AYURVEDIC MEDICINAL PLANT IDENTIFICATION SYSTEM USING EMBEDDED IMAGE PROCESSING TECHNIQUES

A PROJECT REPORT

Submitted by

CB.EN.U4ECE18106 ARNAB DAS
CB.EN.U4ECE18108 B SIVA SAI KUMAR
CB.EN.U4ECE18150 S SHIVA SHANKAR
CB.EN.U4ECE18151 S NAVEEN REDDY

Under the guidance of

Mr. PEEYUSH K P

in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

AMRITA SCHOOL OF ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE 641112

May 2022

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF ENGINEERING, COIMBATORE, 641112

BONAFIDE CERTIFICATE

This is to certify that the project report entitled "AYURVEDIC MEDICINAL PLANT IDENTIFICATION SYSTEM USING EMBEDDED IMAGE PROCESSING TECHNIQUES",

submitted by

CB.EN.U4ECE18106 ARNAB DAS
CB.EN.U4ECE18108 B SIVA SAI KUMAR
CB.EN.U4ECE18150 S SHIVA SHANKAR
CB.EN.U4ECE18151 S NAVEEN REDDY

in partial fulfilment of the requirements for the award of the Degree of **Bachelor of Technology** in **ELECTRONICS AND COMMUNICATION ENGINEERING,** is a Bonafide record of the work carried out under my guidance and supervision at the Amrita School of Engineering, Coimbatore.

Project Advisor
Mr. Peeyush K P
Assistant Professor (Sr. Gr.)

Project Coordinator
Mr. Harikumar M. E
Assistant Professor

Chairperson ECE
Dr. M. Jayakumar

The project was evaluated by us on:

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

We would like to express our heartfelt gratitude to our mentor, Mr. Peeyush K P, Assistant Professor (Sr. Gr.), Department of Electronics and Communication Engineering, for his invaluable advice and unwavering support during this project. We would also want to thank our panel members, Mr. Ganesan M, Assistant Professor, Department of Electronics and Communication Engineering, and Mr. Harikumar M E, Assistant Professor, Department of Electronics and Communication Engineering, for providing us with helpful input and aid in understanding the project's shortcomings.

ABSTRACT

Medicinal plants are a rich resource for drug development and have a huge impact in ayurvedic medicine preparation. Plants can be classified based on various classification methods such as cell, genetic and serum etc. It's difficult for an individual to explore the various classification methods and it's practically not feasible as it demands good knowledge in plant taxonomy and long-term time investment. Due to the shortage of experienced and qualified taxonomists in identification and classification of medicinal plants, this gap can be bridged with the help of various image processing techniques and computer vision. The knowledge of medicinal plants that has been passed down through the years must be conserved and protected. Computer vision, pattern classification, and image analysis technologies are promise for identifying and categorizing medicinal plants.

This work proposes a Deep Learning (DL) based Convolutional Neural Network (CNN)model, to classify medicinal plants using leaf features. We have acquired a segmented medicinal plants dataset, commonly seen in various regions of India. The extracted dataset contains leaf samples from 30 medicinal plants. A Sequential Neural Network is developed considering Alexnet, VGG16 and VGG19 as reference and is used for the efficient identification and classification of the classes. Finally, the classification is performed using Softmax.

And also we performed Classification using Random Forest Classifier (RFC) and LGBM using Gabor, Sobel filters and GLCM & Shannon for feature extraction and made a comparison between the different combinations. The dataset is divided such that 80% is used for training and 20% for testing.

The main objective is to develop a Deep Learning and ML based model to identify and classify plants based on various features, which is done with the help of various feature extractors and to develop a standalone device that clicks a picture and identifies the medicinal plant.

TABLE OF CONTENTS

LIST OF SYMBOLS

SYMBOL	DESCRIPTION	PAGE NO
θ	Standard deviation of the gaussian envelope	15
λ	Wavelength of the sine component	15
Θ	orientation of filter	15
Ψ	Phase offset	15
γ	Spatial aspect ratio	15

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION	PAGE NO
DL	Deep Learning	ii
CNN	Convolutional Neural Network	2
FC	Fully Connected	2
NN	Neural Network	3
ReLU	Rectified Linear Unit	4
ML	Machine Learning	6
KNN	K-Nearest Neighbour	9
SVM	Support Vector Machine	9
ANN	Artificial Neural Network	9
MNN	Medicinal Neural Network	10
GLCM	Gray level Co-occurrence Matrix	11
GLDM	Gray Level Difference Method	11
RFC	Random Forest Classifier	17
LGBM	Light Gradient Boosting Machine	17
NC	Normalized Convolution	21
USB	Universal Serial Bus	24
HD	High Definition	24
OS	Operating System	24
NLM	Non-Local Means filter	26
CLAHE	Contrast Limited Adaptive Histogram Equalization	26
GUI	Graphical User Interface	27

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

Plants play an essential role in the preservation of life and biodiversity on Earth by allowing the flow of air and water for all living things. One of the most significant classes of plants is medicinal herbs, that were used to treat various diseases. The knowledge of medicinal plants that has been passed down through the years must be conserved and protected. Computer vision, pattern classification, and image analysis technologies are promise for identifying and categorizing medicinal plants. One of the most difficult jobs is identifying a medical plant with the needed therapeutic properties. Even if herbal therapy has no negative effects, a patient's life might be lost if a medicinal plant is misidentified. As a result, at this point in time, a completely automated method to accurately recognize medicinal plants is inevitable.

For production of ayurvedic medications, the identification and categorization of medicinal plants is critical. Appropriate classification of medicinal plants benefits horticulturists, botanists, ayurvedic medical therapists, forest service authorities, as well as those engaged in the making of ayurvedic pharmaceuticals. However, a critical shortage of skilled taxonomists exists in this field. There is a growing segment of the population who prefers ayurvedic medicine over other medications. Taxonomists employ leaf, flower, trunk, and branch characteristics to classify based on their respective features. The leaves are the greatest choice for plant categorization.

1.1 Convolutional Neural Network (CNN):

Convolutional Neural Networks are a type of Machine Learning Network that are often used to solve problems related to classification. The connections between the layers of the CNN i.e., nodes and neurons depict the connections of neurons and synapses in the brain. There are series of interconnected layers in the network such as convolution, activation functions, max pooling, average pooling and fully connected (FC) layers. For an input image, network applies a series of convolutional filters of various sizes which varies from one model to other model in order to obtain the model's learning parameters. Max and average pooling layers are applied between convolutional layers to reduce the number of attributes utilized for learning and the computation. Softmax is a final layer which is used for multiclassification purpose.

1.1.1 Neural Network (NN):

A network of neurons is referred to as a Neural Network. Each input layer (nodes) will include some values (features), which will be multiplied by the weights.

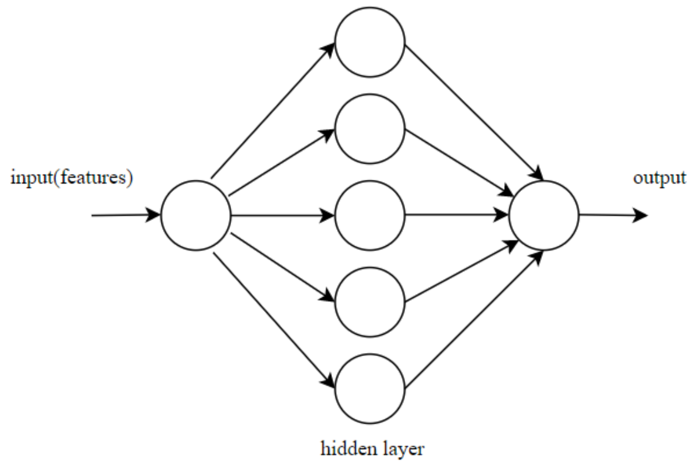


Fig-1.1-Single Layered Neural Network

1.1.2 Convolution layer:

The convolution layer is the initial layer of CNN. Convolution in CNN is mostly used to extract features from the input image. The mathematical procedure of convolution is to mix two sources of data. To build a feature map, convolution is performed to the input data using a convolution filter.

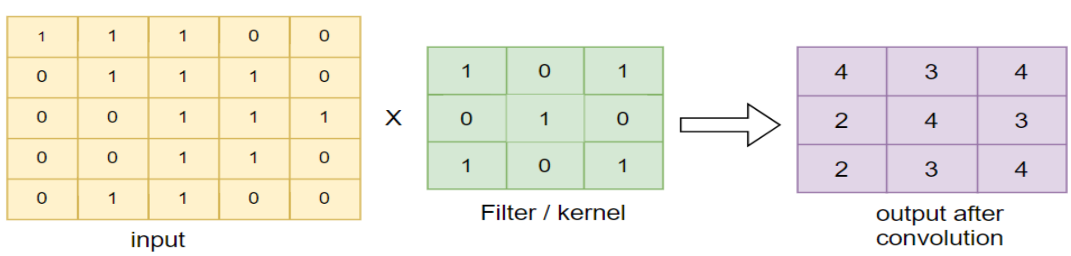


Fig-1.2-Convolution in Neural network

The input to the convolution layer is on the left side of the above fig.1.2, for example, the input image. The 3x3 convolution filter, also known as a kernel, is located on the right side. Convolution is achieved by sliding this filter across the input. Then, at each position, conduct element-by-element matrix multiplication (dot product). The aggregate of the results is entered into the feature map.

1.1.3 Activation Functions:

The activation function comes after the convolution layer. The rectified linear unit (ReLU) is a popular activation function in deep learning applications since it is much faster to calculate than alternatives like the sigmoid function while still delivering decent results.

ReLU function is defined as,

$$f(x) = \begin{cases} x, if\ x > 0 \\ 0, otherwise \end{cases}$$

1.1.4 Pooling Layers:

To minimize dimensionality, pooling was used. It decreases training time and combats overfitting by reducing the number of parameters. While maintaining vital information, pooling layers down sample on each feature map independently. There are three types: min pooling, max pooling, and average pooling.

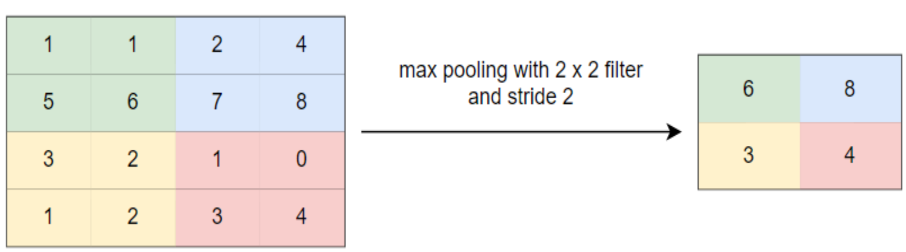


Fig-1.3-Implementation of max pooling

The maximum value in the pooling window is obtained by sliding a window across its input in this case. Two hyperparameters on which pooling depends is: Filter(F) and Stride (S). The primary goal of a pooling layer is to reduce the number of parameters in the input tensor, which helps to reduce overfitting and improves performance.

1.1.5 Fully Connected Layers:

Fully Connected Layer is connected by a series of nodes (neurons). Fully connected layer is added after the convolution and pooling layers. Weights, biases and neurons are all found in this layer. It will link neurons of one layer to neurons of another layer. Because a fully connected layer expects a 1D vector of numbers, the final pooling layer's result is flattened to a vector, which becomes the fully connected layer's input.

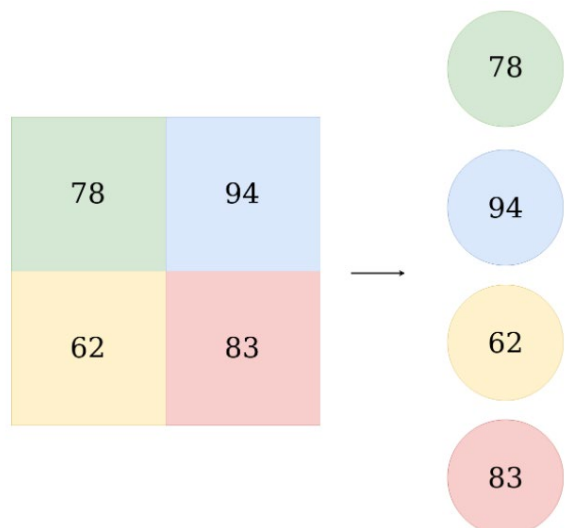


Fig-1.4-flattening

1.1.6 Softmax Layer:

Last layer of the model is softmax layer. Softmax is used for multiclassification problems. Softmax layer is connected to final fully connected layer and returns the projected probability of each group. Softmax functions are sigmoids with several classes. They were used to calculate the probability of numerous classes at the same time.

1.1.7 Dropout:

Dropout is a characteristic utilized in several CNN models. This may eliminate the problem of overfitting in our model. By eliminating specific connections between the nodes on a random basis. Overfitting and Underfitting are the problems generally occur to eliminate this we use Dropout strategy.

1.2 Machine Learning (ML) & Confusion Matrix:

1.2.1 Tensorflow:

Tensorflow is an open-source dataflow programming library that is extensively used for machine learning. The Google Brain team created Tensorflow, which gives developers access to a set of algorithms that make implementing machine learning applications like image categorization and convolutional neural networks easier.

1.2.2 OpenCV:

OpenCV is a set of functions that allows programmers to manipulate and alter picture data for computer vision applications. OpenCV is used in this project to open the images and then resize them so that the input size is uniform. Due to hardware limitations, the current input image size is 256x256, although it may change based on the requirement.

1.2.3 Numpy:

NumPy is a Python library that includes functions for working with multi- dimensional data, such as image matrices. The NumPy library was used to test the idea of introducing distortions to the image input to improve the CNN and classifier's performance. NumPy is also utilized to divide the pixel values of the input images and scale them.

1.2.4 Confusion Matrix

The confusion matrix is a matrix that is used to assess the performance of classification models for a set of test data. It can only be determined if the actual values for test data are known. It evaluates the performance of classification models when making predictions on test data and indicates how good our classification model is.

1.3 Hardware

Raspberry Pi 3 Model B has been used for the implementation of hardware. It's a Single Board Computer (SBC) since it has an entire computer printed on a single Printed Circuit Board. This hardware has been used because it helps the develops to develop code in C, C++, Python, Java, HTML etc. Due to its cheap in cost and value for money, it has become extremely popular among hobbyists and DIY developers. Raspberry pi has all features of a computer with a CPU, GPU, storage, memory etc. So, it's basically a microprocessor-based minicomputer (SBC). Raspberry Pi needs an OS to run (default OS is Raspberry Pi OS) which helps the developers to install its own preferred IDE to develop code and run in on the board unlike other boards where we need to upload the code to the board every time, we make changes. The clock speed of this board is 1.2MHz which is pretty good for IOT and embedded applications. Raspberry Pi 3 supports Python which is a widely used language for developing codes and has a great database with huge support. The only two disadvantages of this board are, firstly, any power interruptions for Raspberry Pi may cause damage to the hardware and/or software applications but for other microcontrollers it just restarts or reboots the system. So, if disrupting the power in raspberry pi might cause software and/or OS crash. Secondly, Raspberry Pi is not an open-source board. So, you cannot find out its PCB template and make a new board out of it.

CHAPTER 2
LITERATURE SURVEY

Deep learning relates to neural network structure, according to the findings of the research [1], it can learn characteristics from huge data automatically and train and categorize plant leaf samples through the use of a neural network model using a back propagation method. Leaf form, color, and texture are the most important factors in plant categorization. Leaf texture veins have more consistent properties than color and form. Author made a comparative analysis between different classification algorithms such as K-Nearest Neighbour (KNN), Support Vector Machine (SVM), Decision Tree and Artificial Neural Network (ANN). From the analysis, author came to conclusion that ANN with backpropagation error algorithm (BP algorithm) produced good results compared to others.

A plant leaf identification system based on morphological characteristics and adaptive boosting methods is provided in this study [2]. They employed k-NN, decision trees, and multilayer perceptrons, among other categorization algorithms. They used a public dataset from Flavia. The accuracy rate has been improved with adaboost approach compared to conventional approach which resulted in around 95%.

Ayurleaf [3], a CNN-based methodology for classifying medicinal plants, has been presented using features such as shape, texture and size etc. They had their own dataset collected which consist of species of around 40 classes. Neural network with deep learning based on Alexnet is used for the employment to extract the features of the flora. To classify the data, SVM classifiers & SoftMax were utilized.

The author compared well-known models VGG16 and VGG19 and presented a complete analysis of deep learning-based Convolution Neural Networks and the different layers associated in the implemented model. They created [4] their own dataset, which includes 64 medicinal plant species and 1000 samples from each. Both the implemented models are same in the functionality but differs in the number of convolutional layers.

The major topic of the paper [5] is image segmentation in a complex background. A vein is a vascular bundle that extends from the leaf's centre to the leaf's periphery in general. The crux of an accurate and entirely automatic segmentation system for medicinal plant leaf images in difficult backdrops is proposed in this study [5].

The OTSU method (OTSU) [5] is a threshold-based image segmentation algorithm. The threshold selection rule in this approach is the largest inter class variation between the background and the foreground image it divides based on the gray scale features. Contrast between two halves is the greatest when the best threshold is opted. The variance value, which is the difference between the two parts of the graph, is an essential uniform gray distribution statistic; the larger the variance value, the larger the difference between the two parts of the graph. The higher degree of segmentation precision.

Batch normalisation, max pooling, average pooling, fully connected layers, softmax activation functions, and data augmentation are among the [6] various layers in CNN listed by the authors. They developed a model [6] that used batch normalisation between layers and was based on VGG-16 as a reference. Batch normalization reduces the number of training cycles required, yet the network produces the same outcomes as the network without normalising.

Authors [7] described many existing approaches for classifying medicinal leaves proposed by various authors. The suggested research [7] will primarily focus on detecting plants based on their therapeutic properties. Their MNN (Medicinal Neural Networks) model is trained using data that they manually collected. The dataset contains four (4) different medicinal plant variants. Furthermore, all of the images are split by concealing the backdrop information with a white screen. They used the Confusion Matrix to evaluate the metrics in their model.

For recognizing contoured borders of overlapped leaves from a complicated backdrop, the working procedures outlined in this study [8] are used. The proposed approach divides the primary picture into leaf regions by segmenting the specified contour areas. The model attained a segmentation rate of 90.46 percent overall, with segmentation rates of 95.34 percent for single leaves and 86.73 percent for overlapping leaves, respectively, according to the experimental data.

The CNN [9] is a form of neural network that is extensively employed in image processing. Convolution, Max-pool, and fully linked layers combine to form CNN. The author uses CNN to construct a hardware device that can distinguish between plastic and non-plastic with a 97.8% accuracy.

According to CNN, a machine [10] has been created that can determine if a plant leaf is healthy or unhealthy. For processing, the author employed a Linux-based machine with a CUDA-enabled GPU, Darknet for image processing and recognition, and Python to interface the program with the darknet.

The key processes in identifying the leaf are feature extraction and classification. The author [11] utilized histogram and texture to extract features and a Support Vector Machine to classify them. They employed five distinct classes, each including four different leaf sizes. Following the categorization, the output forecasts which class the leaf corresponds to.

The database is the most significant element of image processing in general. The author [12] analyses three distinct methods of feature extractions, including Wavelet transformations, Gray Level Co-occurrence matrix (GLCM), and Gray Level Difference method (GLDM), using varied training and validation percentages. The classifier utilized is KNN. Finally, they came to the conclusion that Wavelet transformations had the highest accuracy

CHAPTER 3
METHODOLOGY

Feature extraction is the most important aspect of classification/identification. Image processing, picture feature extraction, and image classification are the three phases that make up the recognition process.

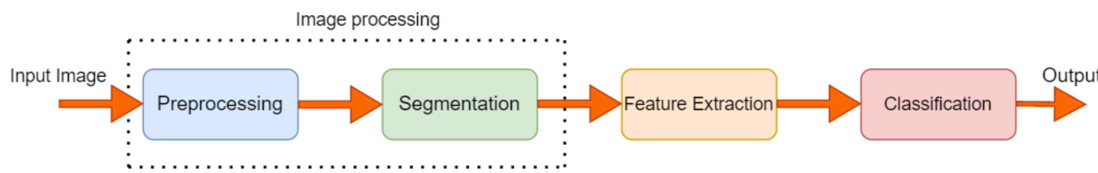


Fig-3.1-Classification Process

Image preprocessing and segmentation techniques are the two key processes in the image recognition process. Image restoration and transformation are two of the most used image preprocessing techniques. Its major goal is to reduce picture interference and noise, increase image usable information, and improve object detection. The purpose of segmentation is to divide a recognized image into subregion.

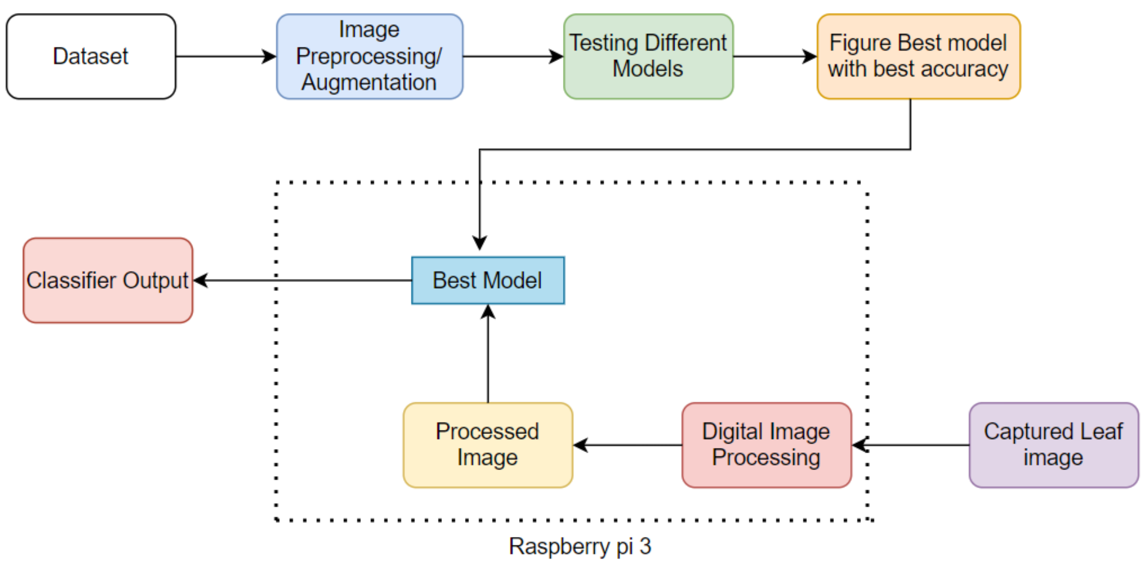


Fig-3.2-Methodology

3.1 Deep Learning Model (CNN):

A CNN sequential model is implemented by considering VGG16 and Alexnet as reference. The input layer is the initial layer, and it defines the size of the input pictures. The convolution layer uses 32 (3 x 3) filters with a stride size of one. The output is then thresholded by a ReLU layer, which is accompanied by the max – pooling layer by a 2x2 filter. The previous layer precisely halves the exact size of its return. A secondary convolution layer follows this max pooling layer in which a 32 kernel is performed with a dimension of 3x3. ReLU layered will be followed, accompanied with the layer of max pool with 2x2 of size of the filter by a stride value of 1. Now the next two layers are back-to-back convolution layers with following configurations. Both of them employ 3 x 3 kernels by a stride value of 1, with 64 & 250 kernels in each. Following these layers, a ReLU layer is added. The next layer is a convolution layer having 128 (3x3) kernels, after that a ReLU activation, an averaging pool layer by a value of (2,2) size. The next levels follow a similar pattern, with 64 (3x3) filters, after that a ReLU activation function, then averaging pool layers by a value of (2,2) size Last Convolution layer is of size 256 (2x2) and followed by activation of the ReLU function and max pool layers with a value of (2x2) size. Final value of the last max-pool layers is transferred to fully connected layers which have 6400-neurons. And it is again processed through another fully connected layer of size 30 as our dataset consists of 30 classes. Final classification of the model is done with the help of SoftMax which is used for multi-classification purposes.

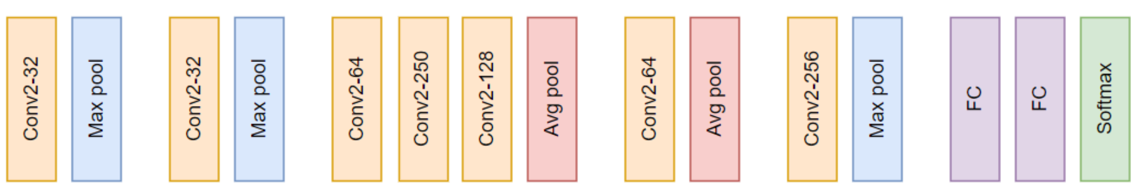


Fig-3.3-Implemented CNN model

3.2 Machine Learning Model:

3.2.1 Gabor Filter:

Gabor is just a convolutional filter that comprises a gaussian and sinusoidal term together. The weights are given by the gaussian component, whereas the directionality is given by the sine component.

Gabor is used to provide texture and edge features. Gabor kernel actually mimics the visual cortex meaning, the way we recognize textures with our eyes can be simulated using Gabor kernel.

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right)$$

λ – Wavelength of the sine component

Θ - Orientation of filter

Ψ- Phase offset

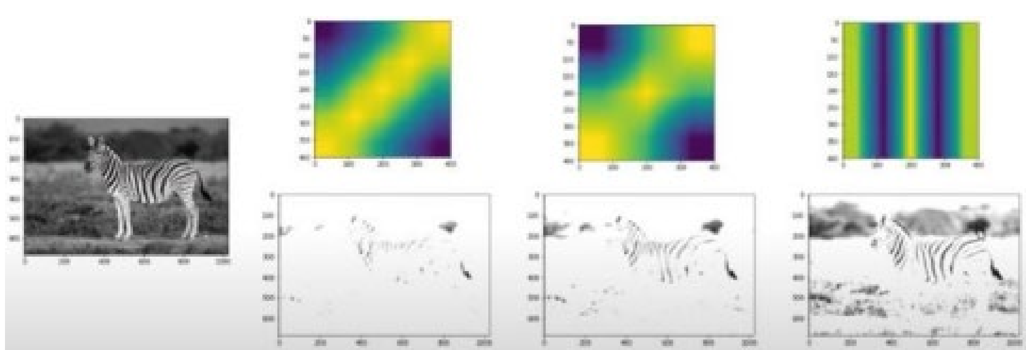
θ - Std. dev. Of the Gaussian envelop

γ – Spatial aspect ratio

where,

$$X' = xcos\theta + ysin\theta \text{ and } Y' = -xsin\theta + ycos\theta$$

For example, if there is any change in the parameters like lambda, theta, phase offset and standard deviation of the Gaussian, we can generate an ideally infinite number of filters.



.

Fig-3.4-Different orientations of Gabor filter

By observing above fig 3.4, one can infer that by changing the parameters of the filter it generates filters with various orientations through which we can extract features from the original image. Usually one filter is not enough, so nearly 12 Gabor filters were generated and convoluted with the image and then fed implemented ML model.

3.2.2 GLCM – Gray Level Co-occurrence Matrices:

Features like aspect ratio, roundness, perimeter, area, length, breadth [2] give better results, but in most of the cases, these features are not enough or efficient enough to classify the leaves. There are chances that two different leaves might have the same length and breadth. But to an extent it's sure that the texture of two different kinds of leaves will be different. The textures determine the patterns that appear repeatedly and are regarded as local changes in picture intensity. In this paper, a statistical approach like the use of GLCM is being suggested [11]. GLCM uses second order statistics for features that can be used to infer the degree of correlation between pairs of pixels. Uses pairs of pixels where the user can define the distance and angle between the pixels. It is recommended to extract GLCM for multiple distances and angles between pixels. A GLCM matrix will be formed for the computation of,

- Contrast: $\sum_{i,j=0}^{levels-1} P_{i,j}(i-j)^2$
- Dissimilarity: $\sum_{i,j=0}^{levels-1} P_{i,j}\,|i-j|$
- Homogeneity: $\sum_{i,j=0}^{levels-1} \frac{P_{i,j}}{1+(i-j)^2}$
- ASM: $\sum_{i,j=0}^{levels-1} {P_{i,j}}^2$
- Energy: $\sqrt{ASM}$
- Correlation: $\sum_{i,j=0}^{levels-1} P_{i,j}\left[\frac{(i-\mu_i)(j-\mu_j)}{\sqrt{({\sigma_i}^2)({\sigma_j}^2)}}\right]$

3.2.3 Random Forest Classifier (RFC):

Random Forest Classifiers and SVMs are the mostly used traditional ML models. As the name says, Random Forest means it has a number of decision trees. Random Forest Classifier randomly picks the attributes or features that are generated previously, the Gabor filter banks among a selection or sub selection of all the attributes or features. RFC uses Gini impurity to pick a node to split that yields maximum information gain. This is used to take the decision of the tree.

A Random Forest is made up of a huge set of discrete trees that work together as a unit and each tree offers a forecast, and the model's prediction is the group with the most votes.

This classifier has been chosen since it helps to minimize overfitting (works great on training data but not on new data). Bootstrap / bagging in Random Forest allows each tree to randomly sample from the dataset. Feature randomness is introduced by only allowing a subset of available features at each node.

In the implemented model we have used 40 n_estimators (decision trees) and the image size which is given to implemented model is 256x256 which is resized from original image which is of size 1600x1200. Features such as Gabor filter banks, original pixel values and Sobel filter are used to train model. In the model, bootstrapping of sample technique is also implemented to minimize overfitting.

3.2.4 LGBM – Light Gradient Boosting Machine:

In this paper, LBGM is also used as a classifier which works great with the GLCM feature extractor. It's a fast, distributed, low memory usage, high-performance gradient boosting framework based on a decision tree algorithm. It splits the tree leaf wise, unlike other models do it tree wise or level wise. Accuracy of the model depends on the parameters we provide. Main parameters on which models depends are,

- Max depth: describes maximum depth of a tree
- Min data in leaf: describes about minimum number of the records a leaf may have
- Bagging fraction: used to speed up the training and avoid overfitting

- Lambda: specifies regularization

3.2.5 Dataset:

When it comes to object categorization, the dataset is crucial. Segmented leaves dataset is obtained from Kaggle which consists of leaves of 30 classes of which each class has an average of 70 images of size 1600x1200.

No of Classes	scientific name of plant	Common name of plant
1	Alpinia Galanga	Rasna
2	Amaranthus Viridis	Arive-Dantu
3	Artocarpus Heterophyllus	Jackfruit
4	Azadirachta Indica	Neem
5	Basella Alba	Basale
6	Brassica Juncea	Indian Mustard
7	Carissa Carandas	Karanda
8	Citrus Limon	Lemon
9	Ficus Auriculata	Roxburgh fig
10	Ficus Religiosa	Peepal Tree
11	Hibiscus Rosa-sinensis Jasminum	Jasmine
12	Mangifera Indica	Mango
13	Mentha	Mint
14	Moringa Oleifera	Drumstick
15	Muntingia Calabura	Jamaica Cherry-Gasagase
16	Murraya Koenigii	Curry
17	Nerium Oleander	Oleander
18	Nyctanthes Arbor-tristis	Parijata
19	Ocimum Tenuiflorum	Tulsi
20	Piper Betle	Betel

21	Plectranthus Amboinicus	Mexican Mint
22	Pongamia Pinnata	Indian Beech
23	Psidium Guajava	Guava
24	Punica Granatum	Pomegranate
25	Santalum Album	Sandalwood
26	Syzygium Cumini	Jamun
27	Syzygium Jambos	Rose Apple
28	Tabernaemontana Divaricata	Crape Jasmine
29	Trigonella Foenum-graecum	Fenugreek
30	Hibiscus Rosa-sinensis	Chinese Hibiscus

Table 3.1- Data set Classes

Data augmentation is a strategy for extending the quantity of data available by generating new data from current data. It allows us to improve the dataset's size and variety without having to acquire fresh information.

The following data enhancements were performed: Rotation of the images, Horizontal flip, zoom, width and height shifting. For data augmentation, Gaussian noise is injected into the picture. The process of blurring a picture, entails averaging nearby pixels. This blurs the image and lowers the amount of information. The dataset size grows larger, and the model's training improves. To enhance the precision of a deep learning-based model, data augmentation technique is applied. The training and testing dataset are randomly split from the dataset. The train set accounts for 80% of the dataset, whereas the testing set accounts for 20%.

CHAPTER 4
IMAGE SEGMENTATION

Because leaves are most typically observed in groups with natural backgrounds, leaf identification and segmentation is a challenging task. Because of their similar colors, the edges of leaves are difficult to see from the photograph. It's also more difficult to separate each individual leaf, especially overlapping ones, because leaves are almost identical in color, texture, and shape.

Leaf segmentation entails two steps:

i) Separating the main leaf region from the natural backdrop,

ii) Extracting individual leaves and covered or overlapped (i.e., objects on object) leaves from the picture.

A contour selection-based leaf segmentation technique is used for segmentation.

Input Image:

The source pic was taken in blue, green, red colors form instead of Red, Green, Blue since we used the Opencv to generate the model

Preserve Edges:

First contours or outlines of each leaf should be identified before segmenting the connected regions inside them, as our main goal is to identify the contour or edges of each leaf. Internal texture is smoothed and boundary edges are preserved using the Edge Preserving Filter at this step.

Stylize Leaf Boundary:

To make every item's edge, shape, or contour sleek and crisp at the same time, we employ a Stylization Filter. In comparison to traditional outline sharpening filters, it uses the Normalized Conv (NC) filter to increase accuracy and speed. Internal texture has no effect on it as it hardens all boundary edges over time.

Multi-Channel to Single-Channel:

The data in BGR format (3 channels) is then processed to a gray level (1 channel). In comparison to BGR images, only one third of the picture data has to be processed in grayscale images, which greatly minimizes the amount of computation and memory usage.

Expand Boundary Edges:

This process enlarges the outline borders of leaves by expanding the areas of bright regions. This processing phase also aids in the separation of overlapping edges. This is done using a Sobel filter.

Smooth Edges:

When the dilatation procedure expands the bright borders, this may enlarge remaining portions of the leaf's internal texture edges. To solve the issue, a smoothing operation is performed after dilatation.

Separate Background and Foreground:

Pixels with single intensity is used to differentiate the pixels in the foreground first from pixels in the backdrop.

Contour Detection:

Identification of contour is done by, recognizing the contours of the connected regions. OpenCV Library has an inbuilt function through detection is possible.

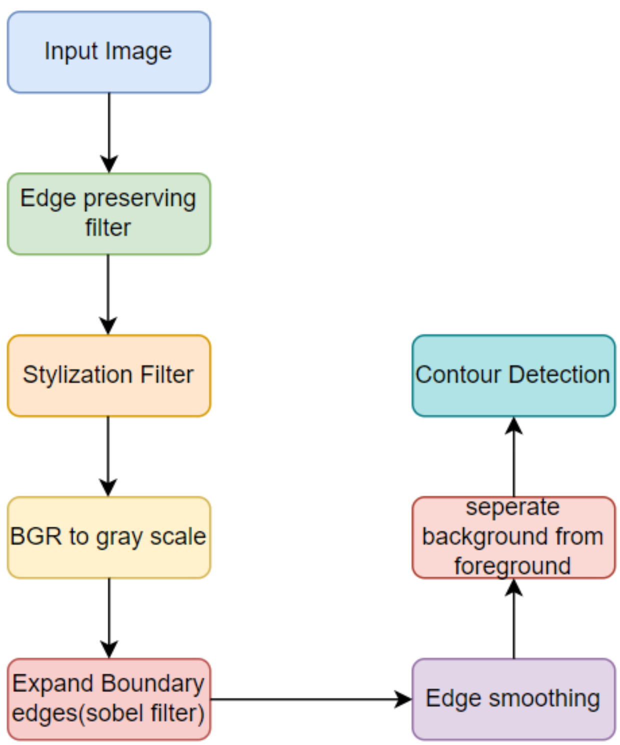


Fig-4.1-Segmentation Flow

CHAPTER 5
HARDWARE AND UI

For the hardware implementation of this project, we have chosen Raspberry Pi 3 Model B V1.2 which is one of the cheapest mini-Computers which you can buy in the market. It’s so small, that its size is comparable to a credit card. It has Broadcomm BCM2837 chipset with a 1.2GHz quad-core 64-bit ARM cortex A53 CPU with 1 Ethernet and 4 USB ports. In this project, one of the USB ports will be connected Webcam which has a specification of 0.9MP with 720p video quality. Raspberry Pi 3 has Debian OS which is a Linux distribution, so it generally works and can be treated like a normal Linux machine.

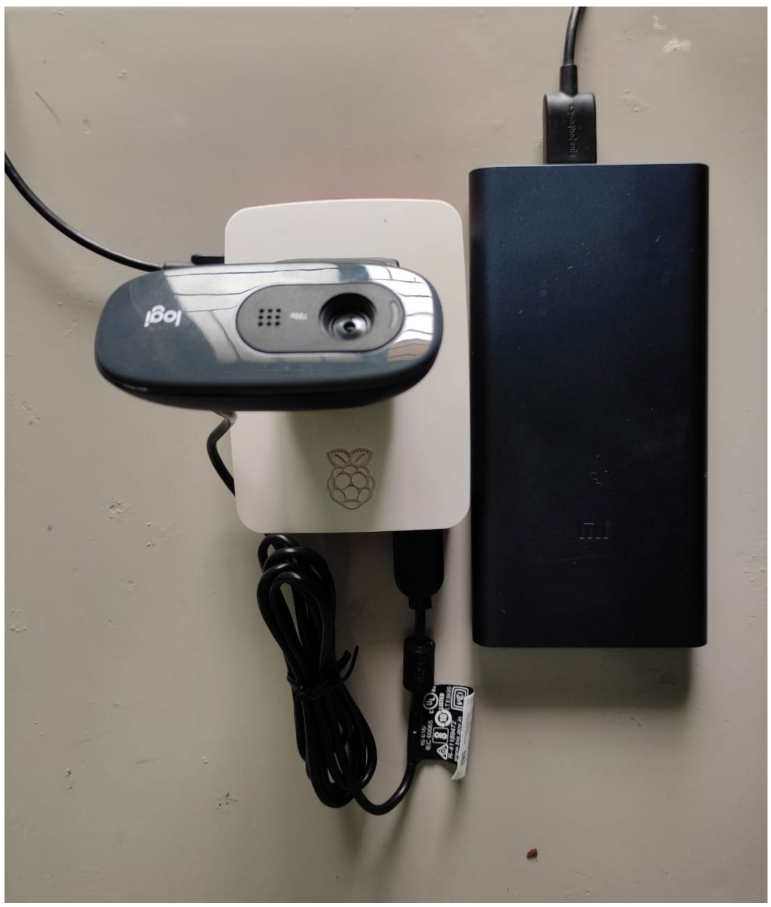


Fig-5.1-Hardware Implementation

It is very essential to have a prototype before going into deployment of large scale. As we can have many advantages like the discovery of design problems, estimating the cost of production and maintenance and identification of improvements. The apparatus used in the implementation of the prototype are Raspberry Pi 3 Model B V1.2 which will be interfaced with Logitech webcam.

The Raspberry PI is used to embed the model developed for the identification of the ayurvedic medicinal plants which plays the crucial role in the prototype. And the Logitech webcam is used to capture the leaf from the surroundings and pass it to the model for the prediction via implemented user interface.

The prototype deployment starts with the webcam as it is used to capture the images of the leaf. In the process of implementing the Logitech webcam is used as it has the following specifications like HD quality, 720 frames per second, short focus, supporting all the platforms like Windows OS, Mac OS, Linux and all other famous platform.

From the webcam the real time data, that is the live feed can be seen on the screen of the system(pc) to capture the object. Here webcam is placed above the leaf and is focused to capture a clear image of the plant/leaf for the further process. In order to capture the image from the live feed we need to press 'q' to capture the image and save it in the system in order to further process the captured image to predict.

While capturing the image of the leaf from the webcam make sure that the cap of the webcam is in open state. Otherwise the system throws an error or the live feed from the webcam will be in a blank out state.

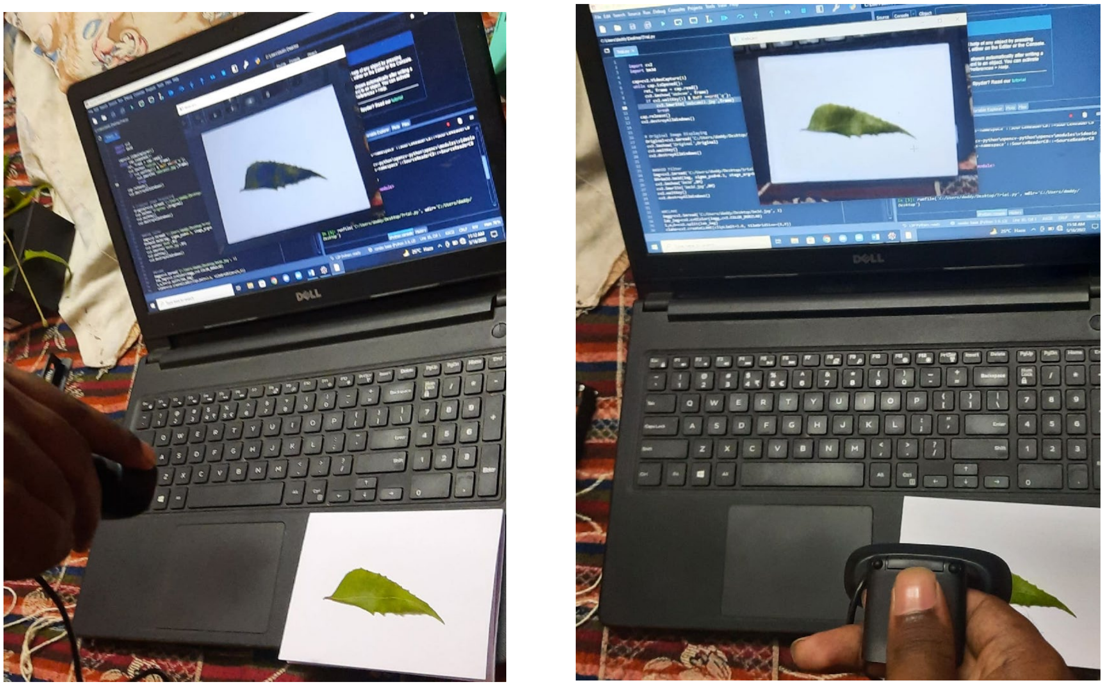


Fig-5.2- Capturing image using Webcam

After successful completion of capturing the image from the webcam it should be processed in order to make sure that the image is perfect so that the prediction part doesn't fail. Before transferring the image to the model few digital Image processing techniques has to be performed for good visualization and better quality of the captured images and the process is called as Pre-processing.

The first step in the implementation of the Pre-processing is the noise removal of the image or it can be called as denoising. As the implementation is done in real time it might contain small noise which also leads to the imperfections of the results from the prediction and also some details. So the denoising of the input image is done to restore the real image for better prediction of results.

For the implementation of the denoising over captured image a BM3D filter is deployed. BM3D stands for the Block matching and 3D filtering. This denoising filter is very accurate and high precision in preserving of the edges and the denoising ability of BM3D is perfect when compared to the other denoising filters like Gaussian filter, NLM filter and median filter.

Generally Gaussian filter is used for blurring of the image as well as denoising of the image. But, it also blurs the images upto some extent alongside denoising the image so, usage of gaussian filter is discarded. The median filter and NLM filter (Non local means filter) will preserve the edges and denoise the image too. But, on comparing all the filters BM3D filter shows very good results and it’s a powerful approach for denoising. So it's deployed in the denoising of the captured image in the pre-processing part.

The second step in the Pre-processing of the image is histogram equalization. The process of the histogram equalization is to improve the contrast of the image so it can be legible. The process of the histogram equalization is performed by spreading the most frequent intensity values that is the spreading out the intensity range of the image. Implementation of this histogram equalization will lead to high contrast image. So, another method is deployed that is CLAHE.

Generally, CLAHE stands for Contrast Limited Adaptive Histogram Equalization. CLAHE works on the small regions of the image rather than the whole image called tiles. With this as a minor drawback it can be resolved and hence it is deployed in the implementation. So, in this way the pre-processing has been done in the implementation of the hardware prototype. In the next stage processed image is passed to the best model present in the Raspberry Pi.

A simple User Interface is designed which will help the user to navigate and use program/model with ease. This UI has been designed in Python language with the help of Tkinter library. Tkinter is Python's one of the standard Graphical User Interface (GUI) library. Python with Tkinter make it simple and quick to design graphical user interfaces. Tkinter gives the Tk GUI toolkit a robust object-oriented interface.

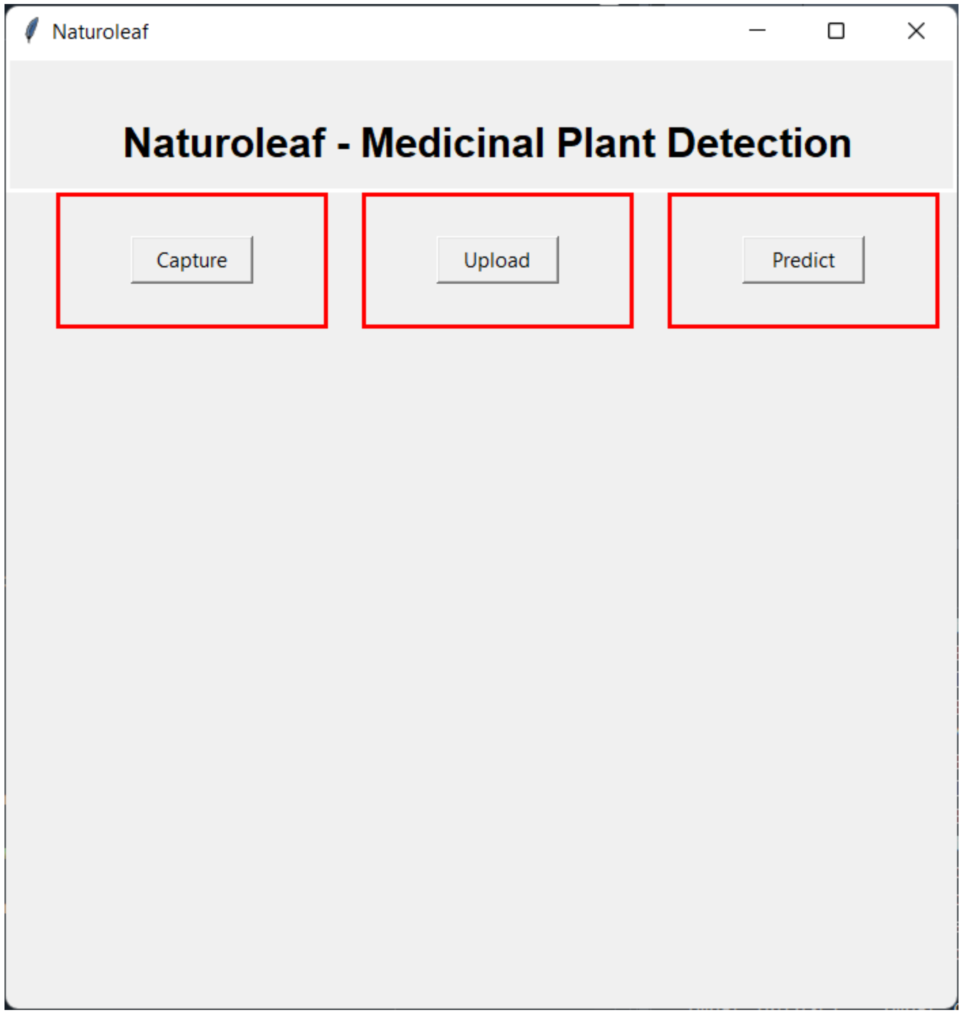


Fig-5.3-Naturoleaf GUI

In the above fig 5.2, one can observe that a simple User Interface (UI) with 3 buttons. The ‘Capture’ button is used to capture an image from the webcam which is been attached to raspberry pi. ‘Upload’ button is used to upload an image from the local machine and use that image to predict. At last, ‘Predict’ button is used to predict the image which has either been captured by the camera or uploaded from the local machine.

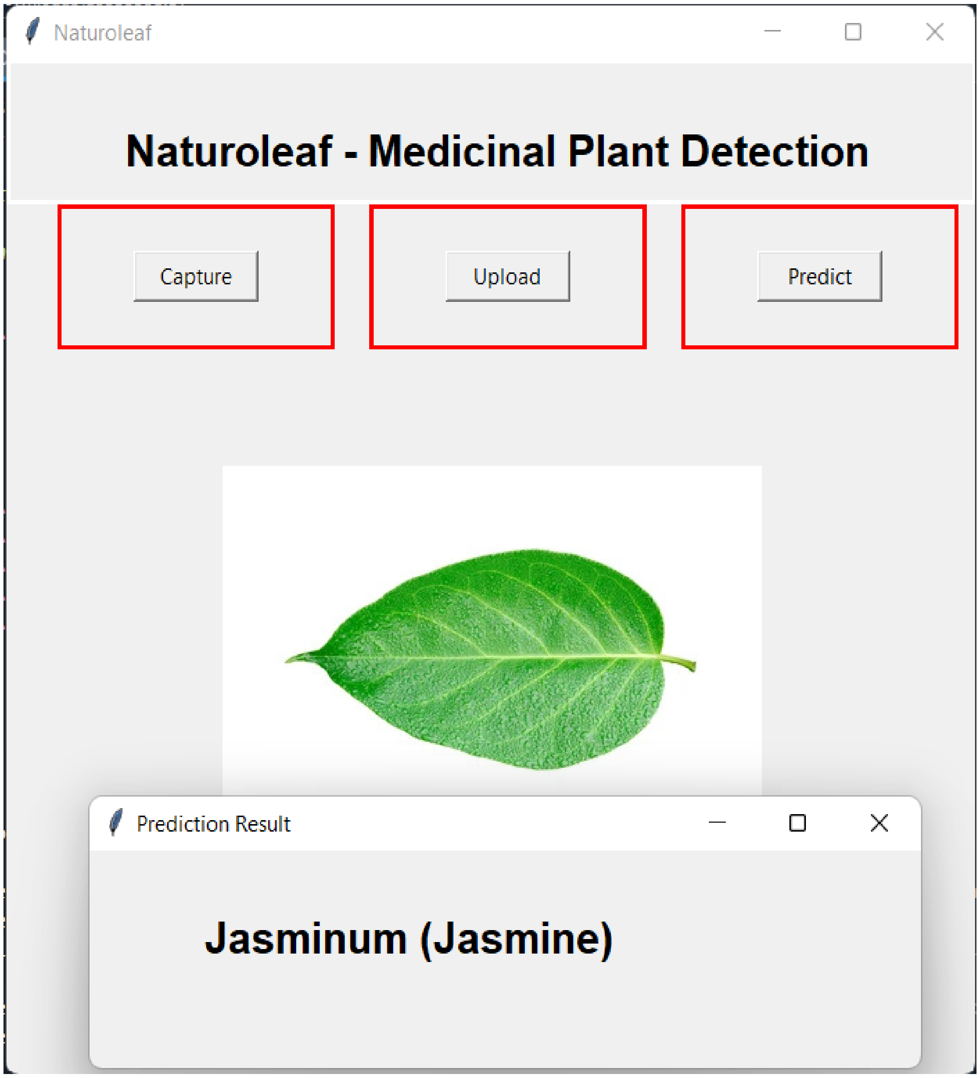


Fig-5.4-UI output from Implemented model

The above picture shows how the UI helps the user to interact with the code and get the desired results when an image is uploaded from the local machine (pc) and interfaced with the model to predict to which class the given input image belongs to. As a result, a window will be popped which describes the scientific and common name of the leaf. In the above fig 5.3 the given input image is classified as, it belongs to jasmine class.

CHAPTER 6
RESULTS

6.1 Deep Learning (CNN) Results:

For the implemented CNN based model which is constructed taking VGG16 and Alexnet as reference consists of 15 layers with various functions. Validation and training accuracy of the model are around 77% and 78% respectively.

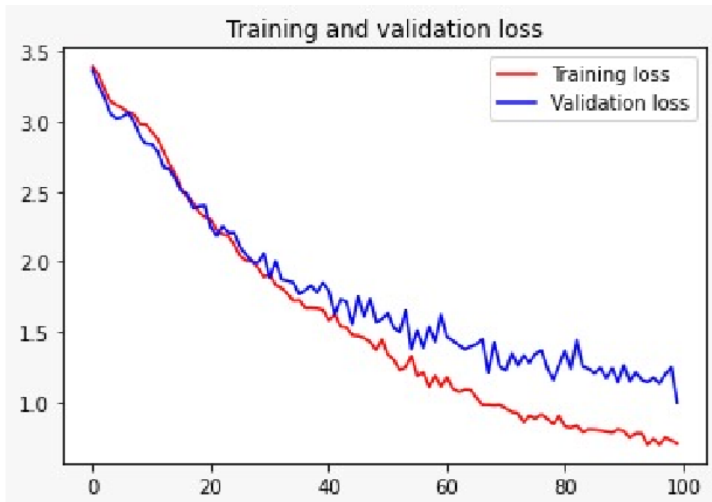


Fig-6.1-Training and Validation Loss

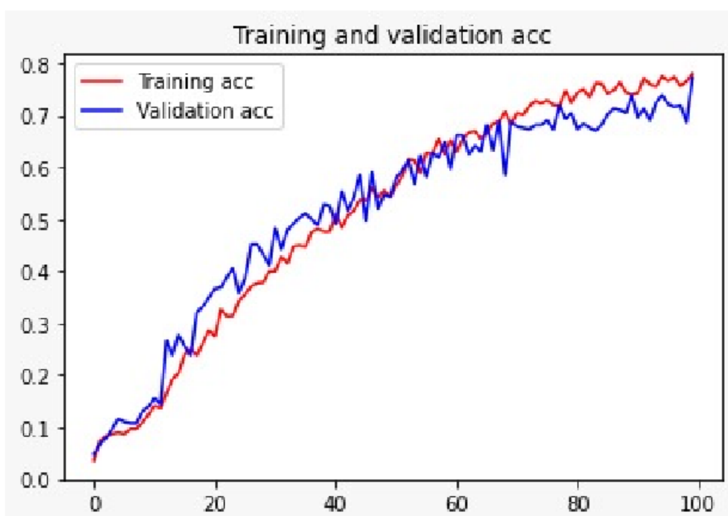


Fig-6.2-Training and Validation Accuracy

By observing above graphs of training & testing accuracy as well as the loss factor of the implemented sequential models. When an image is provided as input to predict, for implemented DL model an accuracy of 75% was achieved.

Verification of the DL model:

When an image is given as input, the proposed CNN model is predicting 7 out of 10 times correctly. Prediction rate of the model can be increased by boosting the training and validation accuracy of the model.

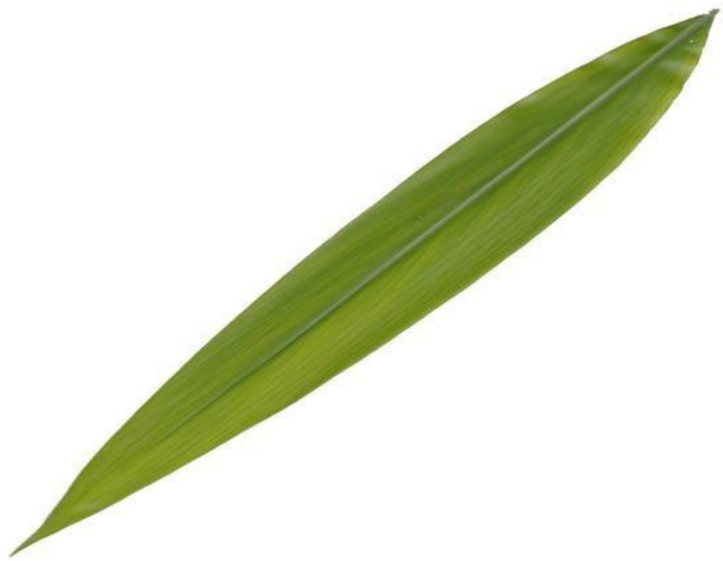


Fig-6.3-Rasna Leaf given as input to DL model to predict

This is an image from the data set of the plant Alpinia Galanga (Rasna). When we give this image for the implemented model to predict it produced an array which shown below,

```
[[1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
  0. 0. 0. 0. 0. 0.]]
```

Fig-6.4-Prediction table of the DL model

Size of the array is 30 which describes the number of plant species in the dataset. In the above output as there is 1 in the 1st position of the array which describes the input image belongs to the 1st species in the dataset which is Alpinia Galanga(Rasna).

6.2 Machine Learning Model Results:

S.no	Class
1	Basella Alba (Basale)
2	Carissa Carandas (Karanda)
3	Ficus Religiosa (Peepal Tree)
4	Jasminum (Jasmine)
5	Mangifera Indica (Mango)
6	Mentha (Mint)
7	Moringa Oleifera (Drumstick)
8	Ocimum Tenuiflorum (Tulsi)
9	Psidium Guajava (Guava)

Table 6.1-Opted classes for model implementation

For the RFC based ML model due to memory constraints, model has been trained with only 9 classes which consist of data from following classes Basella Alba (Basale), Carissa Carandas (Karanda), Ficus Religiosa (Peepal Tree), Jasminum (Jasmine), Mangifera Indica (Mango), Mentha (Mint), Moringa Oleifera (Drumstick), Ocimum Tenuiflorum (Tulsi), Psidium Guajava (Guava). In this paper with accuracy as major metric a comparison has been made between different feature extractors and different classifiers. In the testing phase around 25 images of each leaf class were used to calculate the accuracy of our model. As mentioned above, 9 leaf classes were used to compare different models and feature extractors. Gabor filter bank with Random Forest classifier achieves the highest accuracy of 96%. Next LBGM classifier with GLCM and Shannon entropy as feature extractor were used and achieved an accuracy of 95.5% also Random Forest classifier with GLCM and Shannon entropy extractors achieves an accuracy of same 95.5% max. Table 6.2 gives the details of accuracy after changing various parameters of the classifiers and feature extractors.

Classifier	Features	Accuracy					
RF	**Gabor, pixel and Sobel**	**n_estimator= 40**					
		Max feature	**Accuracy**				
		0.2	0.96				
		0.3	0.96				
		0.5	0.96				
		0.6	0.955				
	GLCM & Shannon entropy	**n_estimator= 40**		**n_estimator= 50**		**n_estimator= 70**	
		Max_feature	**Accuracy**	**Max_feature**	**Accuracy**	**Max_feature**	**Accuracy**
		0.3	0.934	0.3	0.945	0.3	0.956
		0.4	0.923	0.4	0.923	0.4	0.934
		0.8	0.934	0.8	0.934	0.9	0.923
		n_estimator= 90		**n_estimator= 110**		**n_estimator= 140**	
		Max_feature	**Accuracy**	**Max_feature**	**Accuracy**	**Max_feature**	**Accuracy**
		0.3	0.945	0.6	0.945	0.3	0.945
		0.4	0.934	0.8	0.934	0.4	0.934
		0.6	0.923	0.9	0.934	0.6	0.945
LGBM	**GLCM & Shannon entropy**	**lr**	**Accuracy**				
		0.04	0.913				
		0.06	0.934				
		0.08	0.945				
		0.2	0.956				
		0.4	0.956				

Table 6.2-Accuracy comparison of different Classifiers & feature extractors

Gabor filter banks were used to extract the features of leaves which is the most widely used filter and which depends on various parameters like theta, sigma, lambda and gamma etc. It generates filters of various orientations which makes it easier for models to train. 12 Gabor filters were generated with varying parameters of which gamma and lambda values are constant 0.5 and 0.78 respectively. Whereas sigma is incremented from 1 to 5 in steps of 2 in an iterative loop. The final parameter theta which signifies the orientation of the generated filters holds different values in steps of 0.78 for every third iteration cycle.

```
Gabor1 : theta= 0.0 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor2 : theta= 0.0 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor3 : theta= 0.0 : sigma= 5 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor4 : theta= 0.7853981633974483 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor5 : theta= 0.7853981633974483 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor6 : theta= 0.7853981633974483 : sigma= 5 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor7 : theta= 1.5707963267948966 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor8 : theta= 1.5707963267948966 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor9 : theta= 1.5707963267948966 : sigma= 5 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor10 : theta= 2.356194490192345 : sigma= 1 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor11 : theta= 2.356194490192345 : sigma= 3 : lamda= 0.7853981633974483 : gamma= 0.5
Gabor12 : theta= 2.356194490192345 : sigma= 5 : lamda= 0.7853981633974483 : gamma= 0.5
```

Fig-6.5-Gabor filter parameters

The accuracy of each class that has been trained using RFC Classifier and Gabor filter feature extractor is shown below respectively and overall accuracy is around 95.5%.

Class	**Class Name**	**Accuracy**
1	Basella Alba (Basale)	0.9
2	Carissa Carandas (Karanda)	1.0
3	Ficus Religiosa (Peepal Tree)	1.0
4	Jasminum (Jasmine)	0.9
5	Mangifera Indica (Mango)	1.0
6	Mentha (Mint)	0.9
7	Moringa Oleifera (Drumstick)	1.0
8	Ocimum Tenuiflorum (Tulsi)	0.8
9	Psidium Guajava (Guava)	1.0
	Overall Accuracy	**0.955**

Table 6.3-Implemented Classes and Model accuracy

Confusion matrix is plotted below which describes the validation accuracy of each class from Table III.

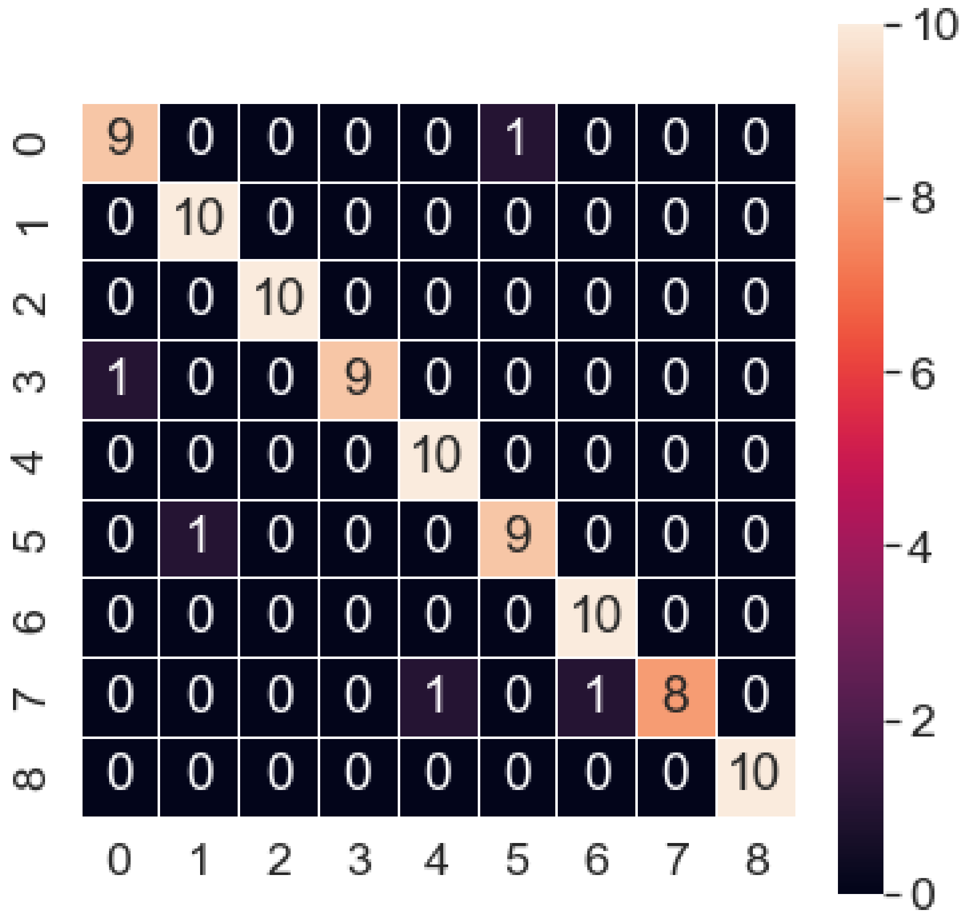


Fig-6.6-Confusion matrix of implemented model

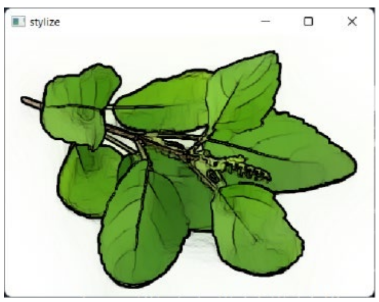
6.3 Segmentation Results:

When an image consisting of multiple leaf images of the same class is given as input, task is to segregate or find contours from the image and give it as an input to implemented models to classify it.

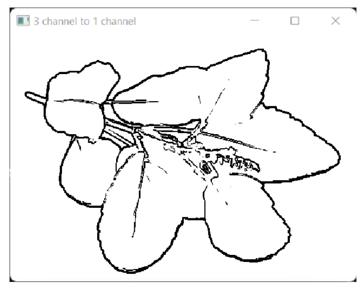
(a)


edge preserve

(b)


stylize

(c)


3 channel to 1 channel

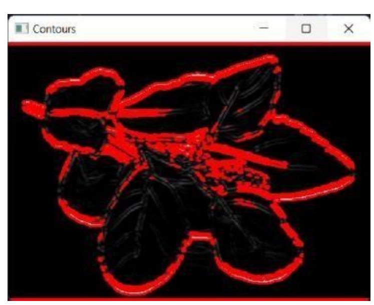
(d)


Sobel Convolution

(e)


blu

(f)


Contours

(g)

Fig-6.7- (a) input image, (b) output of edge preserving filter, (c) output of Stylization filter, (d) conversion from BGR to gray scale, (e) output after applying Sobel filter, (f) output after performing smoothing of edges, (g) depicts the contour detection of the input image.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION:

Implementation of the DL based model and ML models with Classifiers such as RFC, LGBM with different feature extractors is done and made a comparative analysis of model with best accuracy. And implemented a standalone device with raspberry pi and camera module which captures images and classifies to which class the image belongs to, based on the trained data set. And also performed Image segmentation using various techniques and finally found contours of the given image.

7.2 FUTURE SCOPE*:*

1) Implementation of android application and integrate it with hardware for better results and app will be handy and helpful for the researchers/scientists working in the field most of the times in real world applications.
2) Due to memory constrains model is trained with 9 classes as of now, in our future scope training and implementation of the entire data set will be performed.
3) Data set itself can be improved with segmented and non-segmented leaves as it improves the model accuracy, classification and it can be improved by adding images and eliminating unwanted pictures and optimizing each class with same number of images.
4) Still improvement in the ML and DL models can be done for better optimization which requires more research and practice.

REFERENCES:

[1] J. Huixian, “The Analysis of Plants Image Recognition Based on Deep Learning and Artificial Neural Network,” *IEEE Access*, vol. 8, pp. 68828–68841, 2020, doi: 10.1109/ACCESS.2020.2986946.

[2] M. Kumar, S. Gupta, X. Z. Gao, and A. Singh, “Plant Species Recognition Using Morphological Features and Adaptive Boosting Methodology,” *IEEE Access*, vol. 7, pp. 163912–163918, 2019, doi: 10.1109/ACCESS.2019.2952176.

[3] Institute of Electrical and Electronics Engineers. Kerala Section, IEEE Region 10, and Institute of Electrical and Electronics Engineers, *Proceedings of the TENCON 2019 : Technology, Knowledge, and Society : 17-20 October 2019, Grand Hyatt Kochi Bolgatti, Kerala, India.*

[4] J. Joy, S. M., Federal Institute of Science And Technology. Department of Computer Science and Engineering, Institute of Electrical and Electronics Engineers. Kerala Section, and Institute of Electrical and Electronics Engineers, *Proceedings, 2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA) : July 02-04, 2020, Federal Institute of Science And Technology (FISAT), Cochin, Kerala, India*.

[5] L. Gao and X. Lin, “Fully automatic segmentation method for medicinal plant leaf images in complex background,” *Computers and Electronics in Agriculture*, vol. 164, Sep. 2019, doi: 10.1016/j.compag.2019.104924.

[6] M. Dyrmann, H. Karstoft, and H. S. Midtiby, “Plant species classification using deep convolutional neural network,” *Biosystems Engineering*, vol. 151, pp. 72–80, Nov. 2016, doi: 10.1016/j.biosystemseng.2016.08.024.

[7] SCAD College of Engineering and Technology and Institute of Electrical and Electronics Engineers, *Proceedings of the International Conference on Trends in Electronics and Informatics (ICOEI 2019) : 23-25, April 2019*.

[8] R. Khan and R. Debnath, “Segmentation of Single and Overlapping Leaves by Extracting Appropriate Contours,” Nov. 2019, pp. 287–300. doi: 10.5121/csit.2019.91323.

[9] K. Tarun, K. Sreelakshmi, and K. P. Peeyush, “Segregation of Plastic and Non-plastic Waste using Convolutional Neural Network,” in *IOP Conference Series: Materials Science and Engineering*, Nov. 2019, vol. 561, no. 1. doi: 10.1088/1757-899X/561/1/012113.

[10] SCAD College of Engineering and Technology and Institute of Electrical and Electronics Engineers, *Proceedings of the International Conference on Trends in Electronics and Informatics (ICOEI 2019) : 23-25, April 2019*.

[11] IEEE Communications Society and Institute of Electrical and Electronics Engineers, *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI) : 13-16 Sept. 2017.*

[12] Jain College of Engineering, Institute of Electrical and Electronics Engineers. Bangalore Section., and Institute of Electrical and Electronics Engineers, *2020 International Conference for Emerging Technology (INCET) : Belgaum, India. Jun 5-7, 2020.*

CHAPTER 8

PAPER PUBLICATION STATUS

Title of paper: AYURVEDIC MEDICINAL PLANT IDENTIFICATION SYSTEM USING EMBEDDED IMAGE PROCESSING TECHNIQUES
Conference title: SOFT COMPUTING AND SIGNAL PROCESSING (ICSCSP-2022)
Institution: Malla Reddy College of Engineering & Technology, Hyderabad, India
Status: Drafting
Expected date of submission: May 18

Report

ORIGINALITY REPORT

11%	5%	6%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Liverpool John Moores University Student Paper	2%
2	Anu Paulson, S Ravishankar. "AI Based Indigenous Medicinal Plant Identification", 2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA), 2020 Publication	2%
3	M.R. Dileep, P.N. Pournami. "AyurLeaf: A Deep Learning Approach for Classification of Medicinal Plants", TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON), 2019 Publication	1%
4	www.irjet.net Internet Source	1%
5	Submitted to University of Hertfordshire Student Paper	<1%
6	Dyrmann, Mads, Henrik Karstoft, and Henrik Skov Midtiby. "Plant species classification using deep convolutional neural network",	<1%

Biosystems Engineering, 2016.
Publication

7	download.atlantis-press.com Internet Source	<1 %
8	Deepak Sreedharan, M. S. Subodh Raj, Sudhish N. George, S. Ashok. "A Novel Cough Detection Algorithm for COVID-19 Surveillance at Public Places", 2021 8th International Conference on Smart Computing and Communications (ICSCC), 2021 Publication	<1 %
9	Submitted to University of Wales Swansea Student Paper	<1 %
10	ise.ait.ac.th Internet Source	<1 %
11	www.kardyphoto.com Internet Source	<1 %
12	Submitted to Gitam University Student Paper	<1 %
13	Submitted to National College of Ireland Student Paper	<1 %
14	Submitted to Visvesvaraya Technological University Student Paper	<1 %
15	energyinformatics.springeropen.com Internet Source	<1 %

16	Submitted to University of Southampton Student Paper	<1%
17	Bing Han, Yanqi Wu, Lulu Liu. "Prediction and uncertainty quantification of compressive strength of high‐strength concrete using optimized machine learning algorithms", Structural Concrete, 2022 Publication	<1%
18	ad-tech.com.au Internet Source	<1%
19	dokumen.pub Internet Source	<1%
20	www.acarindex.com Internet Source	<1%
21	www.pubfacts.com Internet Source	<1%
22	"Detection and Recognition of Text From Natural Camera Image using Deep Convolutional Network", International Journal of Recent Technology and Engineering, 2019 Publication	<1%
23	inpressco.com Internet Source	<1%
24	www.developer.com Internet Source	<1%

25	"ICT Analysis and Applications", Springer Science and Business Media LLC, 2021 Publication	<1 %
26	tudr.thapar.edu:8080 Internet Source	<1 %
27	Abu Sufian, Anirudha Ghosh, Avijit Naskar, Farhana Sultana, Jaya Sil, M.M. Hafizur Rahman. "BDNet: Bengali Handwritten Numeral Digit Recognition based on Densely connected Convolutional Neural Networks", Journal of King Saud University - Computer and Information Sciences, 2020 Publication	<1 %
28	estudogeral.sib.uc.pt Internet Source	<1 %
29	nuir.lib.nu.ac.th Internet Source	<1 %
30	res.mdpi.com Internet Source	<1 %
31	www.analyticsvidhya.com Internet Source	<1 %
32	www.scribd.com Internet Source	<1 %
33	Reda Ammar, Fathi E. Abd El-Samie, Walid El-Shafai, Noha A El-Hag et al. "Hybrid Method for Contrast Enhancement of Industrial	<1 %

Videoscope Images", 2021 International Conference on Electronic Engineering (ICEEM), 2021
Publication

34	Submitted to University of Nottingham Student Paper	<1 %
35	dspace.akdeniz.edu.tr Internet Source	<1 %
36	iliauni.edu.ge Internet Source	<1 %
37	v3r.esp.org Internet Source	<1 %
38	James R. Strickland. "Raspberry Pi for Arduino Users", Springer Science and Business Media LLC, 2018 Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		