

# **Silk Synergy**

**Shwethashri K and Shreyas S N**

Department of Masters of Computer Applications  
East West Institute of Technology, Bengaluru Karnataka  
shreyassn1715@gmail.com

**Abstract:** *Mulberry is one of the most important agricultural commodities in the world with a number of cuisines scattered across the globe, which are incomplete without it. In developing countries like India, Mulberry has spurred agriculture driven growth in the past century, when export of agricultural produce was the major source of foreign exchange. At times, the prices face a blow from the demand side, while at times facing drastic conditions on the supply side, owing to which, the prices of the commodity have seen a drastic fall. In such years, farmers often cannot afford the services of agricultural consultants for tasks such as detection of leaf diseases and addressing them at the earliest. The solution we have proposed, is a low cost system that uses image processing to detect leaf diseases in the leaves of Mulberry plants to make things easier for both the farmers as well as consumers, since this would balance the prices at a median price, which is both profit-earning for farmers as well as affordable to the consumers at all times of the year.*

*In this project, the affected leaves are captured as images using a camera. These images are then processed further using various methods and the dominant features are extracted from them using various methods. Comparison of the features is done using various algorithms that detect the variance in color and its dominance in the recorded samples. This will help in faster and cost effective addressing of such diseases.*

**Keywords:** *Mulberry*

## **I. INTRODUCTION**

One factor that significantly influences the quantity and quality of mulberry output is the state of the leaves. An image processing-based method called mulberry leaf disease detection employs visual clues on the leaf surface to identify the disease. Numerous methods have been developed to investigate the Mulberry leaves using cameras and computer vision techniques.

Due to the high cost of hiring agricultural consultants and advisors, the complexity and unfriendliness of the development environments, and other factors, sophisticated computer systems to detect such diseases were never developed or made widely available in the past. But given that the platforms are now more user-friendly than before, isn't it time to review this choice? Remedial actions for leaf diseases can be easily adopted when a computer is trained to spot them in the first place. Using image processing is a good strategy since diseases may be perceived by vision, as they influence a leaf's appearance.

In contrast to other applications where a miscommunication could impede project progress, the appearance of a specific undesired characteristic or color on a "mulberry leaf" affects all countries that produce mulberries.

## **II. EXISTING SYSTEM**

The health and productivity of mulberry trees, which are essential for the manufacture of silk, depend on the early detection of illnesses in mulberry leaves. The project's current approach combines machine learning methods with image processing techniques. Initially, digital cameras or cellphones are used to take pictures of mulberry leaves. Then, using image processing techniques like thresholding, edge detection, and segmentation, these photos are processed to extract pertinent aspects like color, texture, and shape

Following feature extraction, the data are fed into machine learning models, which include CNNs, Decision Trees, and Support Vector Machines (SVMs). In order to categorize fresh photos into healthy or unhealthy categories, these models were trained using a dataset that included pictures of both healthy and sick mulberry leaves.

Through feedback mechanisms, where misclassifications are reviewed to increase accuracy and refine algorithms, the system's performance is continuously assessed and improved. The system's ultimate objective is to give farmers and researchers an accurate and effective tool for early disease detection, allowing for prompt management practices and intervention to guarantee the healthy growth of mulberry trees and sustained silk production.

### **III. PROPOSED SYSTEM**

The science of artificial intelligence and deep learning was restored to its previous luster with Yann LeCun's invention of CNN in 1994. Since the initial neural network, known as LeNet5, had a relatively low validation accuracy of 42%, this field has advanced significantly. CNN is now used by practically all large technological companies for more effective performance. CNN is used in the concept of mulberry leaf disease detection. We need to have a fundamental understanding of how the human brain distinguishes between objects despite their differences in qualities before delving into the "functionality and working of CNN" concept. Our brain is made up of many layers of complex neurons, each of which contains information about an object. The neurons then extract and store all of the object's features in our memory so that the next time we see the same object, our brain will match the stored features to recognize it. This process can be mistaken for a straightforward "IF- THEN" function, which it is to some extent, but it also has a feature called self-learning, which gives it an advantage over other algorithms even though it can't match the functionality of the human brain. For the purpose of identifying leaf diseases, an image is processed using a basic CNN. The following needs to be met by our CNN model's data training

There shouldn't be any missing values in our dataset.

When it comes to an image dataset, all of the images must be the same size because an uneven distribution of image sizes in our dataset can reduce the efficiency of our neural network. The dataset must be clearly divided into training and testing sets, and neither the training nor the testing set should contain any irrelevant data outside of our model domain.

Before feeding the photos into the convolution layer, they should be transformed to black and white format. This is because reading images in RGB would need a 3-D NumPy matrix, which will significantly shorten our model's execution time.

Prior to feeding the database into the neural network, any damaged or fuzzy photos should also be removed. Now that we are aware of the data pre-processing guidelines, let's immediately examine how a neural network using convolution functions.

### **IV. IMPLEMENTATION**

Implementation is the platform where the project working is demonstrated. In implementation the system drives for real operations. Every consequence completed by the project will be abundant, if processes are accurately executed according to plan carried out.

#### **Implementation requirement**

To implement Mulberry Leaf Disease Detection using

Convolutional Neural Network, software's used are

Language used to code the project is Python.

Operating System-Windows 10.

Visual studio code:

Visual Studio Code is a freeware source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python and C++. It is based on the Electron framework,[19] which is used to develop Node.js Web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component

(codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a languageagnostic code editor for any language. It supports a number of programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Many Visual Studio Code features are not exposed through menus or the user interface but can be accessed via the command palette.

Programming language used:

Programming language used to design the proposed method is Python. Python is a high-level programming language with dynamic semantics. It is an interpreted language i.e. interpreter executes the code line by line at a time, thus makes debugging easy. Python Imaging Library (PIL) is one of the popular libraries used for image processing. PIL can be used to display image, create thumbnails, resize, rotation, convert between file formats, contrast enhancement, filter and apply other digital image processing techniques etc.

Python is often used as a support language for software developers, for build control and management, testing, and in many other ways. Python is designed by Guido van Rossum. It is very easy for user to learn this language because of its simpler coding. It provides an easy environment to furnish computation, programming visualization. Python supports modules and packages, which encourages program modularity and code reuse. It has various built-in commands and functions which will allow the user to perform functional programming. Apart from being an open-source programming language, developers use it extensively for application development and system development programming. It is a highly extensible language. Python contains many inbuilt functions which helps beginners to learn easily.

Some of the most commonly used functions are: `imread` will read image from specified location, `imshow` will display the output or images on the screen, `cvtColor` function converts binary image into grayscale image. `Compare_ssim` function will compare pre-flood and post-flood images, `findContours` will give the region of difference between two input images.

#### **Key features of python:**

Python is an interpreted language i.e. interpreter executes the code line by line at a time, thus makes debugging easy.

Python is more expressive language, since it is more understandable and readable.

To design and solve problems it offers an interactive atmosphere.

#### **TKinter GUI:**

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task. User need not want to create script or write commands in command prompt. Instead user must be aware of how the programs are performs to accomplish tasks. It includes Radio buttons, Toolbars, Sliders, Axes etc.

#### **Open cv-python:**

OpenCV-Python is a library of Python bindings designed to solve computer vision problems. Visual information is the most important type of information perceived, processed and interpreted by the human brain. Image processing is a method to perform some operations on an image, in order to extract some useful information from it. An image is nothing more than a two dimensional matrix (3-D in case of colored images) which is defined by the mathematical function  $f(x,y)$  where  $x$  and  $y$  are the two coordinates horizontally and vertically. The value of  $f(x,y)$  at any point is gives the pixel value at that point of an image, the pixel value describes how bright that pixel is, and/or what color it should be. In image processing we can also perform image acquisition, storage, image enhancement.

**Flask:**

Flask (source code) is a Python web framework built with a small core and easy-to-extend philosophy. Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks.

Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy. Flask is considered more Pythonic than the Django web framework because in common situations the equivalent Flask web application is more explicit. Flask is also easy to get started with as a beginner because there is little boilerplate code for getting a simple app up and running.

Flask was also written several years after Django and therefore learned from the Python community's reactions as the framework evolved. Jökull Sólberg wrote a great piece articulating to this effect in his experience switching between Flask and Django.

**Google Colab:**

Colaboratory, or "Colab" for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary Python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs. Colab resources are not guaranteed and not unlimited, and the usage limits sometimes fluctuate. This is necessary for Colab to be able to provide resources for free.

Jupyter is the open source project on which Colab is based. Colab allows you to use and share Jupyter notebooks with others without having to download, install, or run anything. Code is executed in a virtual machine private to your account. Virtual machines are deleted when idle for a while, and have a maximum lifetime enforced by the Colab service. Resources in Colab are prioritized for users who have recently used less resources, in order to prevent the monopolization of limited resources by a small number of users.

**Packages:**

Packages are the namespaces which consist of multiple packages and modules themselves. Each package in Python is a directory which must contain a special file called `_init_.py`. This file can be empty, and it indicates that the directory it contains is a Python package, so it can be imported in the same way that the module can be imported.

As our application program grows larger in size with a lot of modules, one can place the similar modules in one package and different modules in different packages. This makes a program easy to manage and conceptually clear. We can import the modules from packages using the `dot` (`.`) operator.

**CSV**

In this work, to import or export spreadsheets and databases for its use in the Python interpreter, the CSV module, or Comma Separated Values format is used.

These CSV files are used to store a large number of variables or data and the CSV module is a built-in function that allows Python to parse these types of files. The text inside a CSV file is organized in the form of rows, and each row consists of the columns, all separated by commas, indicates the separate cells in CSV file. There is no standard for CSV modules hence, these modules make use of "dialects" to support parsing using different parameters.

The CSV module includes all the necessary built-in functions, `csv.reader` and `csv.writer` are the most commonly used built-in functions in Python. These functions are given below:

```
csv.reader(csvfile, dialect='excel', **fmtparams)
```

`csvfile` can be any object which supports the iterator protocol and returns a string each time its `next()` method is called – file objects and list objects are both suitable. If `csvfile` is a file object, it must be opened with the 'b' flag on platforms where that makes a difference. An optional `dialect` parameter can be given which is used to define a set of parameters specific to a particular CSV dialect. It may be an instance of a subclass of the `Dialect` class or one of the strings returned by the `list_dialects()` function. The other optional `fmtparams` keyword arguments can be given to override

individual formatting parameters in the current dialect. Each row read from the csv file is returned as a list of strings. No automatic data type conversion is performed.

```
csv.writer(csvfile, dialect='excel', **fmtparams)
```

It returns a writer object responsible for converting the user's data into delimited strings on the given file-like object. csvfile can be any object with a write() method. If csvfile is a file object, it must be opened with the 'b' flag on platforms where that makes a difference. An optional dialect parameter can be given which is used to define a set of dialect. It may be an instance of a subclass of the Dialect class or one of the strings returned by the list\_dialects() function. The other optional fmtparams keyword arguments can be given to override individual formatting parameters in the current dialect. Floats are stringified with repr() before being written. All other non-string data are stringified with str() before being written.

### **Tensorflow:**

it is a free and open source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural network, it is used for both research and production at Google.

Tensorflow was developed by the Google Brain team for internal Google use. It was released under the Apache License 2.0 on November 9,

2015. Tensorflow is Google Brain's second-generation system. Version 1.0.0 was released on February 11, 2017. While the reference implementation runs in single devices, Tensorflow can run on multiple CPUs and GPUs. Tensorflow is available on 64bit Linux, macOS, Windows and mobile computing platforms including Android and ios. Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs and TPUs), and from desktops to clusters to servers to mobile and edge devices. Tensorflow computations are expressed as state full dataflow graphs. The name tensorflow derive from the operations that such neural networks perform on multidimensional data arrays, which are referred to as tensors.

### **Keras:**

Keras is an open source neural- network library written in Python. It is capable of running on top of Tensorflow, Microsoft Cognitive Toolkit, R, Theano or PlaidML. It is designed to enable fast experimentation with deep neural networks. It focuses in being user-friendly, modular, and extensible CUDA. Keras contains numerous implementations of commonly used neural- network building blocks such as layers, objectives, activation function, optimizers, and a host of tools to make working with image and text data easier to simplify the code necessary for writing deep neural network code. The code is hosted on GitHub, and commonly support forums include the GitHub issues page, and a Slack channel. In addition to Standard neural networks, Keras has support for convolutional and recurrent neural networks. It supports other common utility layers like dropout, batch normalization and pooling. Keras allows users to productize deep models on smart phones, on the web or on the java virtual machine. It also allows use of distributed training of deep learning models on clusters of Graphics processing units (GPU) and tensor processing units (TPU) principally in conjunction with CUDA.

### **NumPy:**

NumPy is a python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python. In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently used in data science, where speed and resources are very important. NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently. This behaviour is called locality of reference in computer science. This is the main reason why NumPy is faster than lists. Also, it is optimized to work with latest CPU architectures.



**Matplotlib:**

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, web application servers, and various graphical user interface toolkits. Matplotlib.pyplot is a collection of command style functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc. Some people embed matplotlib into graphical user interfaces like wxpython or pygtk to build rich applications. Others use matplotlib in batch scripts to generate postscript images from some numerical simulations, and still others in web application servers to dynamically serve up graphs.

Pseudocodes For Pre-Processing Techniques 4.3.1: Pseudocode for picture uploading:

Input image is obtained from the dataset which represents a healthy or diseased leaf.

**Step 1:** Click on upload

**Step 2:** Read the leaf image from dataset `imageA=cv2.imread(args[“first”])`

**Step 3:** Display the name of the image on the webpage

**Step 4:** Shows the name of the image in the dialog box

Pseudocodes for converting rgb to gray:

In this section the RGB image is converted to gray image because it is easy to perform the operations such as median filtering, thresholding and highpass filtering.

**Step 1:** Converting RGB to Gray

**Step 2:** Show the window to display the image

**Step 3:** Display the image in the window

**Step 4:** Multiply each plane with a threshold value **Step 5:** `Gray_pixel=(R_plan*R_thresh)+ (G_plan* G_thresh) + (B_plan * B_thresh);` Pseudocode for Thresholding:

Image thresholding is a simple, yet effective, way of partitioning an image into a foreground and background. This image analysis technique is a type of image segmentation that isolates objects by converting grayscale images into binary images.

Step 1: for  $i:=1$  to  $480*640$  pixels

Step 2: if `new_im(i)<T`

Step 3: change to black, and consider as obstacle set `new_im(i)=0`

Step 4: Change to white, and consider as free space else set `new_im(i)=1`

Pseudocode for High Pass Filtering:

A high-pass filter can be used to make an image appear sharper. These filters emphasize fine details in the image- which is exactly the opposite of low pass filters. High pass filtering works in exactly the same way as a low pass filter except that it uses a different convolution kernel. Unfortunately, while low-pass filtering smooths out noise, high-pass only does the opposite by amplifying noise. However, if the original image is not too noisy- this is avoided to an extent.

Step 1: Displays the high contrast version of original image.

Step 2: Make a copy of the original image. Initialize `color[ ][ ]` `retval copy(original)` Step 3: Blur the copy of original image.

Step 4: Subtract the blur image from original image. `Color[ ][ ]` `highpass difference(original,blurred)`

Step 5: Add the unsharp mask to original image to get sharpened image

Step 6: Run a loop for the entire rows and columns for `row=0` to `original.length` repeat until for `col=0` to `original[row].length` repeat until initialize `color origColor original[row][col]`

initialize `contrastColor highContrast[row][col]` set `color difference contrastColor - origColor` end for end for

#### V. CONCLUSION

With accuracy rates of over 80%, VGG is by far the best model for this use, although requiring greater computing power. The Mobile Net concept may be more appropriate for older devices with slower processors. Despite having lower accuracy rates, it provides a productive experience.

In this study, different CNN architectures are used to detect Mulberry leaf illnesses using a very straightforward yet effective method. Our project focuses on several techniques for classifying and predicting leaf diseases. Additionally, we go through various image processing approaches in the proposed methodology. We can change the existing algorithms to classify leaves with a high degree of accuracy. Farmers will be able to take early precautions and avoid suffering significant losses with the accuracy and timely diagnosis of these diseases. The technique employed will yield the disease's name together with the class to which the photograph belongs. By giving farmers some basic knowledge about the illness, this effort also assists farmers in combating the condition. The outcomes of the mechanism that was put into place were quite precise. The ability for government officials to use this research across a wide range of farms to determine whether some illnesses have become more prevalent than others and are proving challenging to control is a significant improvement.

#### REFRENCES

- [1] The Mulberry Leaf Disease Detection using Convolutional Neural Networks is presented in the Proceedings of the Eleventh International Conference on Contemporary Computing, which was held in Noida, India from August 2-4, 2018. Prajwala T M, et al.
- [2] Image-Based Mulberry Leaf Disease Detection, 10th ICCNT, July 68, 2019, IIT- Kanpur, Kanpur, India, Akshay Kumar, et al.
- [3] Mokhtar, et al., "Detection of Mulberry leaf diseases using Support V vector machines," IEEE International Computer Engineering Conference- ICENCO, Cairo, December 30, 2015.
- [4] Balakrishna K & Rao M, Disease Classification of Mulberry Plant Leaves Using KNN and PNN. 2019; 9(1):51–56; International Journal of Computer Vision and Image Processing.
- [5] Plant leaf disease detection and classification using image processing approaches, Mainkar, P. M., Ghorpade, S., & Adawadkar, M. International Journal of Innovative and Emerging Research in Engineering, 2015.
- [6] Xie C, Shao Y, Li X, & He Y, Evaluation of hyperspectral imaging for the detection of early and late blight infections on mulberry leaves, Scientific Reports, 5, 16564, 2015.
- [7] Jihen Amara, Alsayed Algergawy, Bassem Bouaziz, et al. Deep Learning-based Approach for Classification of Banana Leaf Diseases, BTW, 2017.
- [8] H. Sabrol and K. Satish, Disease classification of mulberry plants in digital photos using classification tree, International Conference on Communication and Signal Processing (ICCSP), 2016 IEEE, 2016