage functions to apply thresholding to an image. Thresholding is a type of image segmentation, where we change the pixels of an image to make the image easier to analyse. In thresholding, we convert a picture from color or grayscale into a binary image, i.e., one that's simply black and white. Most frequently, we use thresholding as a way to pick areas of interest in a picture, while ignoring the parts we aren't concerned with. We want to get rid of the pixels belonging to the shapes "on," while turning the remainder of the pixels "off," by setting their color channel values to zeros. The skimage library has several different methods of thresholding. We will start with the only version, which involves a crucial step of human input. Specifically, during this simple, fixed-level thresholding, we've to supply a threshold value t. The process works like this. First, we'll load the original image, convert it to grayscale, and de-noise it as within the Blurring episode.

Next, we might wish to apply the threshold t such pixels with grayscale values on one side of t are going to be turned "on", while pixels with grayscale values on the other side will be turned "off". Remember that grayscale images contain pixel values within the range from 0 to 1, so we are trying to find a threshold t within the closed range [0.0, 1.0]. We see within the image that the geometric shapes are "darker" than the white background but there's also some light gray noise on the background. One way to determine a "good" value for t is to look at the grayscale histogram of the image and try to identify what grayscale ranges correspond to the shapes in the image or the background. The histogram for the shapes image can be produced as in the Creating Histogram episode.

Since the image features a white background, most of the pixels within the image are white. This corresponds nicely to what we see in the histogram: there is a peak near the value of 1.0. If we would like to select the shapes and not the background, we would like to turn off the white background pixels, while leaving the pixels for the shapes turned on. So, we should always choose a value worth of t somewhere before the massive large peak and switch pixels above that value "off". Let us choose t=0.8

To apply the threshold t, we can use the NumPy comparison operators to create a mask. Here, we want to turn "on" all pixels which have values smaller than the threshold, so we use the less operator < to compare the blurred image to the threshold t. The operator returns a mask that we capture in the variable binary mask. It has just one channel, and every of its values is either False or True. The binary mask created by the thresholding operation can be shown with plt. imshow, where the False entries are shown as black pixels (0-valued) and the True entries are shown as white pixels (1-valued). We can observe that the areas where the shapes were in the original area are now white, while the rest of the mask image is black. We use Gaussian blur with a sigma of 1.0 to denoise the leaf image. Let us check out the grayscale histogram of the denoised image.

We will first construct a Python program to measure this value for a single image. Our strategy will be:

- Read the image, converting it to grayscale because it is read. For this application we don't need the color image.
- Blur the image.
- Use Otsu's method of thresholding to create a binary image, where the pixels that were part of the leaves of the plant are white, and everything else is black.
- Save the binary image so it can be often examined later.
- Count the white pixels within the binary image, and divide by the number of pixels in the image. This ratio will be a measure of the leaf mass of the plant within the image.
- Output the name of the image processed and the leaf mass ratio.

Our intent is to perform these steps and produce the numeric result – a measure of the leaf mass in the image – without human intervention. Implementing the steps within a Python function will enable us to call this function for various images.

Thresholding produces a binary image, where all pixels with intensities above (or below) a threshold value are turned on, while all other pixels are turned off. The binary images produced by thresholding are held in two-dimensional NumPy arrays, since they need just one-color value channel. They are boolean, hence they contain the values 0 (off) and 1 (on). Thresholding is often used to create masks that select only the interesting parts of a picture, or as the first step before edge detection or finding contours.

Finding area and perimeter of leaf: The Perimeter is the length of the arc of the leaf. Area is defined as the area enclosed by the arc of the leaf.

5.1.4. Gaussian blur for the gray scale image

It is the image processing algorithm that enables image manipulations. Gaussian functions are used for several purposes. Both grayscale and color images contain a lot of noise or random variation in brightness or tint among pixel values. The pixels in these images have a high standard deviation which implies that there is a lot of variation within groups of pixels. Since a photo is two-dimensional Gaussian blur uses two mathematical functions (one for the x axis and one for y) to create a third function, also known as a convolution. The third function creates a normal distribution of those pixel values, smoothing out a few of the randomness. How much smoothing depends on the measure of the blur radius you choose. Each pixel will choose up a new value set to a weighted average of its surrounding pixels, with more weight given to the closer ones than to those further away. The result of all this math is that the picture is hazier.

5.1.5. OTSU’S Threshold

Image thresholding is utilized in numerous applications as a pre-processing step. Thresholding is utilized to binarize the image based on pixel power. Otus’s thresholding methodology includes repeating through all the conceivable edge values and calculating a degree of spread for the pixel levels on each side of the limit i.e., the pixels that either drop in foreground or background. The point is to discover the threshold value where the sum of foreground and background spreads is at its minimum. The input to such thresholding calculation is usually a gray scale image and the output is a binary image. If the intensity of a pixel is greater than a threshold, the output pixel is marked as white(foreground), and if the intensity of a pixel is less than or equal to the threshold, the output pixel is marked as black(background).

5.1.6. Morphological Transform

Morphological operations are basic changes applied to binary or gray scale images. More particularly, we apply morphological operations to shapes and structures inside of images. We use morphological operations to extend the size of objects in images as well as diminish them. We can also use morphological operations to close gaps between objects as well as open them. Morphological operations probe an image with a structuring element. This structuring element characterizes the neighborhood to inspect around each pixel. And based on the given operation the measure of the structuring element we are able to adjust our output image.

There are four different types of morphological operations in image processing:

1.Dilation

2.Erosion

3.Opening

4.Closing

5.1.6.1. Dilation

Dilation includes pixels to the boundaries of objects in an image. The number of pixels included or expelled from the objects in a picture depends on the measure and shape of the structuring component utilized to prepare the image. The value of the output pixel is the greatest value of all the pixels within the neighborhood. In a binary image, a pixel is set to 1in case any of the neighboring pixels have the value 1. Morphological dilation makes objects more visible and fills in little gaps in objects. Lines show up thicker, and filled shapes show up bigger.

5.1.6.2. Erosion

Erosion expels pixels to the boundaries of objects in an image. The value of the output pixel is the least pixel of all pixels within the neighborhood. In a binary image, a pixel is set to 0 if any one of the neighboring pixels has the value 0. Morphological erosion expels drifting pixels and lean lines so that only substantive objects stay. Remaining lines show up slenderer and shapes show up smaller.

5.1.6.3. Opening

You can combine dilation and erosion to remove little objects from an image smooth the border of huge objects. The definition of a morphological opening of an image is an erosion taken after by a dilation. Opening operation erodes an image and then dilates the eroded image, using the same structural element for both operations. Morphological opening is valuable for expelling little objects and lean lines from an image whereas protecting the shape and size of bigger objects within the image.

5.1.6.4. Closing

The definition of a morphological closing of an image is a dilation followed by an erosion. The closing operation dilates an image and then the dilated image will be eroded using the same structural element. Morphological closing is valuable for filling little gaps in an image whereas protecting the shape and size of large gaps and objects within the image.

5.1.7. Bitwise AND Operation on original image

Bitwise operations can be utilized in image manipulations and also utilized when working with masks. Masks are binary images that show the pixels in which an operation is to be performed. These bitwise strategies are utilized in numerous computer vision applications like for creating masks of the picture, including watermarks to the image and it is conceivable to make a modern image utilizing these bitwise operators. These operations work on the individual pixels within the image to provide exact results compared with other morphing procedures in OpenCV.   In bitwise operators firstly, the original image of the leaf is splitted into r,g,b and we have to apply bit wise and operation with each layer of original image and closing image, later we will merge the image and convert it into grayscale image and by using grayscale image we will calculate Gray level co-occurrence matrix.. This bit wise and operation as it were, considering pixels that are common with the original image of leaf and closing image of the leaf and remaining pixels are expelled from the output image.

5.1.8. Image texture

Textures are complex visual designs, composed of spatially organized substances that have characteristic brightness, color, shape, size. Texture is a description of the spatial arrangement of color or intensities in an image or a selected region of image. Texture is homogeneous for the human visual system. There are several types of methods for texture features; those are Statistical, Structural, Model and Filter. Texture strategy: Statistical approach First arrange : Surface measures are measurements calculated from the first image values like fluctuation and don't consider pixel neighbor relationships Second arrange: measures consider the relationship between bunches of two pixels within the unique picture Third and higher arrange : surfaces considering the connections among three or more pixels are hypothetically conceivable but not commonly executed due to calculation time and translation difficult. The simple one-dimensional histogram is not useful in characterizing texture. For example, all three images have the same histogram. Hence a two-dimensional dependence matrix known as gray level co-occurrence matrix is extensively used in texture analysis.

5.1.9. Texture properties

- Texture is the property of the area. Texture of a point is undefined.
- Texture involves spatial distribution of Gray levels.
- Texture in an image can be seen at diverse levels of resolution.
- Texture is seen when critical individual shapes are not displayed.
- Texture is ordinarily from a single band.
- Texture highlights from distinctive groups of the image are for the most part diverse and have diverse segregating capabilities of land cover sorts.

5.1.10. Gray level co-occurrence matrix

It is the foremost classical moment to arrange statistical strategy for texture analysis. An image is composed of pixels each with a concentrated indicated Gray level, the GLCM may be an organization of how frequently diverse combinations of Gray levels co-occur in an image or picture determination. Also referred to as co-occurrence disseminations. Texture calculations utilize the substance of the GLCM to grant a degree of the variety concentrated at the pixel of interest.

These operations work on the individual pixels within the image to provide exact results compared with other morphing procedures in OpenCV. Each image is composed of pixels each with an intensity of pixel value, brightness value or a Gray level. If we are looking at a part of an image GLCM tells us about what is the frequency of finding a pair of pixels or pair of Gray levels in a particular orientation in an image over an area.

It is also referred to as co-occurrence distribution. Because how these pixel values will co-occur and these are distributed throughout the image. After applying bitwise and operation we will convert it into a grayscale image, with the help of a grayscale image the GLCM matrix is calculated. From the GLCM matrix the texture measures are calculated and these texture measures tell us about how the variation of intensity is there for a particular pixel.

5.1.11. Computation of co-occurrence matrix

It has a size N × N (N= number of Gray values) i.e., GLCM is a square matrix with the same number of rows and columns as the quantization level of image (image should be resampled to not more than 4 bit).

It is computed based on two parameters

D- Relative distance between the pixel pair.

θ-Relative orientation/ rotational angle.

0 degree horizontal, 45-degree font diagonal, 90 degrees vertical, 135 back diagonals. At 45-degree angle we have divided the entire 360 degree and thus we have caught 8 neighboring pixels of a particular pixel so, these 8 directions we calculate the GLCM. If we are calculating the GLCM at 0 degree then always we are finding out the relationship or pair of pixels with the pixel on horizontally in the same row on the right-hand side on so on. In this way for a particular Angel, we can have 8 GLCM matrices.

5.1.12. Features on co-occurrence matrix

From the original image we have calculated a co-occurrence matrix on this co-occurrence matrix we will generate the 2nd order statistics with different features. There are many numbers of features which can be calculated on this GLCM matrix.

5.1.12.1. Contrast

- Measures the spatial frequency of an image and difference moment of GLCM
- It is the contrast between the most elevated and the most reduced values of a contiguous set of pixels
- It measures the sum of neighborhood varieties displayed within the image.
- In the areas where the texture is more contrast is more.
- There is some linear pattern in the original image. Though this linear pattern is not very distinguishable (clear) in the original image, it is very clear in contrast.
- There are some white patches in the original image in these white patches when we look at the contrast of white patches. What we can see is at the edges where the brightness value is changing and on the edge the contrast is more.
- This brightness area represents homogeneous area

5.1.12.2. Dissimilarity

- Instead of weights increasing exponentially as one moves away from the diagonal as contrast did, the dissimilarity weights increase linearly
- In homogeneous areas the dissimilarity would be less whereas in heterogeneous areas the dissimilarity would be high.

5.1.12.3. Homogeneity

- Homogeneity is called as Inverse Difference Moment,
- Measures image homogeneity as it assumes larger values for smaller gray tone difference in pair elements
- It has most extreme value when all components within the image are same
- Homogeneity decreases if the contrast increases while energy kept constant
- Homogeneity as the name itself suggests if the area is homogeneous that means same gray values so homogeneity is more
- The diagonal elements the homogeneity is more as we move away from the diagonal elements the homogeneity gets decreased

5.1.12.4. Energy

Gives the sum of squared components within the GLCM

5.1.12.5. Correlation

Measure the joint probability event of the required pixel sets

5.1.12.6. Angular Second Moment(ASM)

- Angular second Moment called Uniformity
- Measure the textural uniform ability that is pixel pair repetitions
- Detects disorders in texture
- Energy comes to a greatest value rise to to one

By using GLCM, contrast, dissimilarity, homogeneity, energy, correlation and ASM features are extracted. Here, each feature has 6 values. Therefore, it is considered to have 24 features.

After completing all the pre-processing steps for each image 32 features have been extracted and this process will be done for all the images then the data will be stored in a CSV file.

Data is then split into training and testing data in the ratio of 7:3. After the data is splitted x_train and x_test will be normalized using standard scalar normalization technique to get better accuracy. Then we have applied all the above 7 classifiers among them MLP classifier gave more accuracy i.e., 74.85 and SVM have also gave the same i.e,74.57. So, our preferred models are SVM and MLP.

5.2. FLOW CHART

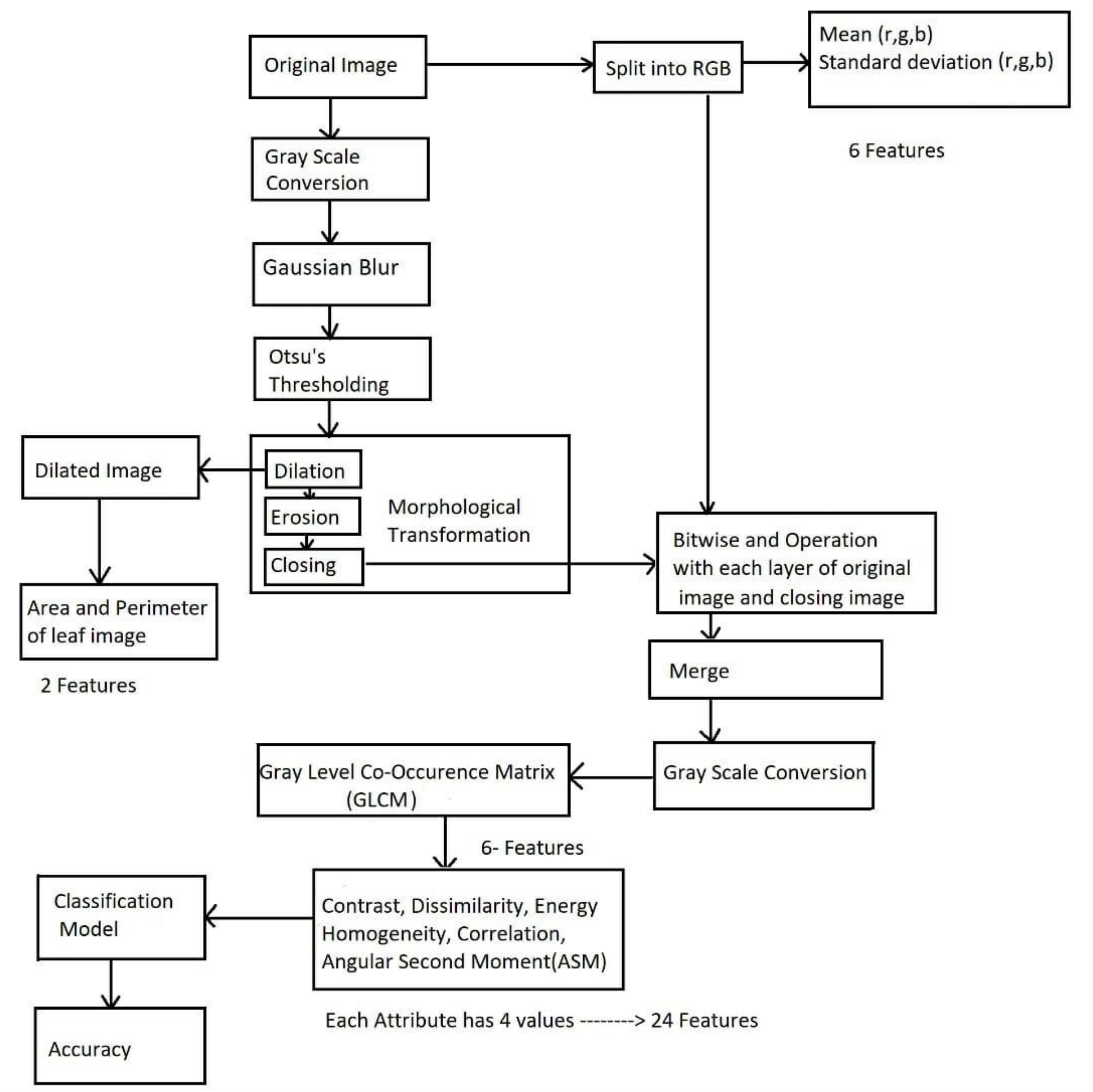
Original Image
Split into RGB
Mean (r,g,b)
Standard deviation (r,g,b)
6 Features
Gray Scale Conversion
Gaussian Blur
Otsu's Thresholding
Dilated Image
Dilation
Erosion
Closing
Morphological Transformation
Bitwise and Operation with each layer of original image and closing image
Area and Perimeter of leaf image
2 Features
Merge
Gray Level Co-Occurence Matrix (GLCM)
Gray Scale Conversion
6- Features
Classification Model
Contrast, Dissimilarity, Energy Homogeneity, Correlation, Angular Second Moment(ASM)
Each Attribute has 4 values --------> 24 Features
Accuracy

Fig1: Flow chart of the model

5.3. Experiment and Results

The data set link is provided by Pranesh Shridhar Kulkarni through Research Gate.

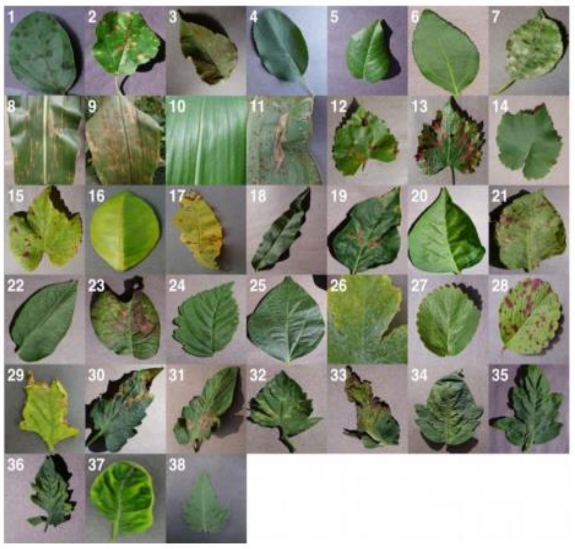

Fig2: Sample Set of Leaf Images

5.4. Performance metric

Using Standard scalar as a normalization technique for x_train and x_test by taking 25 classes of images:

Model	Accuracy	Precision	F1-Score	Recall
SVM	74.57	69.67	63.69	62.20
KNN	70.79	63.20	60.91	60.06
Decision Tree	56.97	47.81	45.69	45.64
Random Forest	71.92	71.08	61.79	60.07
Logistic Regression	69.49	64.38	61.04	60.07
MLP	74.85	66.15	63.57	62.75

5.5. HARDWARE AND SOFTWARE SPECIFICATIONS

5.5.1. DEVICE SPECIFICATIONS

Device name	LAPTOP-4PILMLK5
Processor	Intel(R) Core (TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz
Installed RAM	8.00 GB (7.70 GB usable)
Device ID	E21CFED6-17A3-4A07-86F3-12BCD157B100
Product ID	00325-81463-21784-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	Touch support with 10 touch points

5.5.2. Windows Specifications

Edition	Windows 10 Home
Version	21H2
Installed on	4/5/2021
OS build	19044.1645
Experience	Windows Feature Experience Pack 120.2212.4170.0

6. CONCLUSION

We have used different machine learning algorithms like KNN, Logistic Regression, Naïve Bayes, Decision tree, Random Forest, Support vector machine and multi-layer perception among all the above 7 classifiers MLP and SVM gave best performances so, our preferred model for detecting the plant leaf diseases is MLP or SVM.

7. REFERENCES

[1] Sharath D M, Akhilesh, S.A.Kumar, Rohan M G, Prathap C (2019, April 4). Image based plant disease detection in pomegranate plants for bacterial blight: Semantic scholar. https://www.semanticscholar.org/paper/Image-based-Plant-Disease-Detection-in-Pomegranate-SharathD-Akhilesh/6133b81b82026506eb889dc3d6c33784a0aa6171

[2] Sandesh Raut, Amit Fulsunge(2017). 'Plant Disease Detection in Image processing using MATLAB': Semantic scholar.

https://www.semanticscholar.org/paper/Plant-Disease-Detection-in-Image-Processing-Using-Raut-Fulsunge/77eb4a0c5b5b9d470ebde1172fa74f62010c8829

[3] Arti N. Rathod, Bhavesh A. Tanawala, Vatsal H. Shah(2016,April). ‘Leaf Disease Detection Using Image Processing and Neural Network’: journal 4 research.

http://www.journal4research.org/articles/J4RV2I2033.pdf

[4] K.Gowthami, M.Pratyusha, B.Somasekhar and B.Hemanth(2018, March 9). ‘Detection of Diseases in Different Plants Using Digital Image Processing’: https://www.ijartet.com/

https://www.ijcsmc.com/docs/papers/February2021/V10I2202106.pdf

[5] Prof. Shripad S.Veling, Mr.Rohit S.Kalelkar(2019). ‘Mango Disease Detection by using Image Processing’: https://www.ijartet.com/

https://1library.net/document/q7wjepnz-mango-disease-detection-by-using-image

processing.html

[6] Ms. Kiran R. Gavhale, Prof. Ujwalla Gawande and Mr.Kamal O.Hajari(2014). 'Unhealthy Region of Citrus Leaf Detection Using Image Processing Techniques' : IEEE International Conference for Convergence of Technology.

https://www.academia.edu/33501445/Review_On_Leaf_Disease_Detection_Using_Image_Processing_Techniques

[7] Zarreen Naowal Reza, Faiza Nuzhat, Nuzhat Ashraf Mahsa, Md. Haider Ali(2016). 'Detecting Jute Plant Disease Using Image Processing and Machine Learning': International Conference on Electrical Engineering and Information Communication Technology, IEEE-Semantic scholar.

https://www.semanticscholar.org/paper/Detecting-jute-plant-disease-using-image-processing-Reza-Nuzhat/f55b0267c23a43b32b9bdebd75afff7f0db285c1

[8] Ch.Usha Kumari, S.Jeevan Prasad, G.Mounika(2019). 'Leaf Disease Detection: Feature Extraction with k-means clustering and classification with ANN': International Conference on computing methodologies and communication, IEEE.

https://ieeexplore.ieee.org/document/8819750