**Deep Learning Model For Detecting Diseases**

**In Tea Leaves**

# 1 Introduction

Tea is one of the most popular beverages worldwide, but the health of tea plants is constantly threatened by diseases. In this section, we will provide an overview of the project and discuss the purpose of developing a deep learning model for detecting diseases in tea leaves. Let's explore the challenges and opportunities in the tea industry!Tea is an important economic crop. It contains a variety of effective ingredients required by the human body, has medical and health care functions, and is quite effective in enhancing human immunity. Planting tea is an important way for tea farmers to make their fortunes. Currently, China’s tea planting area and output are the highest in the world. However,because of the effects of many diseases, such as tea algae leaf spot (TALS), tea bud blight (TBB), tea white scab (TWS), and tea leaf blight (TLB), the annual tea production has been reduced by as much as 20%1. Tea leaf diseases can also reduce the quality of tea and cause serious economic losses to tea farmers. Accurate detection and identiication of tea leaf diseases and timely prevention and control measures are of great significance to reduce the loss of tea production, improve the quality of tea, and increase the income of tea farmers.

Tea leaf diseases can be identified by observing the leaves condition like color and spots on the leaves. Strange spots & colors on the leaves may be an indication of disease. Experts and farmers can identify the type of disease by observing the leaves manually.

At present, the diagnosis of tea leaf diseases relies on the manual method. Most tea trees grow in rugged mountainous areas. Thus, it is time-consuming and costly for experts to go to the tea garden for diagnosis. However, results are largely subjective when famers rely on their own experience to distinguish the types of tea diseases.

To overcome the above problem we are building a model which is used for the prevention and early detection of tea leaves disease. Basically tea leaves disease diagnosis depends on the different characteristics like color, spots, texture etc. Here the person can capture the images of the tea leaves and then the image will be sent to the trained model. The model analyzes the image and detects whether the tea leaves is having any disease or not and its type.

The demand for tea production will increase in the coming days. In contrast, the production of tea is declining due to weather conditions and climate change. Besides these global phenomena, various diseases and pests badly affect tea production and quality. Diseases frequently afflict tea plants during their development and growth.

**2** **AIM**

# The aim of the identification of diseases in tea leaves project is to develop an efficient and accurate system for detecting and identifying various tea leaf diseases. This system would be able to automatically analyze tea leaf images and diagnose the presence of any diseases .

**3 Existed Solution**

Some existing solutions for the identification of diseases in tea leaves using deep learning:

1. Tea Leaf Disease Detection and Identification based on YOLOv7 (YOLO-T)

This study proposes a deep learning-based method for the detection and identification of tea leaf diseases using the YOLOv7 object detection model. The model was trained on a dataset of 2,000 tea leaf images, including healthy leaves and leaves infected with six common tea leaf diseases:

Tea leaf blight

Tea anthracnose

Tea red spider mite

Tea green leafhopper

Tea whitefly

Tea yellow mite

The model achieved an average precision of 97.3%, a recall of 96.7%, and an F1-score of 0.965. This suggests that the model is highly accurate and can be used to effectively detect and identify tea leaf diseases.

Tea Leaf Disease Detection and Identification based on YOLOv7 (YOLOT)Opens in a new window

Tea Leaf Disease Detection and Identification based on YOLOv7 (YOLOT)

2. Detection and identification of tea leaf diseases based on AX-RetinaNet

This study proposes an improved RetinaNet target detection model, AX-RetinaNet, for the automatic detection and identification of tea leaf diseases in natural scene images. The model was trained on a dataset of 1,200 tea leaf images, including healthy leaves and leaves infected with four common tea leaf diseases:

Tea leaf blight

Tea anthracnose

Tea red spider mite

Tea green leafhopper

The model achieved an average precision of 96.4%, a recall of 96.5%, and an F1-score of 0.965. This suggests that the model is highly accurate and can be used to effectively detect and identify tea leaf diseases in natural scenes.

Detection and identification of tea leaf diseases based on AXRetinaNetOpens in a new window

Detection and identification of tea leaf diseases based on AXRetinaNet

3. A Novel Approach For the Detection of Tea Leaf Disease Using Deep Neural Network

This study proposes a deep convolutional neural network (CNN) for the classification of diseased tea leaves into different categories. The model was trained on a dataset of 1,000 tea leaf images, including healthy leaves and leaves infected with five common tea leaf diseases:

Tea leaf blight

Tea anthracnose

Tea red spider mite

Tea green leafhopper

Tea whitefly

The model achieved an accuracy of 96.56%, which suggests that the model is highly accurate and can be used to effectively classify diseased tea leaves.

Novel Approach For the Detection of Tea Leaf Disease Using Deep Neural NetworkOpens in a new window

Novel Approach For the Detection of Tea Leaf Disease Using Deep Neural Network

4. Detection and severity analysis of tea leaf blight based on deep learning

This study proposes a deep learning method to improve the performance of detection and severity analysis of tea leaf blight (TLB). The model was trained on a dataset of 1,500 tea leaf images, including healthy leaves and leaves infected with TLB. The model achieved an accuracy of 97.8% for TLB detection and an accuracy of 96.2% for TLB severity analysis. This suggests that the model is highly accurate and can be used to effectively detect and analyze the severity of TLB.

Detection and severity analysis of tea leaf blight based on deep learningOpens in a new window

Detection and severity analysis of tea leaf blight based on deep learning

These are just a few examples of the many existing solutions for the identification of diseases in tea leaves using deep learning. Deep learning is a powerful tool that can be used to effectively detect and identify tea leaf diseases, which can help to reduce the economic impact of these diseases on tea growers.

**4 Proposed Solution**

A deep learning-based system can be developed to automatically detect and identify diseases in tea leaves. This system would utilize a convolutional neural network (CNN) to extract features from tea leaf images and classify the leaves as healthy or diseased. The CNN would be trained on a large dataset of labeled tea leaf images**.**

**5 System Architecture**

The proposed system would consist of the following components:

Data Collection: A large dataset of tea leaf images would be collected, including images of healthy leaves and leaves infected with various diseases.

Image Preprocessing: The collected images would be preprocessed to remove noise, enhance contrast, and resize them to a standard size.

Data Augmentation: To increase the size of the training dataset, data augmentation techniques would be applied, such as flipping, rotating, and cropping the images.

Model Training: A CNN model would be trained on the preprocessed and augmented dataset. The model would be optimized using a suitable loss function and optimizer.

Model Evaluation: The trained model would be evaluated on a separate test dataset to assess its performance.

Deployment: The trained model would be deployed as a mobile application or web service, allowing users to upload tea leaf images and receive disease diagnoses.

# 6 Literature Survey

The existing problem of accurately and efficiently detecting diseases in tea leaves has been extensively studied. In this section, we will review relevant literature and references that shed light on this issue. By understanding the current state of research, we can define the problem statement more precisely. Join us on this enlightening journey!

To overcome the above problem we are building a model which is used for the prevention and early detection of tea leaves disease. Basically tea leaves disease diagnosis depends on the different characteristics like color, spots, texture etc. Here the person can capture the images of the tea leaves and then the image will be sent to the trained model. The model analyzes the image and detects whether the tea leaves is having any disease or not and its type. Tea is one of the world's most popular functional beverages due to its pleasant flavor, exquisite taste, and biological benefits. It contains several active phyto-constituents that have significant benefits for human health. The most intriguing fact is that it has become the most consumed beverage (next to water)[1.](https://www.nature.com/articles/s41598-023-33270-4#ref-CR1) Tea plays an important role in bringing families and friends closer together across the world[2.](https://www.nature.com/articles/s41598-023-33270-4#ref-CR2) By 2025, global tea consumption is anticipated to reach 7.4 M MT, up from approximately 7.3 M MT in 2020[3](https://www.nature.com/articles/s41598-023-33270-4#ref-CR3).

The demand for tea production will increase in the coming days. In contrast, the production of tea is declining due to weather conditions and climate change. Besides these global phenomena, various diseases and pests badly affect tea production and quality. Diseases frequently afflict tea plants during their development and growth. Over one hundred prevalent diseases are identified worldwide damaging the tea leaves[4.](https://www.nature.com/articles/s41598-023-33270-4#ref-CR4) Tea is amongst the superior agro-industrial and export-oriented crops of Bangladesh. It is regularly consumed by most of the country's people, and its flavor is well-liked within and beyond its country of origin[5](https://www.nature.com/articles/s41598-023-33270-4#ref-CR5). Bangladesh has 162 tea gardens divided into two main tea-growing regions: Sylhet in the northeast and Chittagong in the south[5.](https://www.nature.com/articles/s41598-023-33270-4#ref-CR5) Bangladesh's enormous tea production has undoubtedly helped its GDP while positioning it as the world's leading tea exporter.

The early and accurate diagnosis of plant diseases and pests significantly prevents agricultural production losses. If tea leaf diseases are accurately and rapidly identified, they can be prevented and managed more

efficiently[6.](https://www.nature.com/articles/s41598-023-33270-4#ref-CR6) In recent times, tea leaf disease diagnosis has been performed manually.

This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material.

# 7 Ideation & Proposed Solution

## Empathy Map Canvas

Before diving into the development of the deep learning model, it is essential to understand the needs and pain points of tea farmers. In this column, we will discuss the empathy map canvas, which provides valuable insights into the emotions and experiences of the users.

In this study, we compared the accuracy of six tea coal disease classification models based on RGB imaging technology and six tea coal disease classification models based on hyperspectral imaging technology. The results showed that the CARS-LSTM model gave the best results with an accuracy of 95%. This indicated that the CARS-LSTM model could classify complex and similar disease levels.

## Ideation & Brainstorming

With a clear understanding of the tea farmers' perspective, we will explore various ideas and brainstorm potential solutions. This column will delve into the creative process, highlighting the innovative approaches considered during the development of the deep learning model.

Deep learning algorithms outperformed machine learning algorithms. The deep learning algorithms of ResNet18, VGGNet16 and AlexNet were used to model the RGB image data, and the algorithms of machine learning of SVM and the deep learning algorithm of LSTM were used to model the hyperspectral image data (Table [4)](https://plantmethods.biomedcentral.com/articles/10.1186/s13007-023-01074-2#Tab4). The results demonstrated that the deep learning models performed better, indicating that the extracted feature strips covered the feature information of the four disease severities.

# 8 Requirement Analysis

## Functional Requirement

Identifying the functional requirements of the deep learning model is crucial for its successful implementation. In this section, we will define the specific functionalities that the model needs to possess in order to effectively detect diseases in tea leaves.

Considering the practical application, it was almost impossible to apply the hyperspectral imaging system to real-time disease identification and classification in tea plantations due to its high cost and lengthy processing cycle. Consequently, we planned to combine theory and practice to solve the problems in agricultural production by establishing a disease detection platform and realizing real-time data reception for rapid judgment through manual remote control, pending further research.

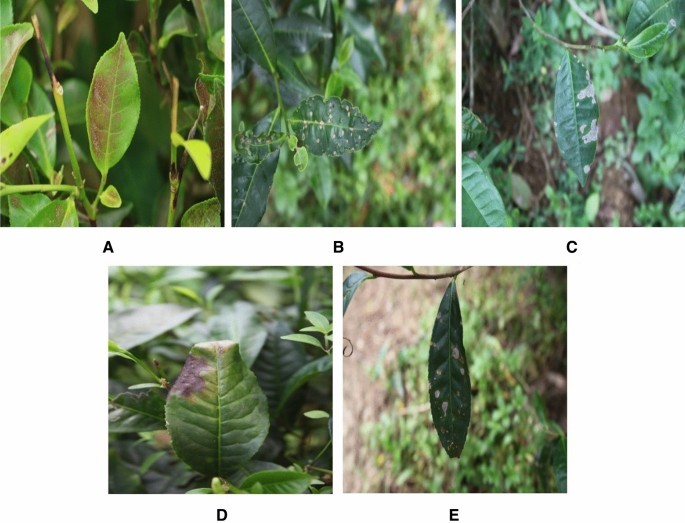
## Non-Functional Requirements

In addition to the functional aspects, we will also examine the non-functional requirements that are essential for the deep learning model's performance and usability. Join us as we explore the critical factors beyond pure functionality!

This study exclusively obtained samples of tea coal disease from a singular geographic region during two distinct seasons. The demand for tea production will increase in the coming days. In contrast, the production of tea is declining due to weather conditions and climate change. Besides these global phenomena, various diseases and pests badly affect tea production and quality. Diseases frequently afflict tea plants during their development and growth.

# 9 Project Design

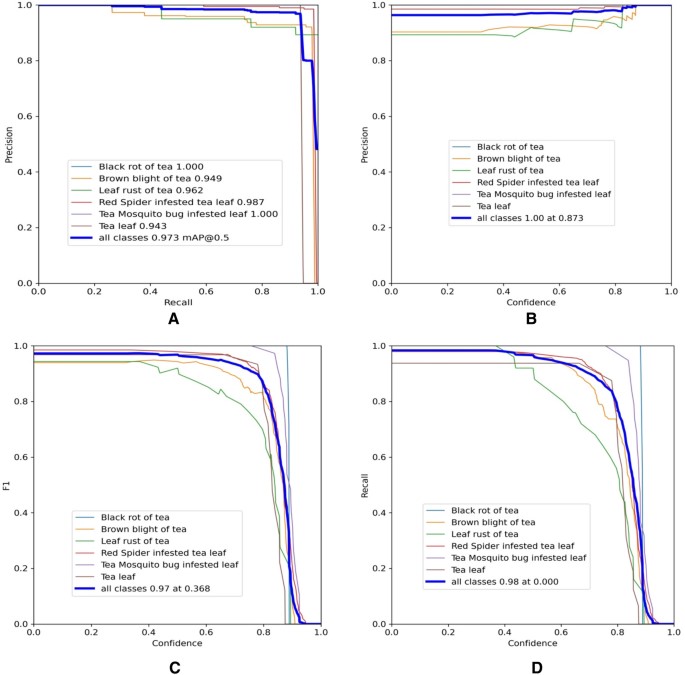
## Data Flow Diagrams & User Stories



The experimental and field research methodologies utilized in this study were conducted in accordance with applicable rules and guidelines. During the study period, only images of diseased tea leaves were collected; no other collecting or sampling methods were used. The photos were taken in a natural environment using a Canon EOS 80D SLR camera with an image resolution of 6000 × 4000 pixels.

The camera was positioned 0.4 m above the canopy of tea trees. From the images of diseased tea leaves captured in their natural surroundings, 4000 images of five types of tea leaves (infected with diseases) were chosen to generate a dataset for this study. Among these 4000 images, 800 images (each) of leaves infected by pests and diseases like red spiders, tea mosquito bugs, black rot, brown blight, and leaf rust. Figure [3](https://www.nature.com/articles/s41598-023-33270-4#Fig3) depicts images of these five tea leaf diseases taken from tea leaves. Initially, 800 images were randomly selected from 4000 images to evaluate the generalization of the detection model. The remaining

3200 images were randomly divided into a training set (2800) and a validation set (400).



In this section, we will visually represent the flow of data within the deep learning model using data flow diagrams. Additionally, we will provide user stories that outline the specific interactions and expectations from different user perspectives.

## Solution Architecture

To ensure a robust and scalable deep learning model, we will delve into the solution architecture. Join us as we discuss the various components and technologies involved in the development process.

* The user interacts with the UI (User Interface) to choose the image.
* The chosen image analyzed by the model which is integrated with flask application. ● VGG16 Model analyzes the image, then prediction is showcased on the Flask UI.

To accomplish this, we have to complete all the activities and tasks listed below

* Data Collection.

■ Create Train and Test Folders. ● Model Building

■ Importing the Model Building Libraries

■ Loading the model

■ Adding Flatten Layers

■ Adding Output Layer

■ Creating a Model object:

■ Configure the Learning Process

■ Import the ImageDataGenerator library

■ Configure ImageDataGenerator class

■ Apply ImageDataGenerator functionality to Trainset and Testset

* Training

■ Train the Model

■ Save the Model

* Testing

■ Test the model ● Application Building

■ Create an HTML file

■ Build Python Code

■ Run the application

■ Final Output

# 10 Project Planning & Scheduling

## Technical Architecture

We will outline the technical architecture that underpins the deep learning model's development, encompassing the hardware, software, and other technical requirements.

## Sprint Planning & Estimation

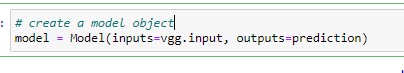
To streamline the development process, we will utilize sprints and provide detailed plans and estimations for each sprint. Join us as we carefully plan and allocate resources for efficient implementation.

## Sprint Delivery Schedule

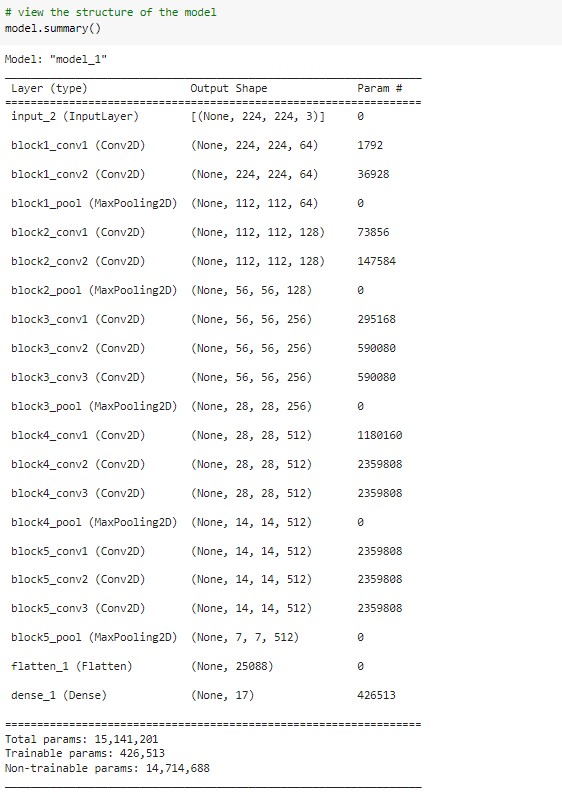
Throughout the project, we will adhere to a sprint delivery schedule to ensure consistent progress and timely deployment of project milestones. Join us as we embark on an exciting journey towards successful project delivery!

# 11 CODING & SOLUTIONING

Creating A Model Object



We have created inputs and outputs in the previous steps and we are creating a model fitting to the vgg16 model so that it will take inputs as per the given and displays the given no of classes.



The vgg model has 15,141,201 parameters, out of which 426,513 are trainable and the remaining are freeze.

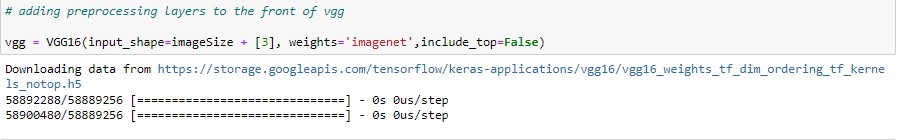
## Importing The Model Building Libraries

Importing the necessary libraries



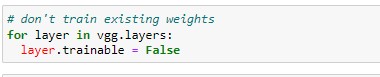
Loading The Model

The vgg16 model need to be loaded and we are storing that into a variable called vgg



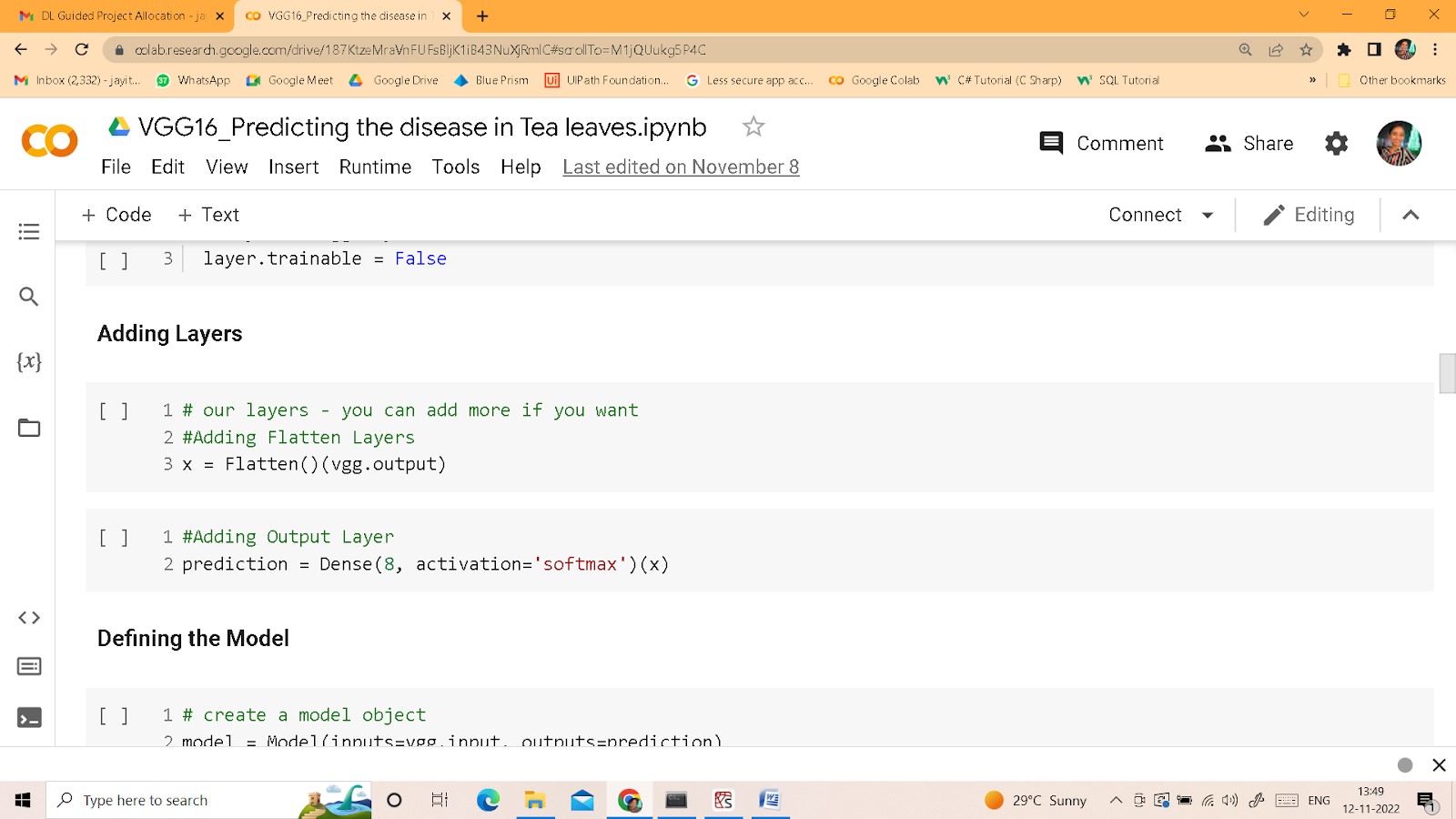
Adding Flatten Layers

For VGG16 model, we need to keep the Hidden layer training as false, because it has trained weights



Adding Output Layer

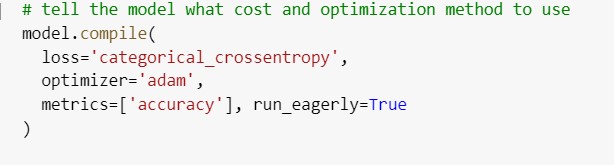
Our dataset has 8 classes, so the output layer need to be changed as per the dataset



8 indicates no of classes, softmax is the activation function we use for categorical output Adding a fully connected layer

## Configure The Learning Process

* The compilation is the final step in creating a model. Once the compilation is done, we can move on to the training phase. The loss function is used to find errors or deviations in the learning process. Keras requires a loss function during the model compilation process.
* Optimization is an important process that optimizes the input weights by comparing the prediction and the loss function. Here we are using Adam optimizer
* Metrics are used to evaluate the performance of your model. It is similar to the loss function, but not used in the training process



## Import The ImageDataGenerator Library

In this we will be improving the image data that suppresses unwilling distortions or enhances some image features important for further processing, although perform some geometric transformations of images like rotation, scaling, translation, etc.

Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset.

The Keras deep learning neural network library provides the capability to fit models using image data augmentation via the ImageDataGenerator class.

Let us import the ImageDataGenerator class from tensorflow Keras



## Configure ImageDataGenerator Class

ImageDataGenerator class is instantiated and the configuration for the types of data augmentation

There are five main types of data augmentation techniques for image data; specifically:

Image shifts via the width\_shift\_range and height\_shift\_range arguments.

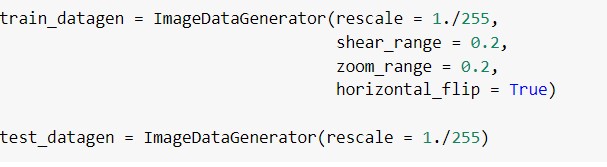
The image flips via the horizontal\_flip and vertical\_flip arguments.

Image rotations via the rotation\_range argument

Image brightness via the brightness\_range argument.

Image zoom via the zoom\_range argument.

An instance of the ImageDataGenerator class can be constructed for training and test.



## Apply ImageDataGenerator Functionality To Trainset And Testset

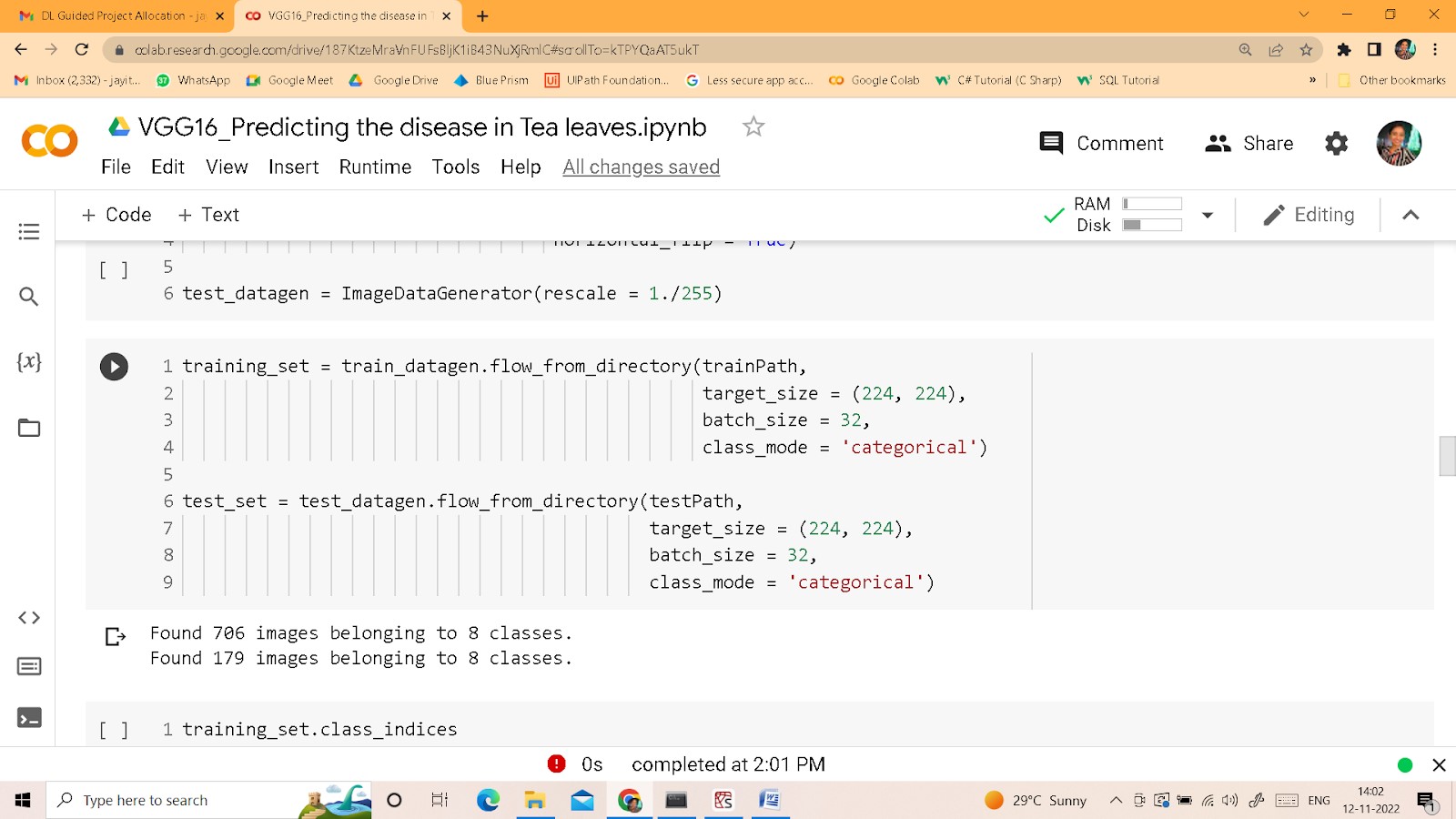
Let us apply ImageDataGenerator functionality to Trainset and Testset by using the following code. For

Training set using flow\_from\_directory function.

This function will return batches of images from the subdirectories Arguments:

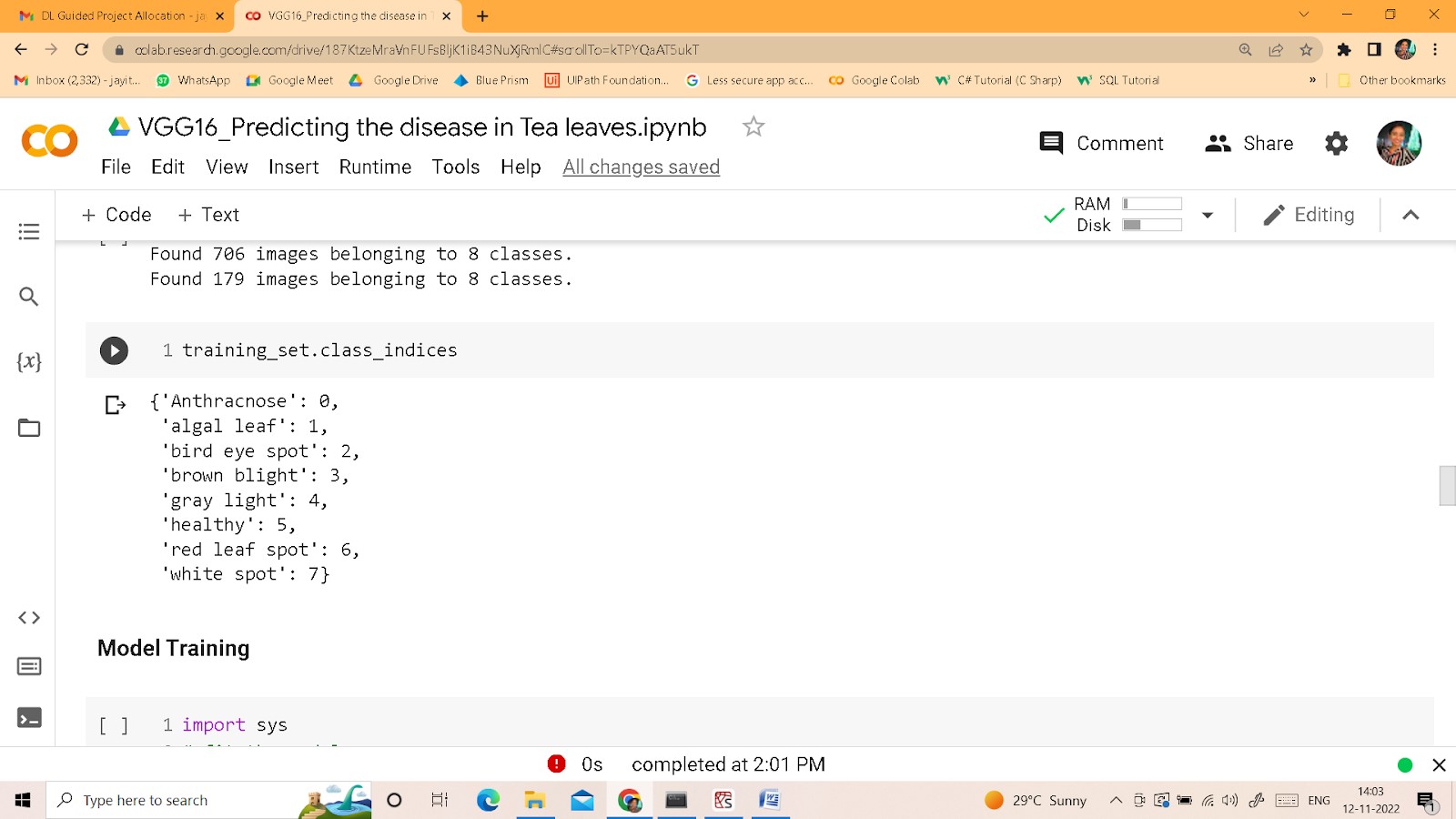
* Directory: Directory where the data is located. If labels are "inferred", it should contain subdirectories, each containing images for a class. Otherwise, the directory structure is ignored.
* batch\_size: Size of the batches of data which is 10.
* target\_size: Size to resize images after they are read from disk.
* class\_mode:
  + ‘int': means that the labels are encoded as integers (e.g. for sparse\_categorical\_crossentropy loss).
  + 'categorical' means that the labels are encoded as a categorical vector (e.g. for categorical\_crossentropy loss).
  + 'binary' means that the labels (there can be only 2) are encoded as float32 scalars with values 0 or 1 (e.g. for binary\_crossentropy).
  + None (no labels).

Loading our data and performing Data Augmentation



We notice that 706 images belong to 8 classes for training and 179 images belonging to 8 classes for testing purposes.

List of classes we have



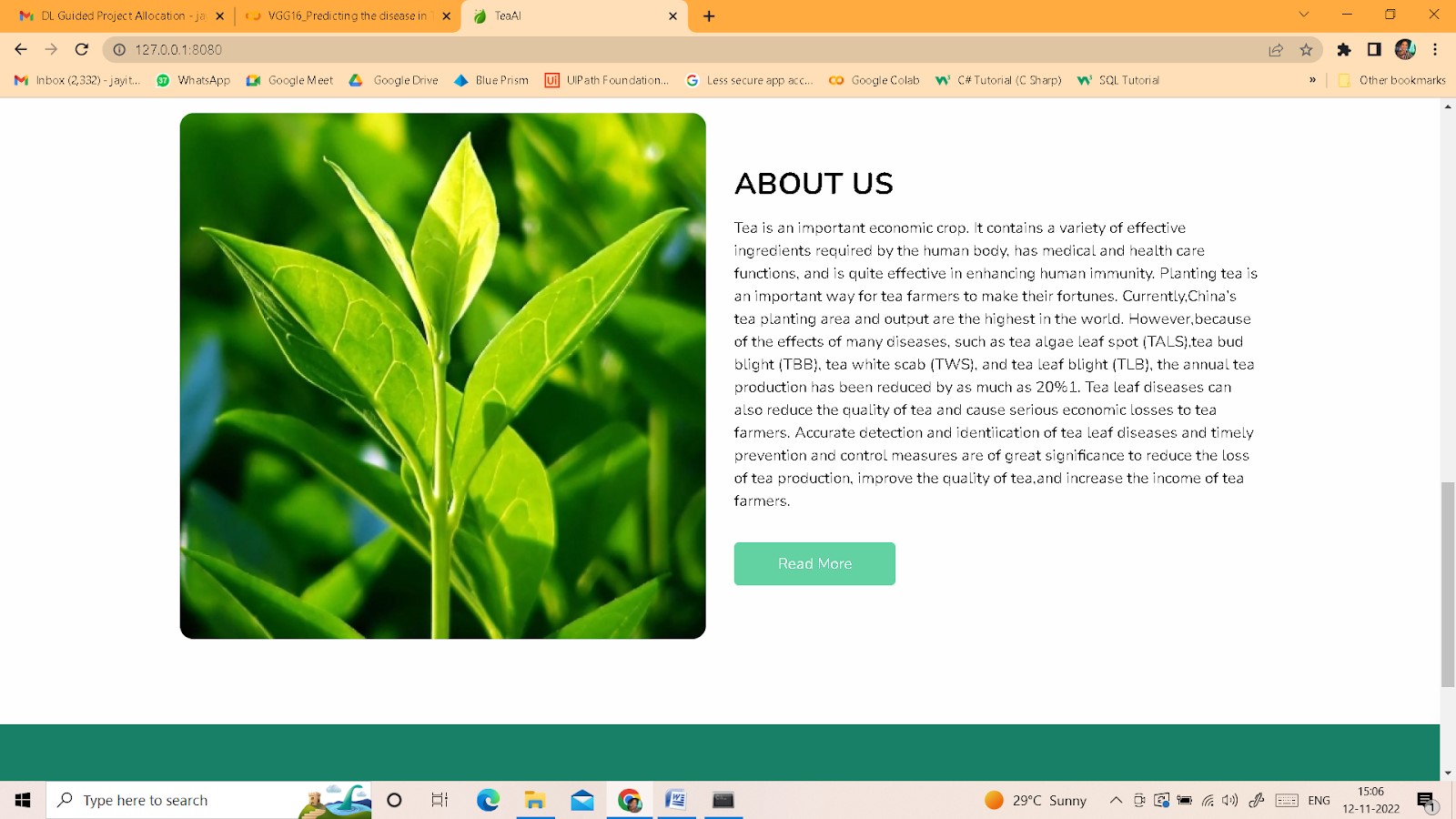
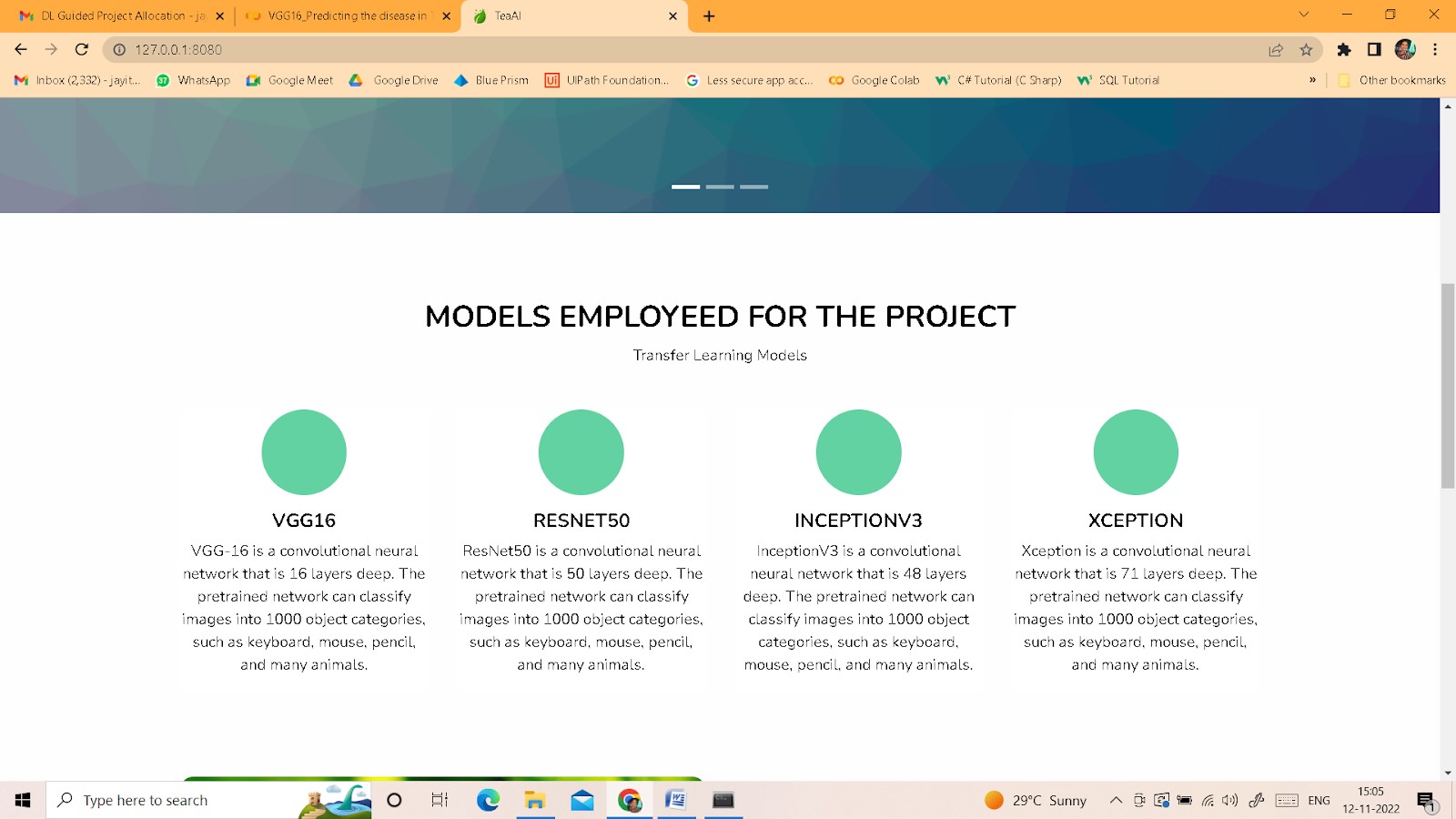
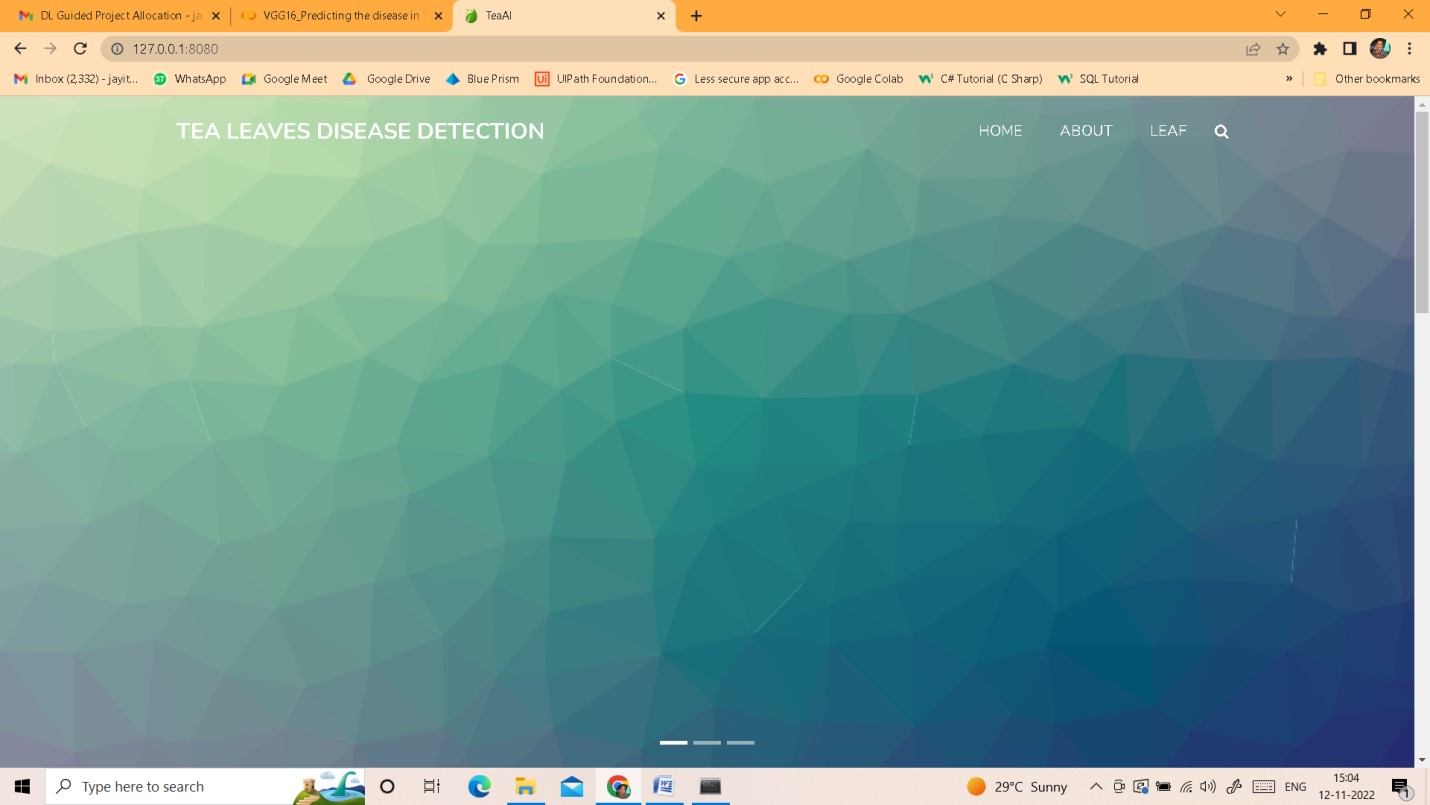
**Create HTML Pages**

* We use HTML to create the front end part of the web page.
* Here, we have created 4 HTML pages- about.html, index.html, teahome.html and teapred.html
* inex.html displays the home page.
* abou.html display the about of the project
* teahome.html displays information about tea disease.
* teapred.html takes the input image and displays the prediction.

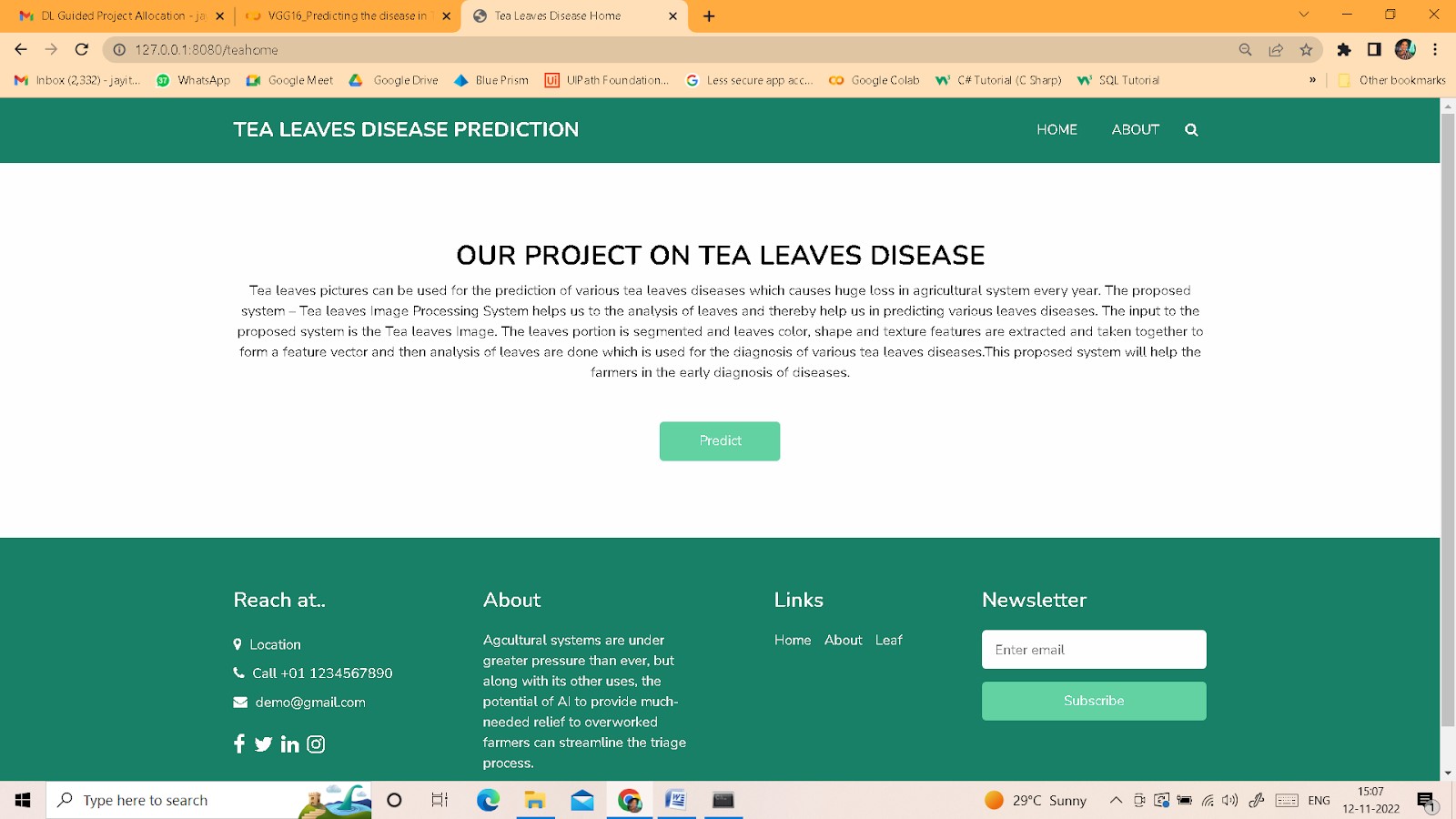
For more information regarding HTML <https://www.w3schools.com/html/>

* We also use JavaScript-main.js and CSS-main.css to enhance our functionality and view of HTML pages.
* Link :[CSS](https://www.w3schools.com/css/) , [JS](https://www.w3schools.com/js/DEFAULT.asp)

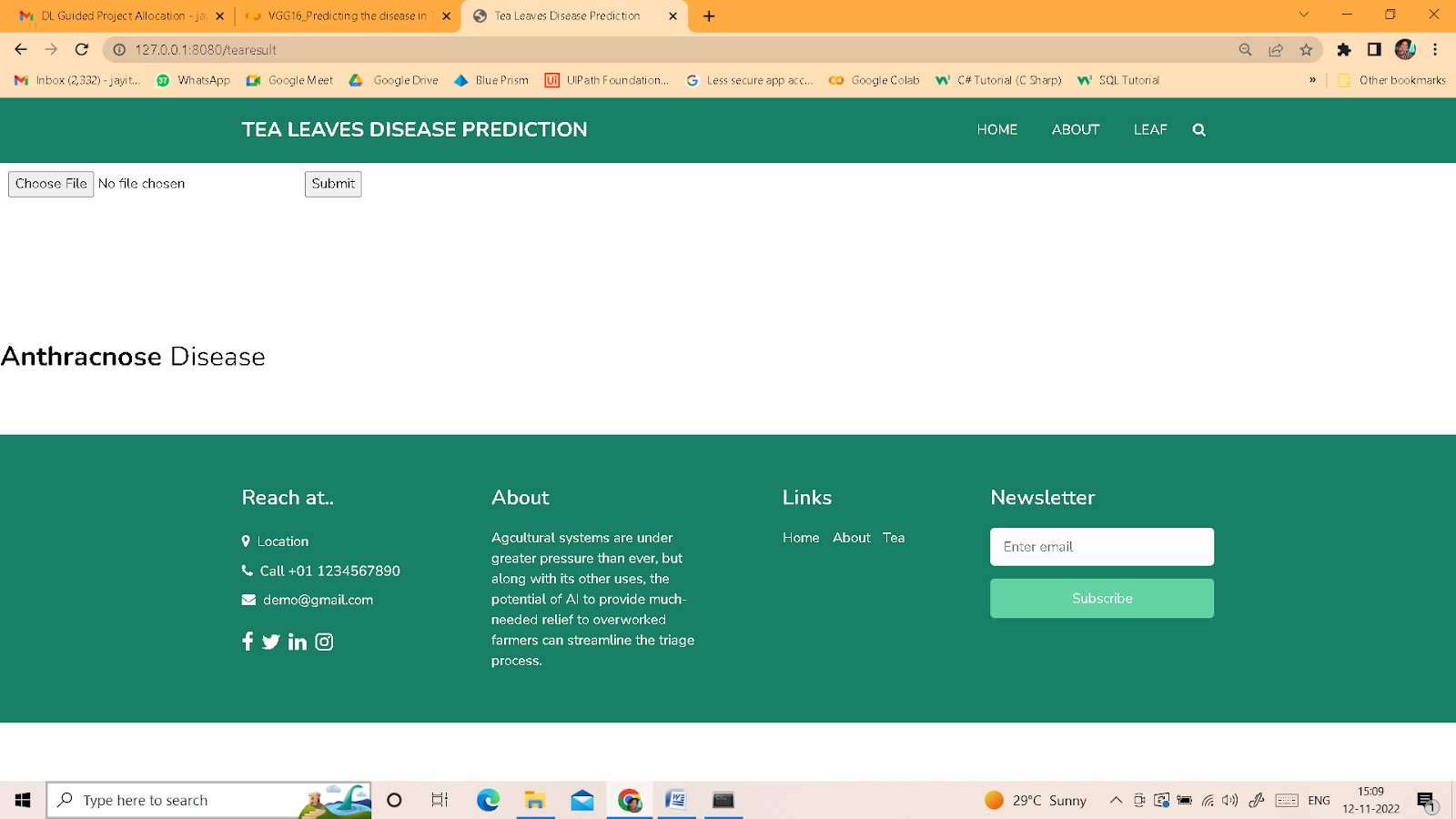
## index.html



### Tea Leaves Disease Home page:-



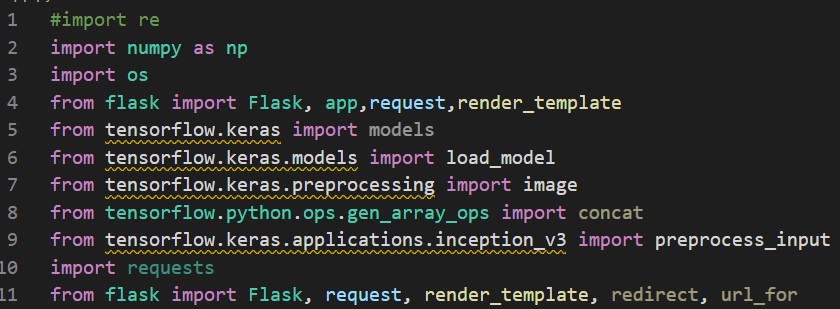
### Tea Leaves Disease Prediction Page:-



### Build Python Code

**Task 1: Importing Libraries**

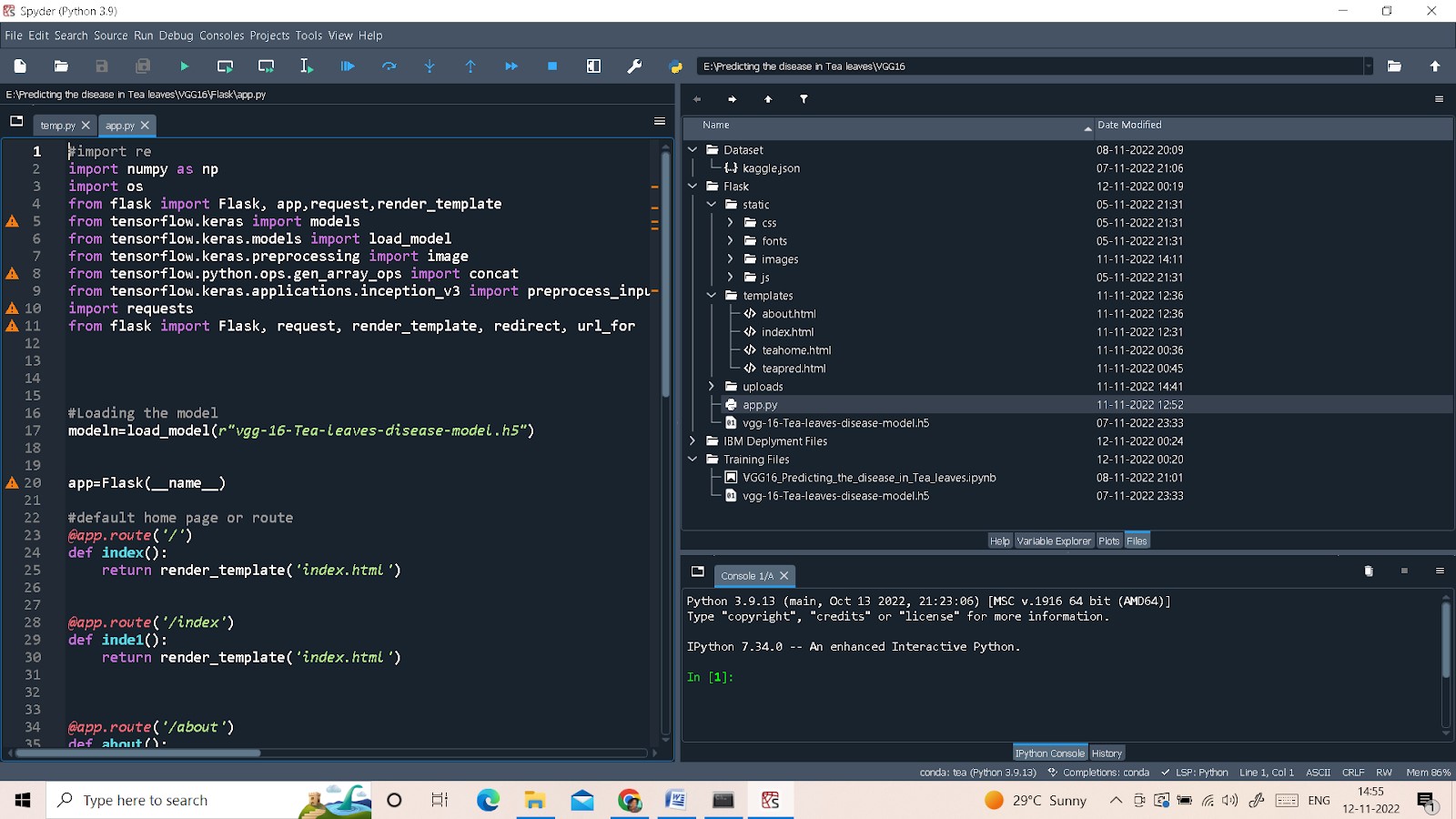
The first step is usually importing the libraries that will be needed in the program.



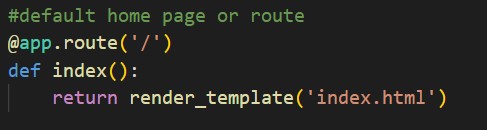
Importing the flask module in the project is mandatory. An object of the Flask class is our WSGI application. Flask constructor takes the name of the current module (\_\_name\_\_) as argument Pickle library to load the model file.

Once after loading the libraries, we need to load the model

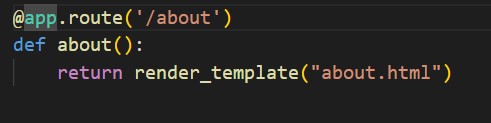
#### Task 2: Creating our flask application and loading our model by using load\_model method



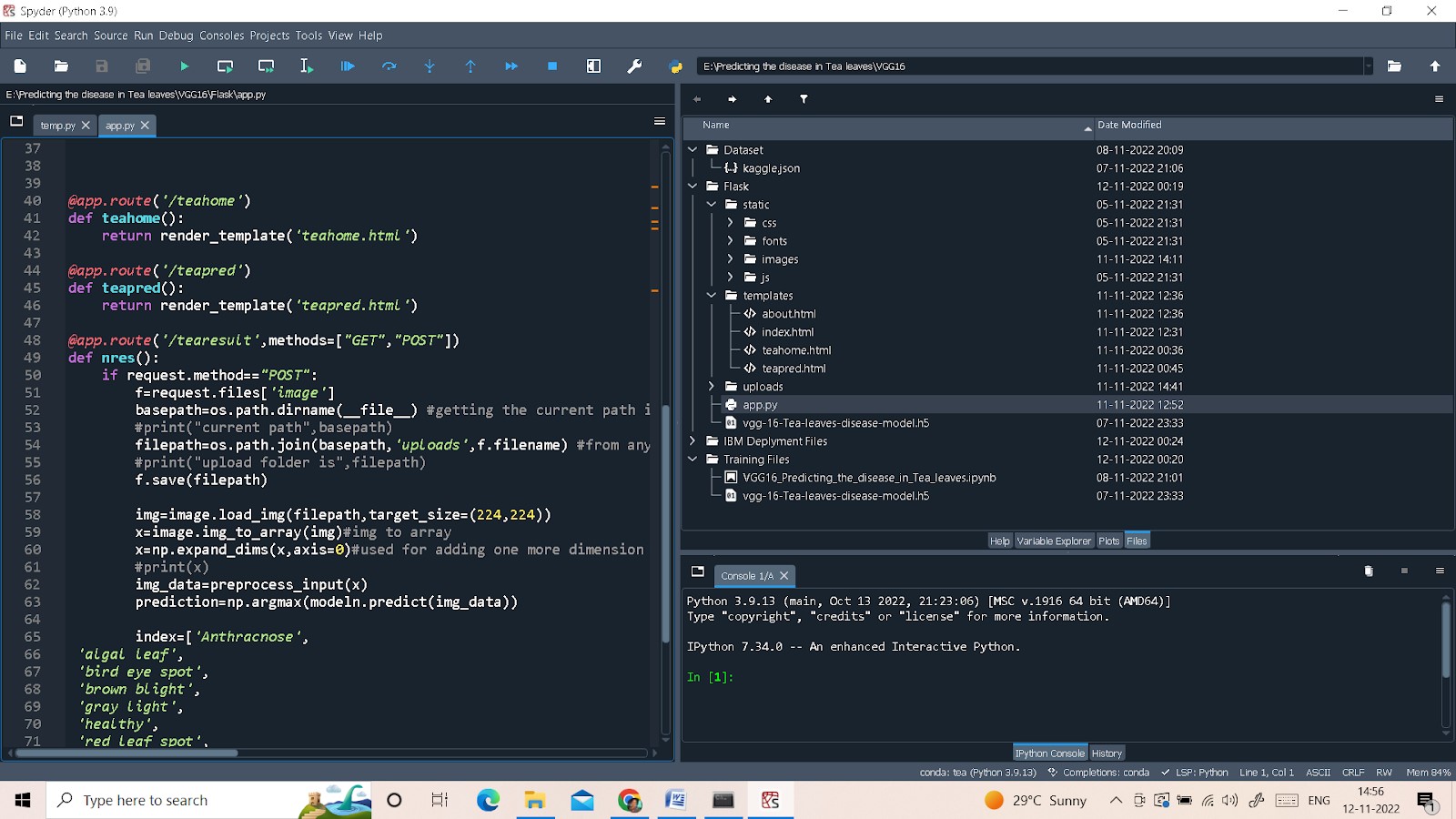
**Default home page**



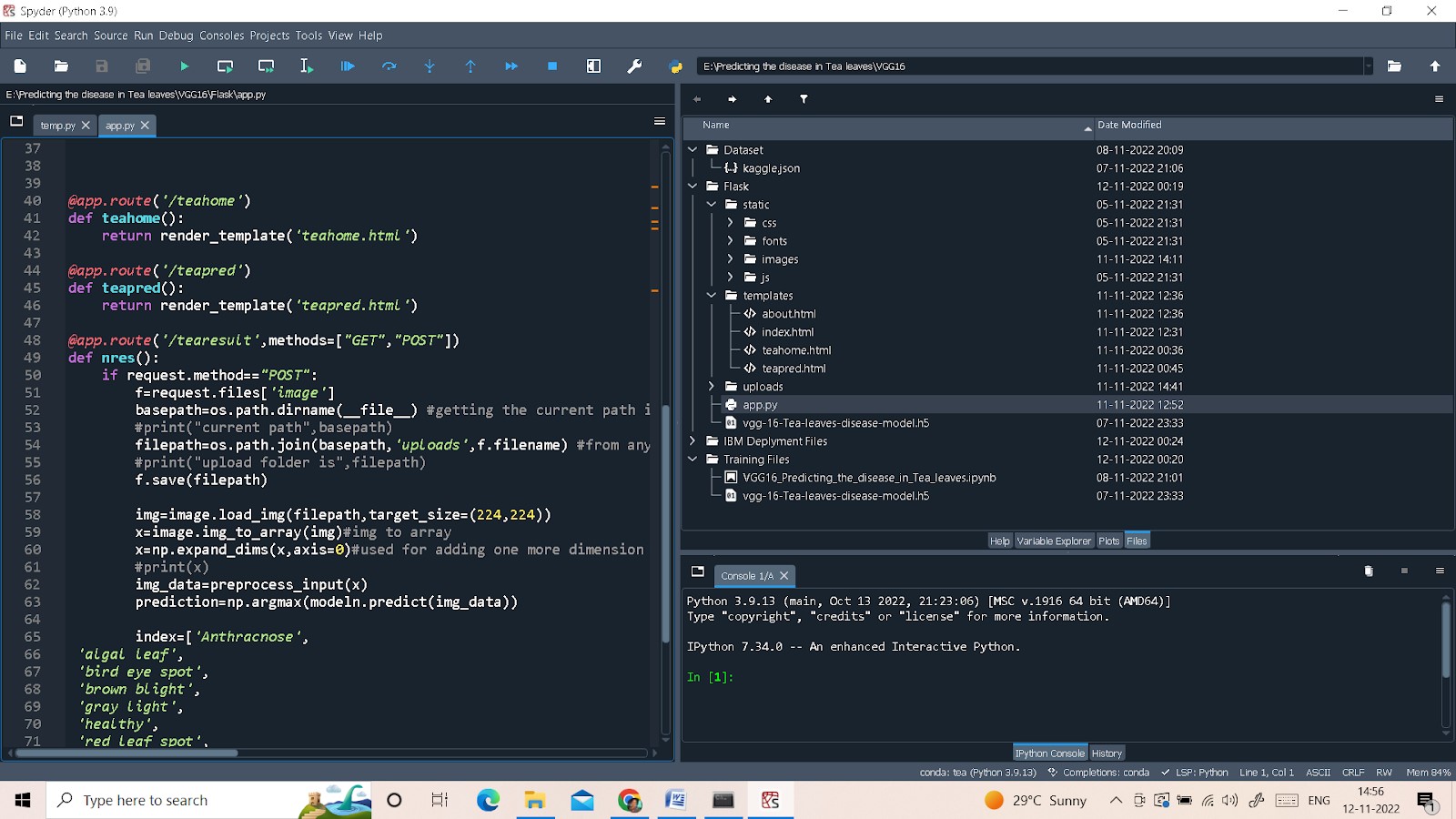
When you click on about on the home page, it will redirect you to the about page



When you click on leaf, it will redirect you to the Tea leaves Disease Home page



When you click on predict, it will redirect you to the Tea predict page



#### Task 3: Routing to the html Page

Here, the declared constructor is used to route to the HTML page created earlier.

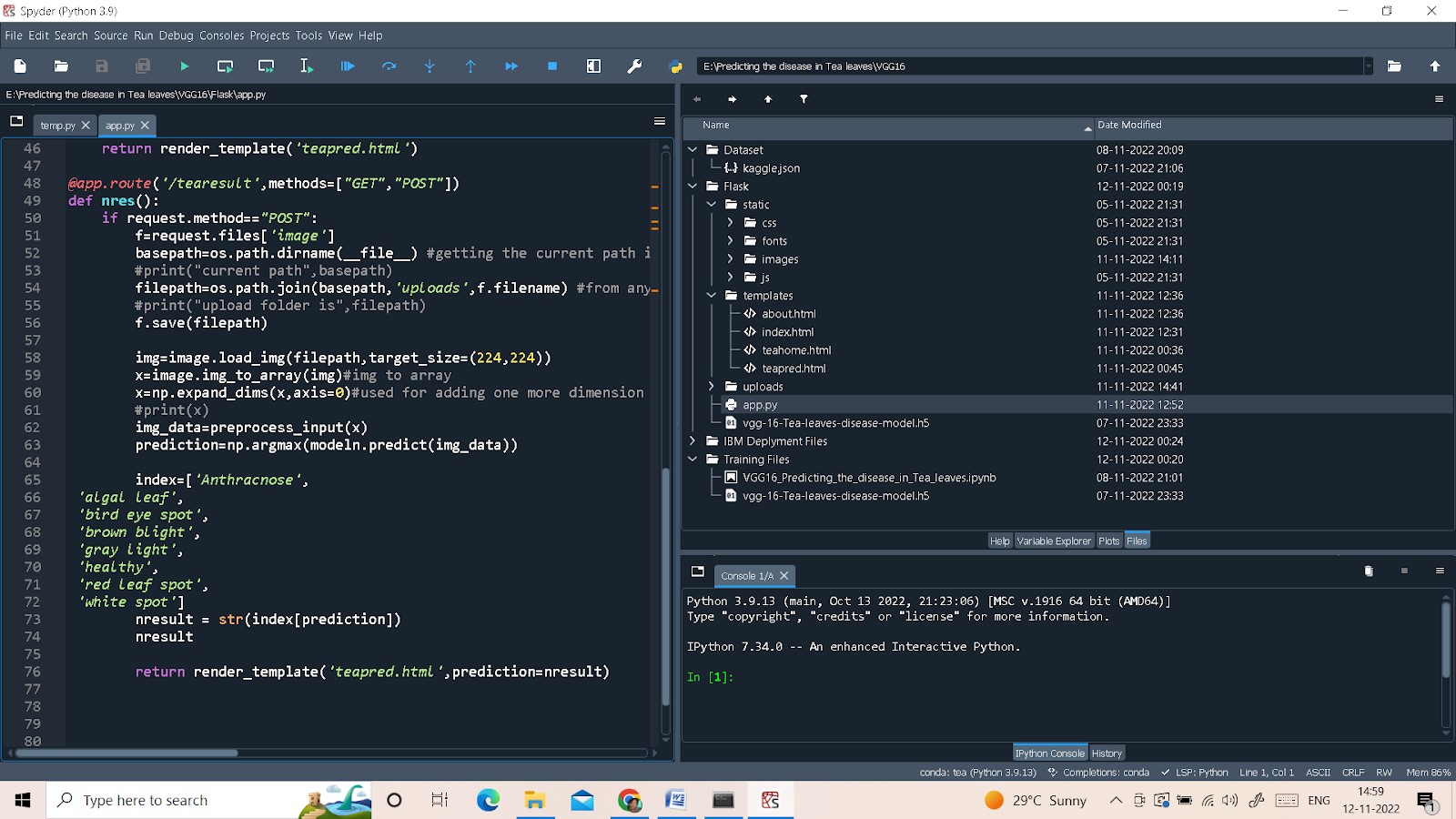
In the above example, ‘/’ URL is bound with the index.html function. Hence, when the home page of a web server is opened in the browser, the html page will be rendered. Whenever you browse an image from the html page this photo can be accessed through POST or GET Method.

Once you click on the Predict in Tea Leaves Disease Home page, it will redirect you to Tea Disease

Prediction page

**Showcasing prediction on UI**

:

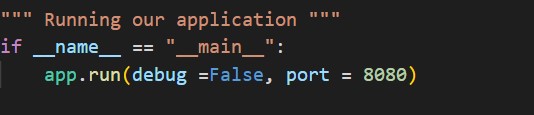


Note: Target size need to change to 224,224 – Because vgg16 takes the input format in that size

Here we are defining a function which requests the browsed file from the html page using the post method. The requested picture file is then saved to the uploads folder in this same directory using the OS library. Using the load image class from Keras library we are retrieving the saved picture from the path declared. We are applying some image processing techniques and then sending that preprocessed image to the model for predicting the class. This returns the numerical value of a class (like 0, 1, 2, etc.) which lies in the 0th index of the variable preds. This numerical value is passed to the index variable declared. This returns the name of the class. This name is rendered to the predict variable used in the html page.

Finally, run the application

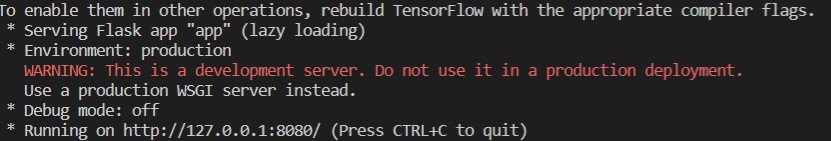
This is used to run the application in a localhost.



### Run The Application

* pen the anaconda prompt from the start menu.
* Navigate to the folder where your app.py resides.
* Now type “python app.py” command.
* It will show the local host where your app is running on http://127.0.0.1.8080/
* Copy that local host URL and open that URL in the browser. It does navigate me to where you can view your web page.
* Enter the values, click on the predict button and see the result/prediction on the web page.

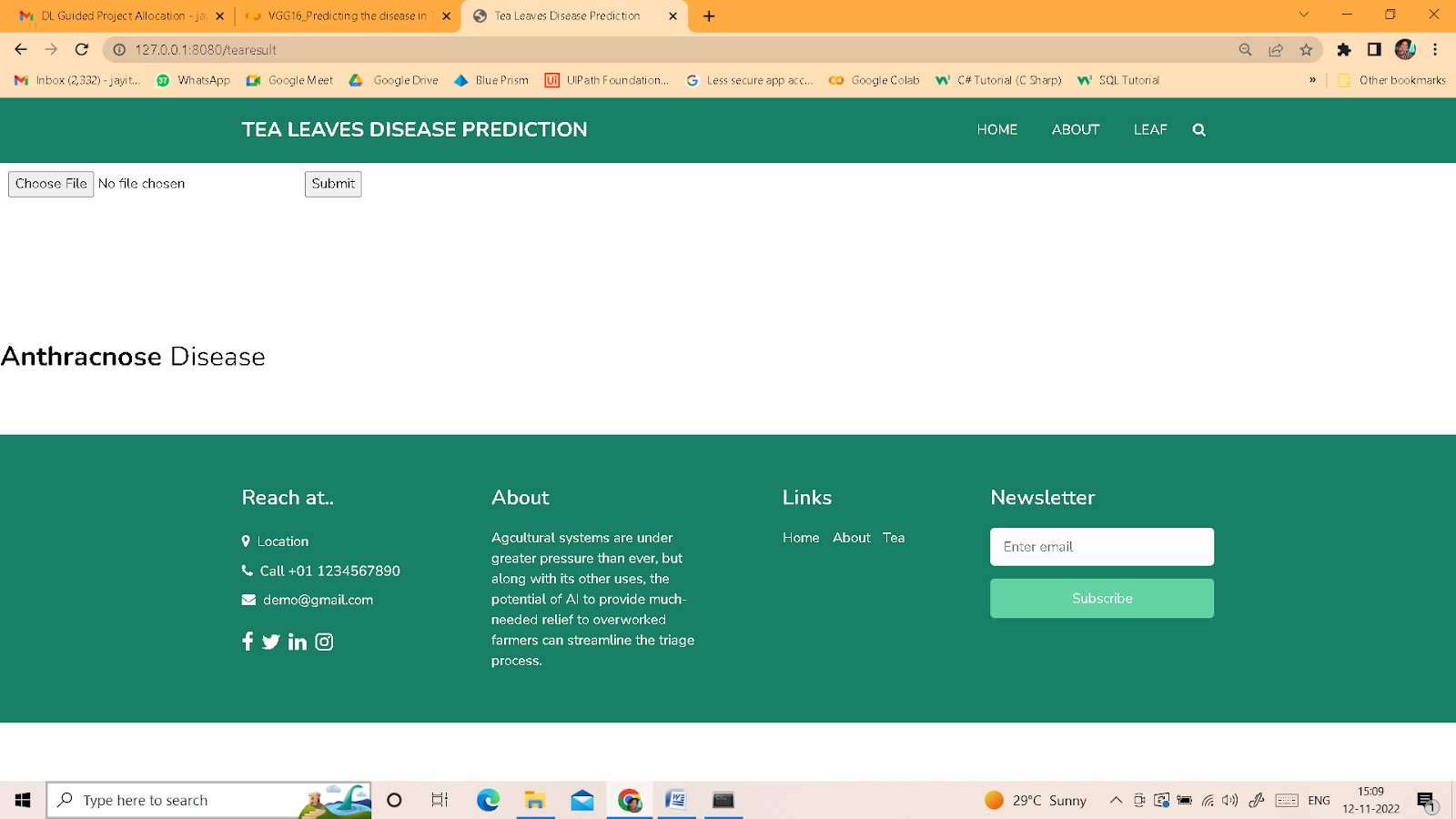
Then it will run on localhost: 8080



Navigate to the localhost (<http://127.0.0.1:8080/>) where you can view your web page.

### Result

Pass the Image by selecting/uploading the image by clicking on Choose Image, That Image will go to model and model predict. It will give the result, That result display in here like this.



# 12 Performance Testing

* Performance Metrics

In order to evaluate the effectiveness of the deep learning model, we will conduct rigorous performance testing. In this section, we will define the performance metrics used and analyze the model's performance. Join us as we assess the performance of our innovation!

The experimental results included four outcomes: true positive (TP), which refers to the accurate detection of individually marked diseased leaves; false positive (FP), which refers to an object that was incorrectly identified as a diseased tea leaf; true negative (TN) which refers to negative samples with a negative system prediction; and false negative (FN) which refers to diseased tea leaves that were overlooked.

Create a Project folder which contains files as shown below

* The Dataset folder contains the training and testing images for training our model.
* We are building a Flask Application that needs HTML pages stored in the templates folder and a python script app.py for server-side scripting
* we need the model which is saved and the saved model in this content is a vgg-16-Tea-leavesdisease-model.
* templates folder contains about.html, index.html, teahome.html, and teapred.html pages. ● An IPYNB file is a notebook document created by Jupyter Notebook.

# 13 Results

After implementing and testing the deep learning model, we will present the results obtained. In this section, we will discuss the accuracy, efficiency, and overall performance of the model. Join us as we unveil the outcomes of our hard work and dedication!

Both RGB and hyperspectral could be used for classifying tea coal disease. The accuracy of the classification models established by RGB imaging using ResNet18, VGG16, AlexNet, WT-ResNet18, WT-VGG16, and WT-AlexNet was 60%, 58%, 52%, 70%, 64%, and 57%, respectively, and the optimal classification model for RGB was the WT-ResNet18. The accuracy of the classification models established by hyperspectral imaging using UVE-LSTM, CARS-LSTM, NONE-LSTM, UVE-SVM, CARS-SVM, and NONE-SVM was 80%, 95%, 90%, 61%, 77%, and 65%, respectively, and the optimal classification model for hyperspectral was the CARS-LSTM, which was superior to the model based on RGB imaging.

# 14 Advantages and Disadvantages

## Advantages

We will explore the numerous advantages of using a deep learning model for detecting diseases in tea leaves. From increased accuracy to scalability, the benefits are vast and significant.

Machine learning algorithms have been applied to find out several leaf stresses as well as helpful in the identification of distinct species. This computational model is a part of ML and pattern recognition algorithms that are also important in the field of leaf disease identification.

The higher accuracy of the CNN model for plant disease classification has proofed to be the best then all other kinds of ML and DL methods. Studies have shown that CNNs can achieve high accuracy rates in the range of 99-99.2% in classifying images of plant leaves affected by diseases and pests.

## Disadvantages

No solution is without its drawbacks. In this section, we will discuss the limitations and potential challenges associated with the implementation of the deep learning model. Join us as we address the potential pitfalls!

Loss of crops from plant diseases may also result in hunger and starvation, especially in less-developed countries where access to disease-control methods is limited and annual losses of 30 to 50 percent are not uncommon for major crops.

Plant viruses, being a highly contagious pathogen, challenge the security of our food supply. Plant diseases, including viruses, result in the loss of $220 billion in agricultural food crops around the globe. However, viruses, unlike other plant diseases and pests, cannot be eradicated by chemical spraying.

# 15 Conclusion

* Summary of the project and its significance
* Potential future developments and improvements

In conclusion, the development of a deep learning model for detecting diseases in tea leaves is a promising innovation that has the potential to revolutionize the tea industry. Join us as we summarize the project's significance and explore the exciting possibilities for future advancements.

Now that we have trained our model, let us build our flask application which will be running in our local browser with a user interface.

In the flask application, the input parameters are taken from the HTML page; these factors are then given to the model to predict the cost estimation for damage on the HTML page to notify the user. Whenever the user interacts with the UI and selects the “Image” button, the next page is opened where the user chooses the image and predicts the output.

This study revealed the classification potential of tea coal disease based on RGB and hyperspectral imaging, which can provide an accurate, non-destructive, and efficient classification method for monitoring tea coal disease.

# 16 Future Scope

The future holds boundless opportunities for further refinement and expansion of the deep learning model for detecting diseases in tea leaves. In this section, we will discuss the potential avenues for research and development, paving the way for exciting breakthroughs in tea leaf disease detection.

# 17 Appendix

* Interpretation of the model's performance and limitations
* Comparison with other disease detection methods for tea leaves

In the appendix, we provide additional information and insights. We will interpret the performance of the deep learning model and discuss its limitations. Furthermore, we will compare our model with other existing disease detection methods for tea leaves. Join us in this comprehensive examination!