JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

# Coffee Bean Classification Using Deep Learning

**Sahana.L Assistant Professor, Department of Computer Science**
**St.Joseph's First Grade College Jayalakshmipuram , Mysore-570012**

## Introduction

The definition of Deep learning is that it is the branch of machine learning that is based on artificial neural network architecture. An artificial neural network or ANN uses layers of interconnected nodes called neurons that work together to process and learn from the input data. In a fully connected Deep neural network, there is an input layer and one or more hiddenlayers connected one after the other. Each neuron receives input from the previous layer neurons or the input layer. The output of one neuron becomes the input to other neurons in thenext layer of the network, and this process continues until the final layer produces the output of the network. The layers of the neural network transform the input data through a series of nonlinear transformations, allowing the network to learn complex representations of the inputdata. In the fast-evolving era of artificial intelligence, Deep Learning stands as a cornerstone technology, revolutionizing how machines understand, learn, and interact with complex data. At its essence, Deep Learning AI mimics the intricate neural networks of the human brain, enabling computers to autonomously discover patterns and make decisions from vast amountsof unstructured data. This transformative field has propelled breakthroughs across various domains, from computer vision and natural language processing to healthcare diagnostics andautonomous driving.

Training deep neural networks involves optimizing a loss function that measures the discrepancy between the predicted and actual outputs. This process typically uses gradient- based optimization methods, such as Stochastic Gradient Descent (SGD) and its variants (e.g.,Adam, RMSprop). Deep Neural Networks represent a powerful and versatile tool in the field of artificial intelligence. Their ability to learn complex patterns and representations has led to significant breakthroughs across various domains. As the field continues to evolve, addressingthe challenges and ethical implications will be crucial to harnessing the full potential of DNNsfor beneficial and equitable outcomes.

## Overview

Coffee is a beloved beverage enjoyed worldwide, but have you ever wondered about the specific type of bean hidden within your cup. This project introduces a novel system that utilizes the power of image recognition to unveil the secrets of your coffee beans. By combininguser-friendly image input with cutting-edge Visual Geometry Group (VGG19) technology, users can gain valuable insights into the origin, flavour profile, and roasting recommendationsfor their beans. This system empowers coffee enthusiasts of all levels. Simply register or login,upload an image of your coffee beans, and let the intelligent algorithms do the work. Througha combination of image pre-processing, feature extraction, and advanced VGG-19 model comparison, the system accurately classifies your beans and provides a detailed description. Imagine identifying rare Arabica beans or perfectly roasting a batch of Robusta for a richer flavour. This innovative coffee bean classification system empowers you to take your coffee experience to the next level.

## Objective

- Automated Classification: Leverage VGG19 model to accurately identify the type of coffeebean present in a user-uploaded image.
- In-Depth Bean Analysis: Provide detailed descriptions beyond just the bean type. This includes information on origin, flavour profile, and roasting recommendations for a more comprehensive coffee experience.
- User-Friendly Interface: Design a system that is easy to navigate for coffee enthusiasts of all levels. This includes a streamlined registration/login process, intuitive image input, andclear presentation of classification results.
- Performance Optimization: Evaluate and potentially compare different VGG models to ensure the system delivers the most accurate and reliable coffee bean classification.

## Existing System

- Visual Inspection: Experts would meticulously examine the physical characteristics of thebeans, such as size, shape, and colour. Factors like uniformity and the presence of defects also played a role in classification.
- Sensory Evaluation: The aroma, acidity, body, and flavour profile of brewed coffee were crucial for determining the bean type. Experienced tasters, often referred to as "cuppers," would meticulously evaluate these qualities through a standardized cupping process.
- Geographic Origin: Knowledge of the bean's origin played a significant role in traditional classification. Different regions are known for cultivating specific coffee bean varieties with distinct characteristics.

- Grading Systems: Various commercial grading systems existed, often based on screen sizeand defect count. While offering a basic level of classification, these systems lacked the precision and detail provided by expert analysis.

## Disadvantages

- Computational Intensity: CNN models, especially deep architectures, can becomputationally intensive, requiring significant processing power and time for inference.
- Data Dependency: The success of CNN models heavily relies on the availability of diverse and well-labeled datasets. Insufficient or biased data may affect the model's generalization to different scenarios.
- Limited Interpretability: CNNs are often considered as "black-box" models, making it challenging to interpret the decision-making process and understand how specific features contribute to predictions.
- Over fitting Risk: Transfer learning, while beneficial, may lead to overfitting if the pre-trained models are not appropriately fine-tuned to the specific characteristics of the riceplant health domain.
- Lack of Real-time Processing: In certain implementations, the computational demandsof the CNN models might hinder real-time processing, limiting the system's responsiveness for time-sensitive applications.

## Proposed system

In the proposed system, we aim to revolutionize coffee bean classification assessment, specifically focusing on the classification of roasted beans. The workflow begins with the inputof coffee bean images, where a meticulous pre-processing stage ensures normalized and appropriately resized images. Subsequently, the system identifies Regions of Interest (ROIs) within the images, concentrating on areas where pest-induced symptoms manifest prominently. This ROI extraction process optimizes computational efficiency and directs the model'sattention to the most relevant features. This system goes beyond simply identifying the bean type. It delves deeper, providing a comprehensive description that includes the bean's origin, flavor profile (think fruity or nutty notes), and even roasting recommendations to unlock its full potential. Feature extraction is conducted using the powerful VGG19 model, renowned fortheir ability to capture intricate hierarchical features in images. These pre-trained models act as robust feature extractors, deciphering the distinctive patterns associated with different features of coffee beans. The proposed system holds significant promise in advancing coffee bean classification, providing farmers with a comprehensive tool for precise identification of coffee beans. By leveraging the capabilities of VGG19 and SVM, the system aims to enhancethe accuracy and efficiency of bean-related classification, ultimately contributing to proactiveand effective agricultural management practices.

## Advantages

- VGG19 is a pre-trained deep learning model that has been trained on a large-scale image dataset, such as ImageNet. Leveraging pre-trained weights and learned featurescan expedite the training process and improve the model's performance
- VGG19 has a deep architecture comprising multiple layers, allowing it to learn hierarchical representations of input images.
- The VGG19 architecture supports transfer learning, allowing the model to adapt its learned representations to the task of classifying coffee beans with accuracy.
- Applicability Across Varieties: The robust feature extraction capability of VGG19 model allows the system to be versatile across different coffee bean varieties.

## LITERATURE SURVEY

Every Software development requires the survey process. The Survey process is neededto get the requirement for the software. The Survey also consists of studying the present systemand also studying about the tools needed for the development of the software. A proper understanding of the tools is very much essential. Following is an extract of the information ofthe material collected during literature survey. Literature survey is a methodology of identifying the problems in the existing system through research and proposing the development of the system to solve the problems of existing system.

### Survey Papers

In this chapter a brief discussion is done based on the various methods and techniqueswhich are used in Plant Leaf Disease Detection. This survey done will be used to implement the proposed by considering these problems

**[1] TITLE:** Classification of coffee bean varieties based on a deep learning approach
**AUTHORS:** D. Buonocore, M. Carratù, M. Lamberti
**YEAR:** 2022

a coffee beans fraud detection based on a deep learning approach is proposed, which has beenachieved after classifying the two coffee varieties to distinguish them in a real-time industrial scenario. The coffee bean quality is typically defined by visual inspection, which is subjective,needing significant effort and time, and susceptible to fault detection. For these reasons, a different method is required to be objective and precise. Therefore, object detection techniqueswere employed to automatically classify the coffee bean samples according to their specie using an own dataset consisting of over 2500 coffee beans. Furthermore, a convolutional neuralnetwork (CNN) based on the YOLO algorithm was employed to categorize the coffee beans automatically. The result of this study has revealed that the object detection technique could beused as an effective method to classify coffee bean species and discover food fraud.

**[2] TITLE**: Multi-Label Classification of Defective Green Coffee Bean Images UsingEfficientNet Deep Learning Mode

**AUTHORS:** Hira Lal Gope, Hidekazu Fukai and Renshu Aoki

**Year:** 2022

As the grade of green coffee beans largely depends on total number of defective beans in a given quantity of sample beans, removing defective beans is important for ensuring their quality and market price. On the other hand, the prevalent sorting method, i.e., manual handpicking, can be affected by human condition and is time- consuming. Studies have been proposed to classify defective fruits, vegetables, and beans by exploiting image processing andmachine learning techniques so far. However, general single-label classification algorithms arenot suitable for sorting coffee beans because some of them have multiple properties, e.g., broken and fade. In this study, we propose a deep neural network model with modifications tothe EfficientNet-B1 by putting branches, which correspond to each defect after feature extraction layers, to classify coffee beans with multi-label. The proposed multi-branch EfficientNet-B1 model significantly improved overall performance, with an f1-score of 0.8229, compared to single EfficientNet-B1 model.

**[3] TITLE**: Classification of types Roasted Coffee Beans using Convolutional Neural NetworkMethod

**AUTHORS:** Halifa Sekar Metha, Kusrini, Dhani Ariatmanto

**YEAR:** 2022

In the current digital era, the role of technology in the agricultural industry is very necessary toincrease yields which can have an impact on the productivity and welfare of farmers. Coffee isa drink that has been very popular for many years. Due to the high demand for coffee beans, this research aims to develop a system that can classify types of roasted coffee beans based onimages using the Convolution Neural Network (CNN) method. Coffee bean processing is the most important stage in the coffee industry, classifying coffee beans often requires more in- depth knowledge and extensive experience regarding coffee beans. Therefore, this system canbe a more effective solution. The author collects a dataset containing types of roasted coffee beans, then the Convolutional Neural Network (CNN) can analyze in the form of visual patternseach type of coffee bean. This implementation is expected to help the coffee industry identifycoffee beans quickly and accurately.

**[4] TITLE**: Classification of coffee bean species using image processing, artificial neuralnetwork and K nearest neighbors

**AUTHORS:** Edwin R. Arboleda, Arnel C. Fajardo, Ruji P. Medina

**YEAR:** 2022

The quality of coffee beans differs from each other based on the geographic locations of its sources. The coffee bean quality is conventionally determined by visual inspection, which is subjective, requiring considerable effort and time and prone to error. This calls for thedevelopment of an alternative method that is precise, non-destructive and objective. This paperwas conducted with the objective of developing an appropriate computer routine that can characterize coffee beans from the different towns of Cavite, Philippines. Imaging techniqueswere employed to automatically classify the coffee bean samples according to their specie. Important coffee bean features based in morphology such as area of the bean, perimeter, equivalent diameter, and percentage of roundness were extracted from 195 training images and60 testing images. Artificial neural network (ANN) and K nearest neighbor (KNN) were employed to automatically categorize the coffee beans.

## REQUIREMENT SPECIFICATIONS

Software Requirement Specification (SRS) is essential information, which shapes the establishment of the software development process. SRS records the necessities of a frameworkas well as has a depiction of its significant components.

The focus in this stage is one of the users of the system and not the system solutions. The resultof the requirement specification document states the intention of the software, properties and constraints of the desired system. SRS constitutes the understanding amongst customers and designers with respect to the substance of the product that will be created. SRS should be preciseand totally signify the framework prerequisites as it makes a colossal commitment to the generaldevelopment plan.

One of the most essential information is SRS (Software Requirement Specification). It gives the detailed information about establishment of software development process. It records the important necessities of the frame work also holds the depiction of the important components.These things will be in the IEEE standards. The recommendations would shape the explanationbehind giving clear image of the item to be made filling in as measure for execution of an understanding among client and the developer. One of the important steps involved in the development process is system requirements. This SRS(Software Requirement Specification) is followed after resource analysis phase. Its main task is to decide what a software product does. In this stage the main focus is the user, and not the system solution. SRS(Software Requirement Specification) gives the results like intention of the software, properties and constraints of the desired system. The main advantage of SRS(Software Requirement Specification) is that it gives a clear understanding among the clients and the product developers with respect to the product that is developed. SRS(Software Requirement Specification) which is documented should accurate and the prerequisites of the frame work should be signified as it makes colossal commitment to the general development plan process.

### Functional Requirements Specification

**Image Input Handling:**

- The system must accept images of coffee beans as input, ensuring compatibility withvarious image formats commonly used in agricultural monitoring.

**Image Pre-Processing:**

- Normalize and resize input images to a standard format suitable for analysis.
- Enhance image quality to facilitate better feature extraction, including techniques suchas noise reduction and contrast adjustment.

**ROI Extraction:**

- Automatically identify and extract Regions of Interest (ROIs) within the images wherepest-induced symptoms are likely to appear.
- Ensure the ROI extraction process is efficient and accurately targets relevant areas.

**Feature Extraction:**

- Utilize VGG19 model to extract detailed hierarchical features from the pre-processed images.
- The system should extract key features from the pre-processed image that are relevantfor coffee bean classification. These features could include colour distribution, shape characteristics, and texture patterns.

**Database Integration:**

- The system might integrate with a database to store information about various coffee bean types, including origin details, flavour profiles, and roasting recommendations. This database would be used to generate detailed descriptions based on the classification results.

## Non-functional Requirements

**Usability:**

- The user interface should be intuitive and user-friendly for people with varying levelsof technical expertise.
- Clear instructions and guidance should be provided for registration, image upload, and interpreting results.

**Performance:**

- The system should be able to process image uploads and deliver classification resultsquickly and efficiently.
- Response times should be optimized to avoid frustrating wait times for users.

**Security:**

- The system should implement robust security measures to protect user data. Thisincludes:

- Secure storage of user credentials (e.g., passwords) using encryption techniques.
- Measures to prevent unauthorized access to user accounts or system functionalities.
- Protection against potential cyberattacks or data breaches.

**Reliability:**

- The system should be highly available and operational for most of the time. This minimizes downtime and ensures users can consistently access the classificationservice.
- The system should be designed to handle potential errors gracefully and provide informative messages to users if issues arise.

**Scalability:**

- The system should be able to handle an increasing number of users and image uploadswithout significant performance degradation. This ensures smooth operation as the userbase grows.

**Maintainability:**

- The system should be designed with maintainability in mind. This allows for future updates, bug fixes, and integration of new features without extensive codemodifications.

## System Requirements

### Minimum Hardware Requirements

| | | |
|---|---|---|
| • Processor | : | Intel i5 2.53GHz |
| • Hard Disk | : | 30GB |
| • RAM | : | 8 GB or above |

### Software Requirements

| | | |
|---|---|---|
| • Operating system | : | Windows 8 and above |
| • Coding Language | : | Python |
| • Version | : | 3.7 & above |
| • IDE | : | IDLE |
| • Framework | : | Flask |
| • Frontend | : | HTML, CSS, Bootstrap, JavaScript |

# SYSTEM DESIGN

This chapter delves into the architectural blueprint of our innovative coffee bean classificationsystem.

Here, we'll explore the intricate components that work together to unlock the secrets hidden within your coffee beans through the power of image recognition.

- User-Centric Interface: A user-friendly web platform or mobile app provides a seamless experience. This includes intuitive registration, login options, and a clear interface for uploading coffee bean images.
- Intelligent Image Processing Pipeline: Working behind the scenes, a sophisticated pipeline takes center stage. Uploaded images undergo pre-processing to ensure consistency, followed by feature extraction – identifying key characteristics like colour,shape, and texture. These features become the language for the next stage.
- The Power of CNNs: At the heart of the system lies a powerful Convolutional Neural Network (CNN) model. Trained on a vast dataset of labelled coffee bean images, this model acts as a master decoder, identifying intricate patterns and accurately classifyingthe bean type in your picture.
- Database Integration: The system might integrate with a comprehensive database containing information about various coffee bean types. This database feeds the detailed descriptions displayed alongside the classification results. Imagine accessing information on origin, flavour profile, and even roasting recommendations – all based on your uploaded image.
- Security remains paramount throughout the design process. Robust measures are implemented to safeguard user data and ensure the integrity of the entire classificationprocess. Scalability is also considered, anticipating a potential growth in the user base.

By meticulously outlining the system design, this chapter lays the foundation for a secure, user-friendly, and scalable coffee bean classification system. This paves the way for a tool that empowers coffee enthusiasts to embark on a coffee exploration journey, unlocking the hiddenpotential within their beans.

## System Architecture

Dataset collectio
SIGN IN
Register & Login by user
Uplo

Fig 4.2: System Architecture

## Collection of Datasets

- We are going to collect datasets for the prediction from Kaggle.com
- The data sets consist of many classes.

**Data Pre-Processing**

- In data pre-processing we are going to perform some image pre-processing techniqueson the selected data
- And Splitting data into train and test

**Data Modelling**

- The spitted train data are passed as input to the VGG19 algorithm, which helps intraining.

**Build Model**

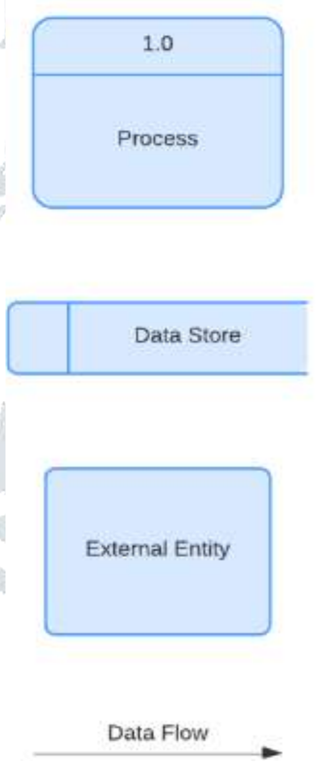
- Once the data is trained and if it showing the accuracy rate as high, then we needto build model file

## Data Flow Diagram

DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. Structure of DFD allows starting from a broad overview and expand it toa hierarchy of detailed diagrams. DFD has often been used due to the following reasons:

- Logical information flow of the system
- Determination of physical system construction requirements
- Simplicity of notation
- Establishment of manual and automated systems requirements

## Basic Notation:



•

**Fig 4.3.1: Data Flow Diagram Basic Notation**

- **Process:** any process that changes the data, producing an output. It might perform computations, or sort data based on logic, or direct the data flow based on business rules. A short label is used to describe the process, such as “Submit payment.”

- **Data store:** files or repositories that hold information for later use, such as a database table or a membership form. Each data store receives a simple label, such as “Orders.”
- **External entity:** an outside system that sends or receives data, communicating with thesystem

being diagrammed. They are the sources and destinations of information entering or leaving the system. They might be an outside organization or person, a computer system or a business system. They are also known as terminators, sources andsinks or actors. They are typically drawn on the edges of the diagram

- **Data flow:** the route that data takes between the external entities, processes and datastores. It portrays the interface between the other components and is shown with arrows, typically labelled with a short data name**.**

**DATAFLOW DIAGRAM**

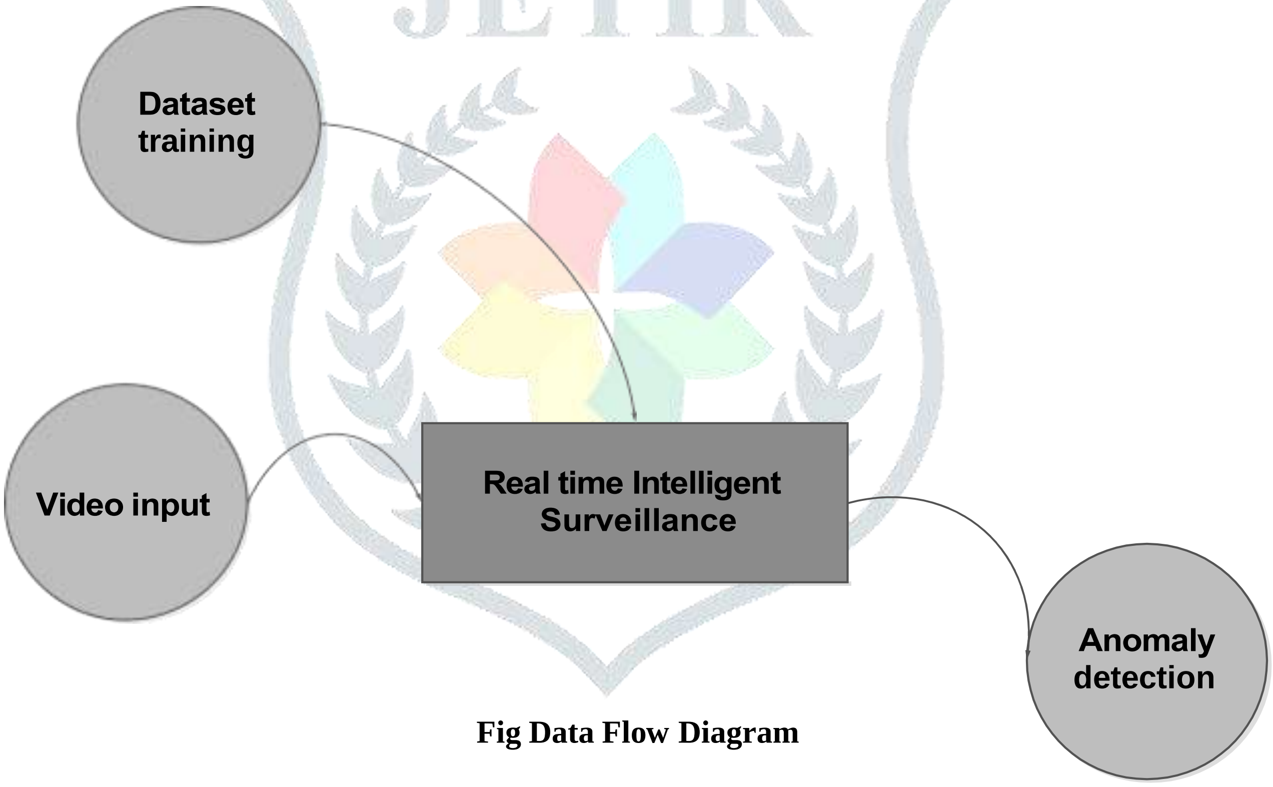


**Fig Data Flow Diagram**

## Use-Case Diagram

Use case diagram is a graph of actors, a set of use cases enclosed by a system boundary, communication associations between the actor and the use case. The use case diagram describes how a system interacts with outside actors; each use case represents a piece of functionality that a system provides to its users. A use case is known as an ellipse containingthe name of the use case and an actor is shown as a stick figure with the name of the actorbelow the figure.

The use cases are used during the analysis phase of a project to identify and partition system functionality. They separate the system into actors and use case. Actors represent roles thatare played by user of the system. Those users can be humans, other computers, pieces of hardware, or even other software systems.

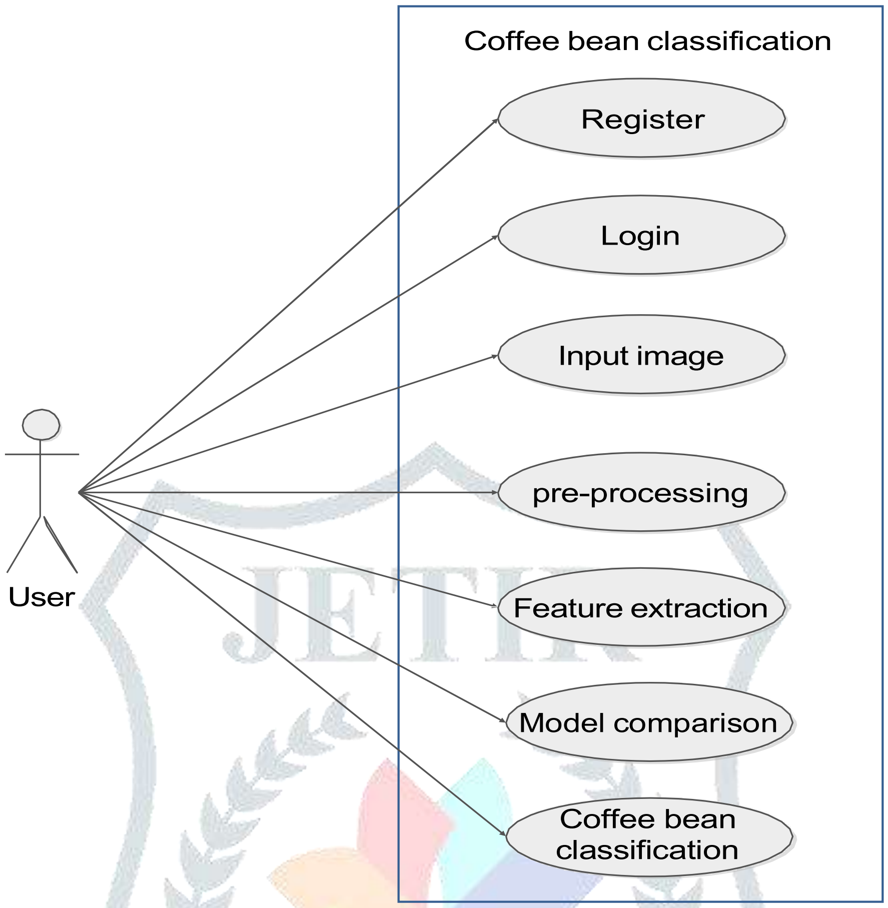


**Fig 4.4: Use Case Diagram**

## Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are sometimes called **event diagrams**, **event scenarios.**UML sequence diagrams are used to represent or model the flow of messages, events and actions between the objectsor components of a system. Time is represented in the vertical direction showing the sequence of interactions of the header elements, which are displayed horizontally at the topof the diagram Sequence Diagrams are used primarily to design, document and validate thearchitecture, interfaces and logic of the system by describing the sequence of actions that need to be performed to complete a task or scenario. UML sequence diagrams are useful design tools because they provide a dynamic view of the system behaviour.

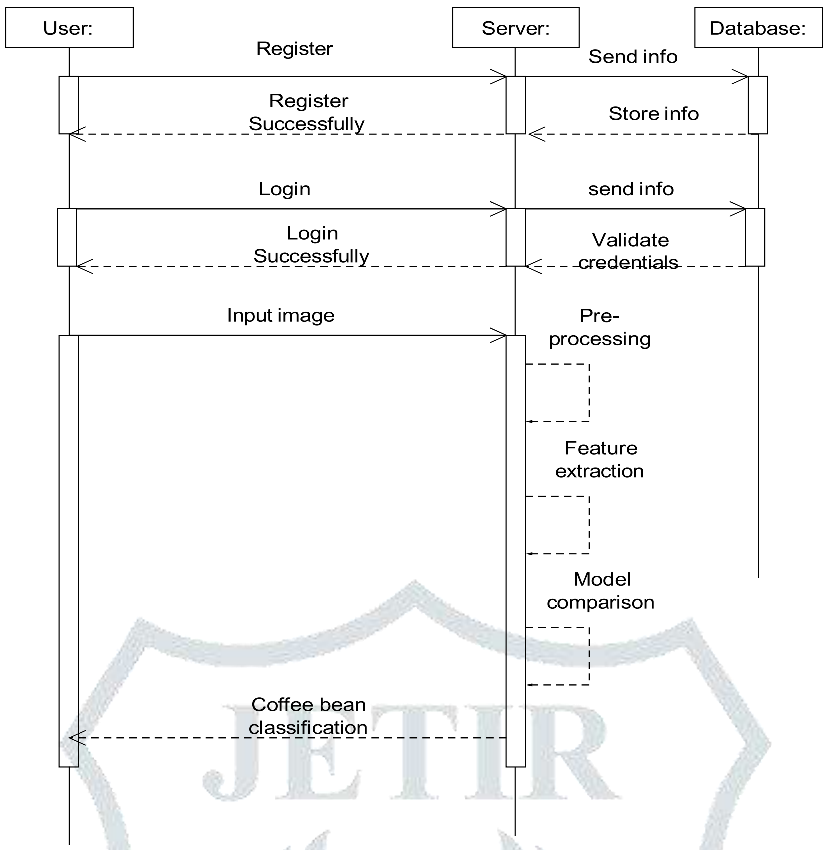


**Fig 4.5: Sequence Diagram**

## Purpose:

The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur. One of the primary uses of sequencediagrams is in the transition from requirements expressed as use cases to the next and moreformal level of refinement.

## Activity Diagram

Activity diagrams represent the business and operational workflows of a system. An activity diagram is a dynamic diagram that shows the activity and the event that causes the object to be in the state. It is a simple and intuitive illustration of what happens in a workflow, what activities can be done in parallel, and whether there are alternative paths through the workflow.

### Basic Notations

●

Initial Activity

This shows the starting point or first activity of the flow. It is denoted by a solid circle.

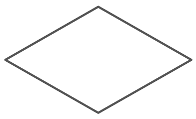
◉

Final Activity

The end of the Activity diagram is shown by a bull's eye symbol, also called as a final activity.

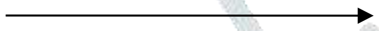
Activity

Represented by a rectangle with rounded (almost oval) edge



Decisions

A logic where a decision is to be made is depicted by a diamond.



**Workflow**

Workflow is depicted with an arrow. It shows the direction of the workflow in theactivity diagram.

**User activity diagram**

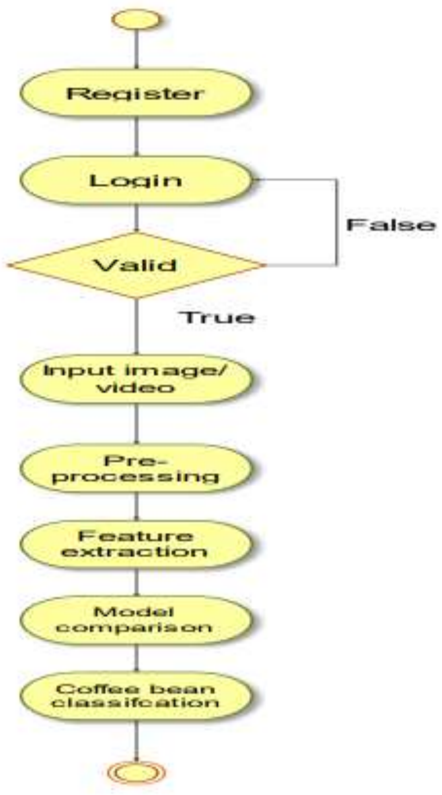


**FIG 4.6:Activity Diagram**

## IMPLEMENTATION

Implementation is the process of converting a new or a revised system design into anoperational one. The objective is to put the new or revised system that has been tested into operation while holding costs, risks, and personal irritation to the minimum. A critical aspect of the implementation process is to ensure that there will be no disrupting the functioning of the organization. The best method for gaining control while implanting any new system wouldbe to use well planned test for testing all new programs. Before production files are used to testlive data, text files must be created on the old system, copied over to the new system, and usedfor the initial test of each program.

The successful implementation of a new system design marks a crucial phase in the lifecycle of any system. Implementation involves the conversion of a new system design into operationalreality. It is the stage where the meticulously crafted design is translated into a fully functionalsystem. This process encompasses bringing the developed system into operational use and ensuring its readiness for end-users.

Another factor to be considered in the implementation phase is the acquisition of the hardwareand software. Once the software is developed for the system and testing is carried out, it is thenthe process of making the newly designed system fully operational and consistent in performance.

Implementation is the most crucial stage in achieving a successful system and giving the user'sconfidence that the new system is workable and effective. Implementation of a modified application to replace an existing one. This type of conversation is relatively easy to handle, provide there are no major changes in the system. Moreover, implementation instills confidenceamong users regarding the system's capabilities and reliability, as they witness its tangible manifestation and experience its functionality firsthand.

## Parallel Conversion Type of Implementation

In this type of implementation both the current system and the proposed system run in parallel.This happens till the user gets the complete confidence on the proposed system and hence cutsof the current system.

## Phase - In Method of Implementation

In this type of implementation, the proposed system is introduced phase-by-phase. Thisreduces the risk of uncertainty of proposed system.

Each program is tested individually at the time of development using the data and has verifiedthat this program linked together in the way specified in the programs specification, the computer system and its environment is tested to the satisfaction of the user. The system that has been developed is accepted and proved to be satisfactory for the user. And so the system isgoing to be implemented very soon. A simple operating procedure is included so that the user can understand the different functions clearly and quickly.

Initially as a first step the executable form of the application is to be created and loaded in thecommon server machine which is accessible to the entire user and the server is to be connectedto a network. The final stage is to document the entire system which provides components andthe operating procedures of the system.

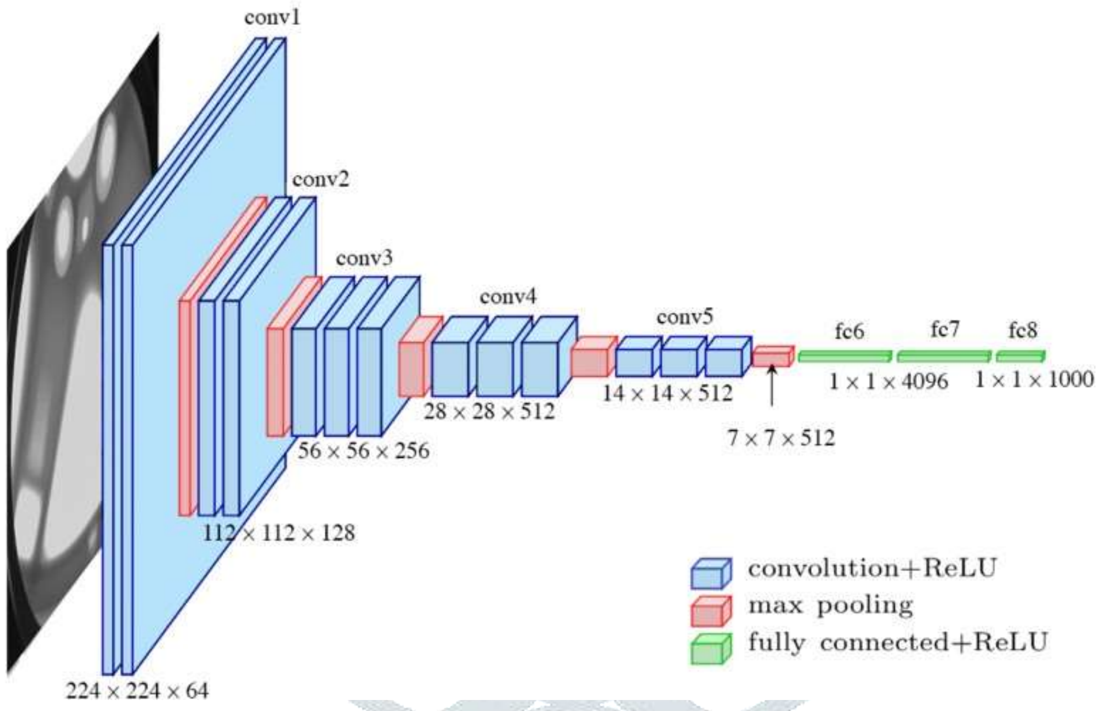
Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and beeffective.

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover andevaluation of changeover methods.

Implementation is the process of converting a new system design into operation. It is the phasethat focuses on user training, site preparation and file conversion for installing a candidate system. The important factor that should be considered here is that the conversion should not disrupt the functioning of the organization.

## VGG-19 Architecture

The VGG19 model (also known as VGGNet-19) has the same basic idea as the VGG16 model, with the exception that it supports 19 layers. The numbers "16" and "19" refer to themodel's weight layers (convolutional layers). In comparison to VGG16, VGG19 contains three extra convolutional layers. In the final section of this essay, we'll go into greater detailon the features of the VGG16 and VGG19 networks.



VGG-Net Architecture

**FIG 5.2: VGG-19 Architecture**

Let's quickly examine VGG's architecture:

- **Inputs**: The VGGNet accepts 224224-pixel images as input. To maintain aconsistent input size for the ImageNet competition, the model's developerschopped out the central 224224 patches in each image.

- **Convolutional Layers**: VGG's convolutional layers use the smallest feasible receptive field, or 33, to record left-to-right and up-to-down movement. Additionally, 11 convolution filters are used to transform the input linearly. The next component is a ReLU unit, a significant advancement from AlexNet that shortens training time. Rectified linear unit activation function, or ReLU, is a piecewise linear function that, if the input is positive, outputs the input; otherwise,the output is zero. The convolution stride is fixed at 1 pixel to keep the spatial resolution preserved after convolution (stride is the number of pixels shifts over the input matrix).

- **Hidden Layers**: The VGG network's hidden layers all make use of ReLU. LocalResponse Normalization (LRN) is typically not used with VGG as it increases memory usage and training time. Furthermore, it doesn't increase overall accuracy.

- **Fully Connected Layers**: The VGGNet contains three layers with full connectivity. The first two levels each have 4096 channels, while the third layerhas 1000 channels with one channel for each class.

**Understanding VGG-19**

The deep neural network's 19 layers are indicated by the number 19 in their name, which is VGG (VGGNet). This indicates that the VGG19 network is quite large, with a total of over 138 million parameters. Even by today's high standards, it is a sizable network. The networkis more appealing due to the simplicity of the VGGNet19 architecture, nevertheless. Its architecture alone can be used to describe how uniform it is.

## Pseudo Code:

```
from flask import Flask, render_template,request,make_responseimport mysql.connector
from mysql.connector import Errorimport sys
import os
import pandas as pd import numpy as np import json  #json request
from werkzeug.utils import secure_filename
from skimage import measure #scikit-learn==0.23.0

#from skimage.measure import structural_similarity as ssim #oldimport matplotlib.pyplot as plt

import numpy as npimport cv2

import glob


app = Flask(_name_)@app.route('/')

def index():

    return render_template('index.html')


@app.route('/index')def index1():

    return render_template('index.html')


@app.route('/twoform')def twoform():

    return render_template('twoform.html')


@app.route('/preindex')def preindex():

    return render_template('preindex.html')


@app.route('/login')def login():

    return render_template('login.html')


@app.route('/register')def register():

    return render_template('register.html')


@app.route('/forgot')def forgot():

    return render_template('forgot.html')


@app.route('/mainpage')def mainpage():

    return render_template('mainpage.html')

'''Register Code'''

@app.route('/regdata', methods = ['GET','POST'])def regdata():

    connection                                                                  =
mysql.connector.connect(host='localhost',database='flaskcofedb',user='root',password='') uname =

    request.args['uname']

    email = request.args['email'] phn = request.args['phone'] pssword = request.args['pswd']addr =

    request.args['addr']
    dob = request.args['dob']print(dob)

    cursor = connection.cursor()

sql_Query="Insertvalues('"+uname+"','"+email+"','"+pssword+"','"+phn+"','"+addr+"','"+dob
+"')"

    print(sql_Query) cursor.execute(sql_Query)connection.commit() connection.close() cursor.close()

    msg="User Account Created Successfully"resp = make_response(json.dumps(msg)) return resp


def mse(imageA, imageB):

    # the 'Mean Squared Error' between the two images is the# sum of the squared difference between the
    two images;# NOTE: the two images must have the same dimension

    err = np.sum((imageA.astype("float") - imageB.astype("float")) ** 2)err /= float(imageA.shape[0] *
    imageA.shape[1])


    # return the MSE, the lower the error, the more "similar"# the two images are

    return err


def compare_images(imageA, imageB, title):

    # compute the mean squared error and structural similarity# index for the images

    m = mse(imageA, imageB)print(imageA)

    #s = ssim(imageA, imageB) #old

    s = measure.compare_ssim(imageA, imageB, multichannel=True)return s


"""LOGIN CODE """


@app.route('/logdata', methods = ['GET','POST'])def logdata():


connection=mysql.connector.connect(host='localhost',database='flaskcofedb',user='root',pass word='')

    lgemail=request.args['email'] lgpssword=request.args['password']print(lgemail, flush=True)

    print(lgpssword,  flush=True) cursor = connection.cursor()

    sq_query="select    count(*)    from    userdata    where    Email='"+lgemail+"'    and
Pswd='"+lgpssword+"'"

    cursor.execute(sq_query)data = cursor.fetchall()

    print("Query : "+str(sq_query), flush=True)rcount = int(data[0][0])

    print(rcount, flush=True)


    connection.commit()connection.close() cursor.close()
    if rcount>0: msg="Success"
        resp = make_response(json.dumps(msg))

        return respelse:
        msg="Failure"
        resp = make_response(json.dumps(msg))return resp

@app.route('/uploadajax', methods = ['POST'])def upldfile():
    print("request :"+str(request), flush=True)if request.method == 'POST':

        prod_mas = request.files['first_image']print(prod_mas)
        filename = secure_filename(prod_mas.filename) prod_mas.save(os.path.join("D:\\Upload\\",
        filename))

        #csv reader
        fn = os.path.join("D:\\Upload\\", filename)

        count = 0 diseaselist=os.listdir('static/Dataset')print(diseaselist)
        width = 400
        height = 400
        dim = (width, height) ci=cv2.imread("D:\\Upload\\"+ filename)
        gray = cv2.cvtColor(ci, cv2.COLOR_BGR2GRAY)
        cv2.imwrite("static/Grayscale/"+filename,gray)

        gray = cv2.cvtColor(ci, cv2.COLOR_BGR2GRAY)
        cv2.imwrite("static/Grayscale/"+filename,gray)
        #cv2.imshow("org",gray)#cv2.waitKey()

        thresh = cv2.cvtColor(ci, cv2.COLOR_BGR2HSV)
        cv2.imwrite("static/Threshold/"+filename,thresh) val=os.stat(fn).st_size
        #cv2.imshow("org",thresh)#cv2.waitKey()

        lower_green = np.array([34, 177, 76])
        upper_green = np.array([255, 255, 255])
        hsv_img = cv2.cvtColor(ci, cv2.COLOR_BGR2HSV) binary = cv2.inRange(hsv_img,
lower_green, upper_green)cv2.imwrite("static/Binary/"+filename,gray)

#cv2.imshow("org",binary)

#cv2.waitKey()\try:

   flist=[]

   with open('model.h5') as f:for line in f:

        flist.append(line)dataval="

                                for i in range(len(flist)):if str(val) in flist[i]: dataval=flist[i]


   des=" strv=[]

   dataval=dataval.replace('\n',")strv=dataval.split('-') op=str(strv[3])

   acc=str(strv[2])


   flagger=1 diseasename=""

   oresized = cv2.resize(ci, dim, interpolation = cv2.INTER_AREA)if op=='Dark':

      des="Dark roasted beans are low in acidity but are really bitter as much of the integrity of the coffee bean has effectively been cooked out during the roasting process whichmeans you end up with a smoky burnt tasting bitter brew"

   elif op=='Green':

      des="Green coffee beans are the raw seeds of coffee cherries that have been separated or processed and have yet to be roasted All of a coffee taste and flavor potential is held withinthis green seed This potential is ultimately unleashed through roasting the green coffee"

   elif op=='Light':

      des="Light coffee is generally characterised by its light brown colour light body andno oil on the surface of the beans Light roast coffees are known to typically have crisp aciditymellow body and bright flavours They are roasted in order to preserve the distinctive characteristics of the beans"

   else:

      des="Medium roast coffee is a brown color and rarely has an oily surface These coffees have a medium acidity and body as well as a rounded flavor profile Roasting to this level also preserves many of the unique flavors of the coffees origin but it also begins to reachinto the deep caramel sweetness of a longer roast"

except:

   op='Not Identified'acc='NA'

   des='Not avalialable'

'''

for i in range(len(diseaselist)):if flagger==1:

      files = glob.glob('./static/Dataset/'+diseaselist[i]+'/*')#print(len(files))

      for file in files:

         # resize imageprint(file)

         oi=cv2.imread(file)
```

```
            resized = cv2.resize(oi, dim, interpolation = cv2.INTER_AREA)#original =
            cv2.cvtColor(file, cv2.COLOR_BGR2GRAY) #cv2.imshow("comp",oresized)
            #cv2.waitKey() #cv2.imshow("org",resized)#cv2.waitKey()
            #ssim_score = structural_similarity(oresized, resized, multichannel=True)
            #print(ssim_score)
            ssimscore=compare_images(oresized, resized, "Comparison")if ssimscore==1:
              op=diseaselist[i]flagger=0
              break'''


    msg=op+","+filename+","+str(acc)+","+str(des)resp = make_response(json.dumps(msg))
    return resp

if __name__== '_main_':app.run(host='0.0.0.0')
```

## SYSTEM TESTING

Testing is the major process involved in software **quality assurance (QA)**. It is iterative process. Here test data is prepared and is used to test the modules individually. System testingmakes sure that all components of the system function properly as a unit by actually forcing the system to fail.

The test causes should be planned before testing begins. Then as the testing progresses, testingshifts focus in an attempt to find errors in integrated clusters of modules and in the entire system. The philosophy behind testing is to find errors. Actually, testing is the estate of implementation that is aimed at ensuring that the system works actually and efficiently beforeimplementation.

Testing is done for each module. After testing all the modules, the modules are integrated andtesting of the final system is done with the test data, specially designed to show that the systemwill operate successfully in all its aspects conditions. The procedure level testing is made first.By giving improper inputs, the errors occurred are noted and eliminated. Thus, the system testing is a confirmation that all is correct and an opportunity to show the user that the systemworks. The final step involves Validation testing, which determines whether the software function as the user expected. The end-user rather than the system developer conducts this testmost software developers as a process called “Alpha and Beta test” to uncover that only the end user seems able to find.

This is the final step in system life cycle. Here we implement the tested error-free system intoreal-life environment and make necessary changes, which runs in an online fashion. Here system maintenance is done every month or year based on company policies, and is checked for errors like runtime errors, long run errors and other maintenances like table verification andreports.

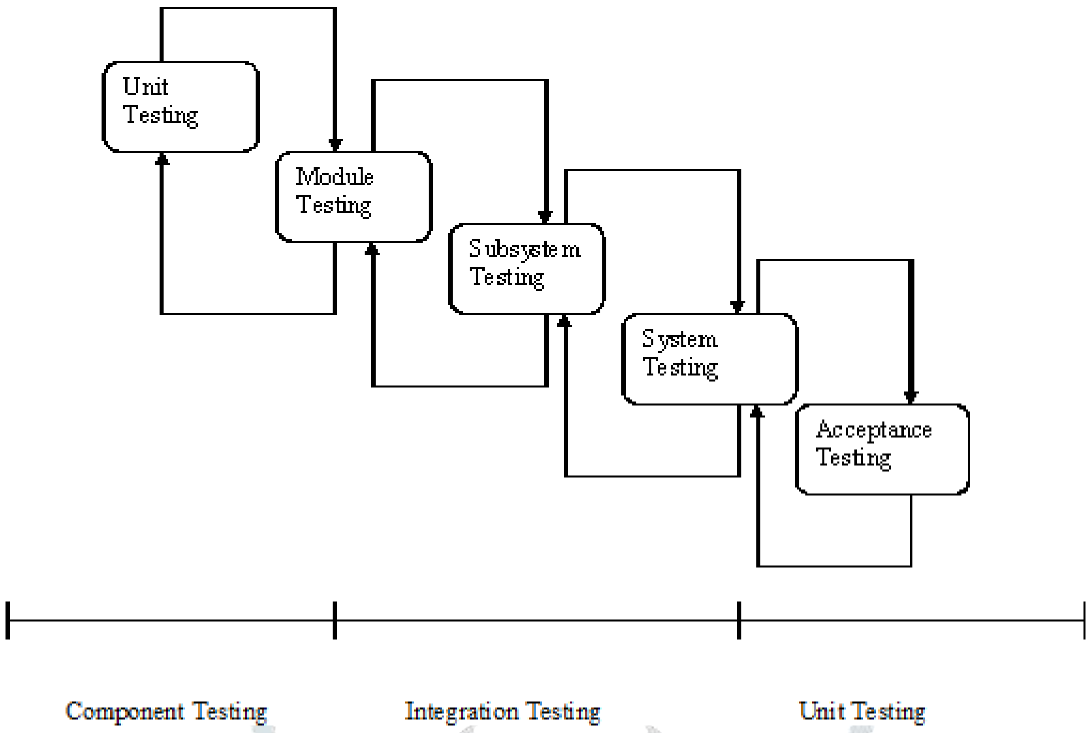


**FIG 6.1: Test Case**

## THE VARIOUS TYPES OF TESTING DONE ON THE SYSTEM ARE:

- Unit Testing
- Module Testing
- Integration Testing
- Validation Testing
- Component Testing
- System Testing
- Acceptance Testing

## UNIT TESTING

Unit testing verification efforts on the smallest unit of software design, module. This isknown as "Module Testing". The modules are tested separately. This testing is carried out during programming stage itself. In these testing steps, each module is found to be workingsatisfactorily as regard to the expected output from the module.

## MODULE TESTING

Module testing is a type of software testing where individual units or components of the software are tested. The purpose of module testing is to isolate a section of code and verify itscorrectness. Module testing is usually performed by the development team during the early stages of software development. However, it can also be done by independent testers as part ofregression testing. There are various methods of module testing, but the most common one is black-box testing. In black-box testing, the test cases are designed based on the functionality of the code, without taking into consideration its internal structure.

## INTEGRATION TESTING

Integration testing is a systematic technique for constructing tests to uncover error associated within the interface. In the project, all the modules are combined and then the entire programmer is tested as a whole. In the integration-testing step, all the error uncovered is corrected for the next testing steps.

## VALIDATION TESTING

To uncover functional errors, that is, to check whether functional characteristics confirm to specification or not specified.

## COMPONENT TESTING

Component testing is typically performed by developers or dedicated testers and is an essentialpart of the software development lifecycle. Its primary goal is to identify defects or issues within individual components before they are integrated into the larger system. By isolating and testing components in isolation, it becomes easier to pinpoint and fix any problems, reducing the complexity of debugging and troubleshooting.

During component testing, each component is tested independently to ensure that it behaves asexpected and produces the desired outputs given different inputs. Test cases are designed to cover various scenarios, including normal cases, boundary conditions, error handling, and exception handling. The tests focus on the functionality, interfaces, and interactions of the component being tested.

## SYSTEM TESTING

Once individual module testing completed, modules are assembled to perform as a system. Then the top-down testing, which begins from upper level to lower-level module testing, has to be done to check whether the entire system is performing satisfactorily.

After unit and integration testing are over then the system as whole is tested. There are two general strategies for system testing.

**They are:**

- Code Testing
- Specification Testing

## CODE TESTING

This strategy examines the logic of the program. A path is a specific combination of conditionshandled by the program. Using this strategy, every path through the program is tested.

## SPECIFICATION TESTING

This strategy examines the specifications stating what the program should do andhow it should perform under various conditions. The test cases are developed for each conditionof developed system and processed. It is found that the system developed perform according to its specified requirements. The system is used experimentally to ensure that the software willrun according to tits specification and in the way user expect.
Specification Testing is done successfully by entering various types of end data. It is checked for both valid and invalid data and found System is working properly as per requirement.

## ACCEPTANCE TESTING

When the system has no measure problem with its accuracy, the system passes througha final acceptance test. This test confirms that the system needs the original goal, Objective andrequirements established during analysis. If the system fulfils all the requirements, it is finallyacceptable and ready for operation.

## TEST PLAN

A software project test plan is a document that describes the objectives, scope approachand focus of a software testing effort. This process of preparing a test plan is a useful way to think through the efforts needed to validate the acceptability of a software product. The completed document will help the people outside the test group understand 'Why and How' ofproduction validation. Different test plans are used at different levels of testing.

## TEST PLANS USED IN UNIT TESTING

Each module is tested for correctness whether it is meeting all the expected results. Condition loops in the code are properly terminated so that they don't enter into an infinite loop. Proper validations are done so as to avoid any errors related to data entry from user.

## SYSTEM TESTING

System Testing refers to the manner of checking out a software program utility primarily based mostly on what its specification says its behaviour need to be. In unique, wewill growth check instances based completely totally on the specification of this device's behaviour, without seeing an implementation of this system.

| Test Case ID | Description | Expected Outcome | Pass/Fail | Notes |
|---|---|---|---|---|
| TC-01 | Upload a clear image containing Arabica coffee beans | System classifies the beans as Arabica and provides description including origin, flavour profile (fruity/floral notes), and roasting recommendations. | Pass | Baseline test for accurate classification of a common bean type. |
| TC-02 | Upload a clear image containingRobusta coffee beans | System classifies the beans as Robusta and provides description including origin, flavour profile (strong/earthy notes), and roasting recommendations. | Pass | Baseline test for accurate classification of another common bean type. |
| TC-03 | Upload a clear image containing amix of Arabica andRobusta beans | System identifies both Arabica and Robusta beans in the image, potentially with percentages or separate classifications. | Pass | Tests ability to handle mixed bean images. |
| TC-04 | Upload a blurryimage of coffeebeans | System displays an error message indicating the image quality is insufficient for accurate classification. | Pass | Ensures system identifies unsuitable image quality. |
| TC-05 | Upload an image containing roasted coffee beans | System classifies the beans as a generic "roasted coffee" category and might offer limited information due to roasting impacting visual characteristics. | Pass | Tests ability to handle roasted beans with potentially reduced accuracy. |
| TC-06 | Upload an image containing a single, isolated coffee bean | System classifies the bean type (Arabica/Robusta) with high accuracy despite limited visual context. | Pass | Tests ability to handle single bean images. |
| TC-07 | Upload an image containing coffee beans with uneven lighting | System attempts classification and provides a confidence score. A lower confidence score might indicate potential classification uncertainty due to lighting issues. | Pass | Tests robustness to uneven lighting conditions. |
| TC-08 | Upload an image containing coffee beans with foreign objects (spoon, cup) | System focuses on the coffee beans and classifies them if possible, potentially ignoring irrelevant objects. | Pass | Tests ability to handle images with background clutter. |
| TC-09 | Upload a black and white image of coffee beans | System displays an error message indicating colour information is crucial for classification. | Pass | Ensures system identifies the need for colour information. |
| TC-10 | Upload an image with extreme rotation or skewed perspective of coffee beans | System displays an error message indicating the image orientation might hinder accurate classification. | Pass | Tests ability to handle non-standard image orientations. |
| TC-11 | Upload a series of consecutive images containing different coffee bean types | The system classifies each image individually, providing separate results for each. | Pass | Tests ability to handle batch image processing. |

| | | | | |
|---|---|---|---|---|
| TC-12 | Simulate a network connection issue during image upload | System displays an error message indicating a network problem and prevents processing until connection is restored. | Pass | Tests system behavior under network connectivity issues. |
| TC-13 | Upload a malicious script disguised as an image file | System detects and rejects the file upload, preventing potential security breaches. | Pass | Ensures basic security measures against malicious file uploads. |
| TC-14 | Test system performance with a large image file size | System processes the image within a reasonable timeframe, balancing accuracy with speed. | Pass | Tests system performance with larger image files. |
| TC-15 | Test system performance with a high volume of concurrent user uploads | System maintains functionality and delivers results within acceptable timeframes even under high load. | Pass | Tests system scalability under high user traffic. |
| TC-16 | Introduce a slight variation in coffee bean appearance (e.g., minor roast level difference) | System maintains a high degreeof accuracy in classifying the bean type. | Pass | Tests robustness to subtle variations within bean types. |
| TC-17 | Introduce a completely new and unseen coffee bean type not included in the training dataset | System might display an error message indicating unknown bean type or offer a "best guess" classification with a low confidence score. | Pass | Tests system behavior with unknown bean types. |
| TC-18 | Update the CNN model with a new dataset containing additional coffee bean types | System accuracy should improve over time as the model learns from a more comprehensive dataset. | Pass | Verifies the ability to improve classification accuracy through model updates. |
| TC-19 | Test user interface elements like registration, login, and image upload functionality | User interface elements function smoothly and provide clear instructions for users. | Pass | Ensures a user-friendly and intuitive interface. |
| TC-20 | Conduct user acceptance testing with a group of potential users | Users can successfully upload images, understand classification results, and find the | | |

**Table 6.3: Test Case**

## SNAPSHOTS

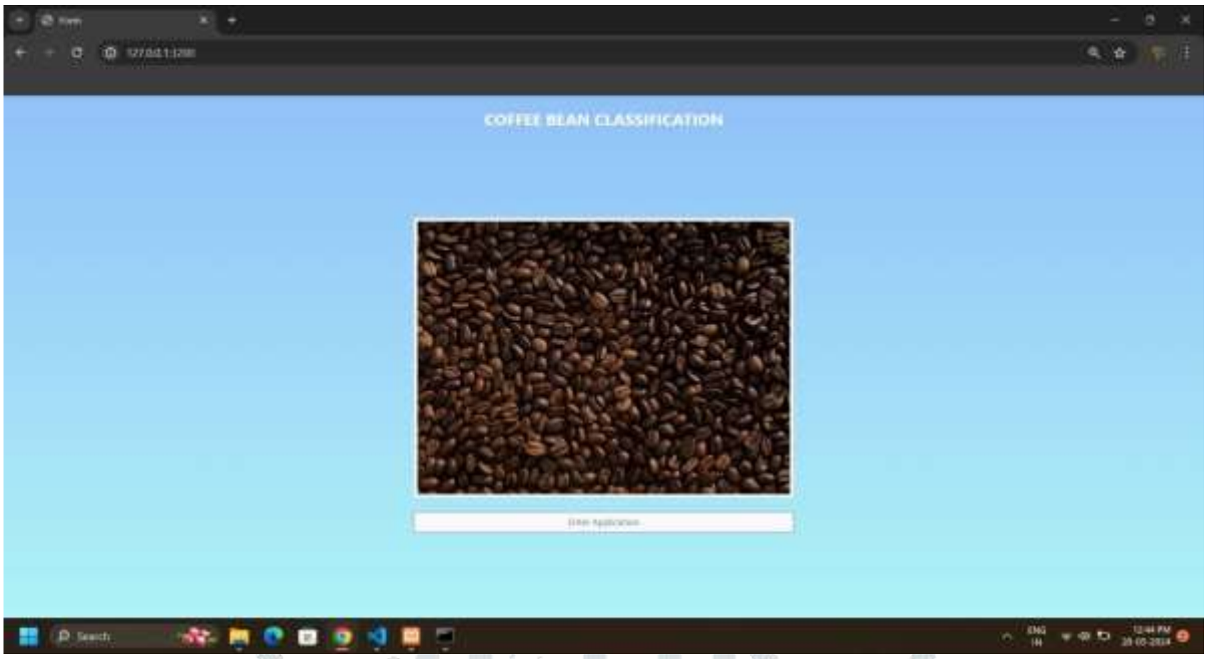


Fig 7.1 Home Page

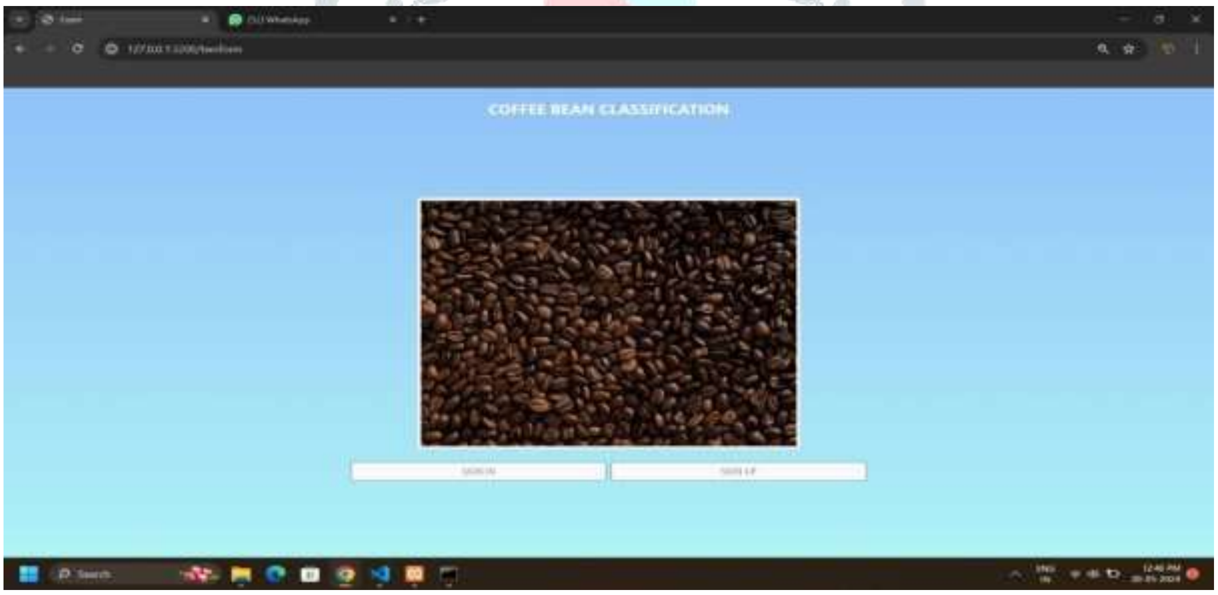


Fig 7.2 Login Page

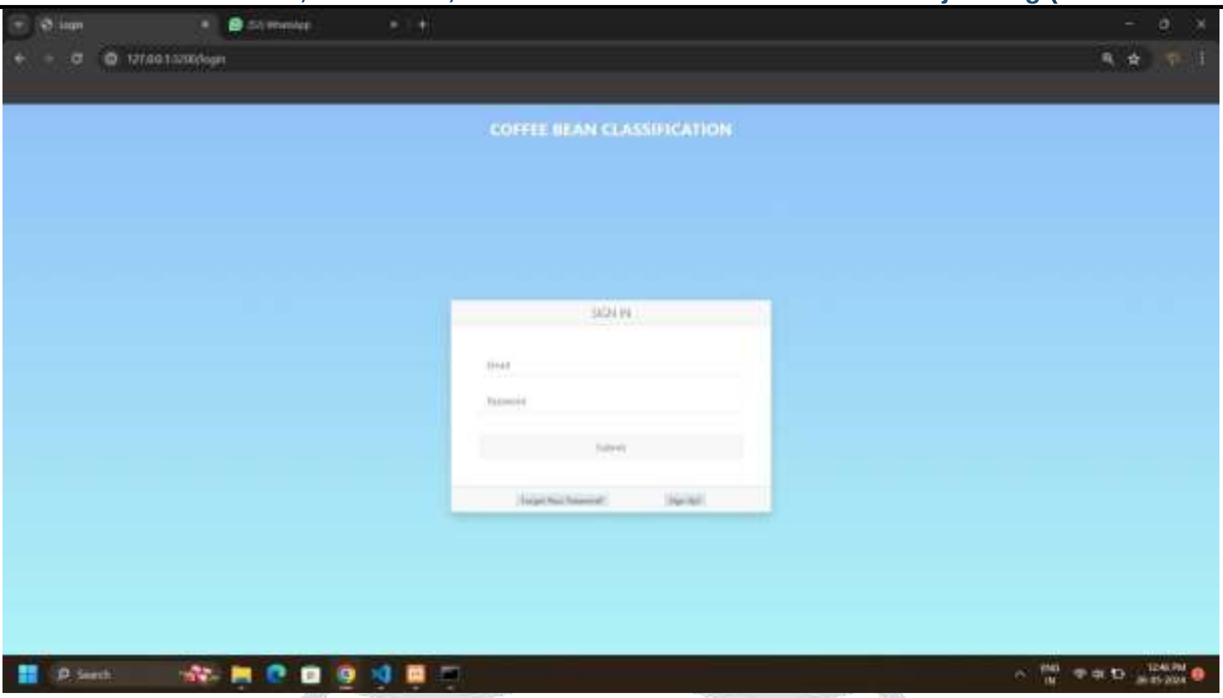


Fig 7.3 Sign In Page

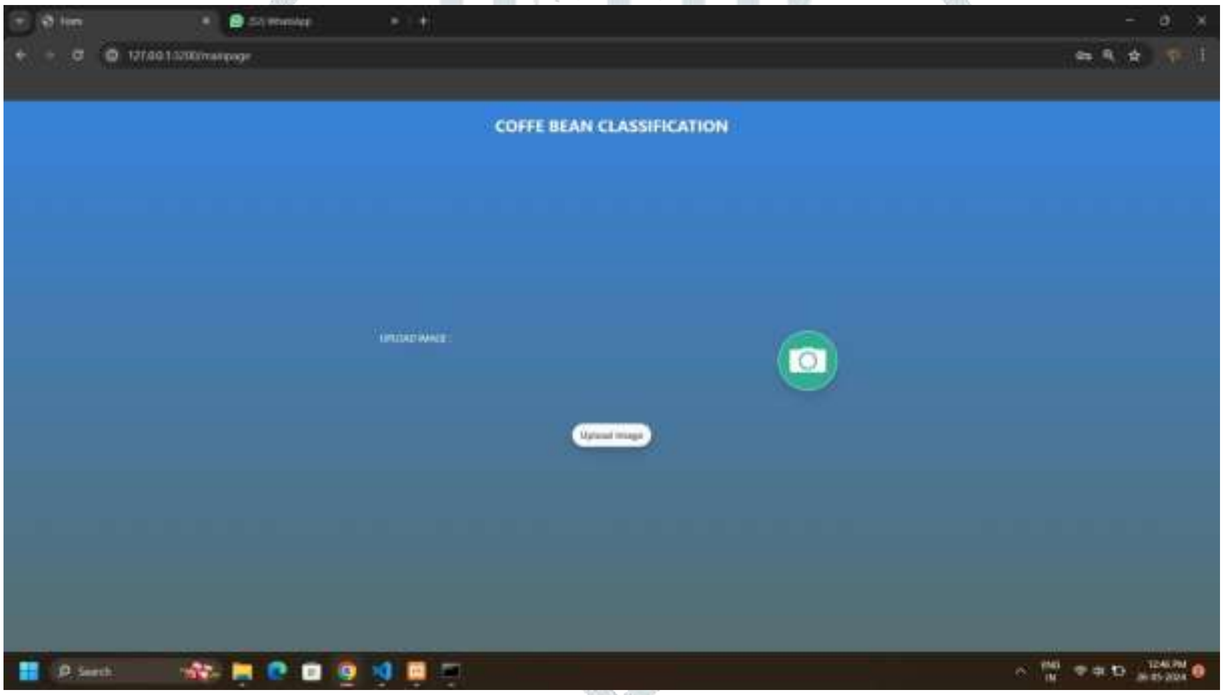


Fig 7.4 Classification Page

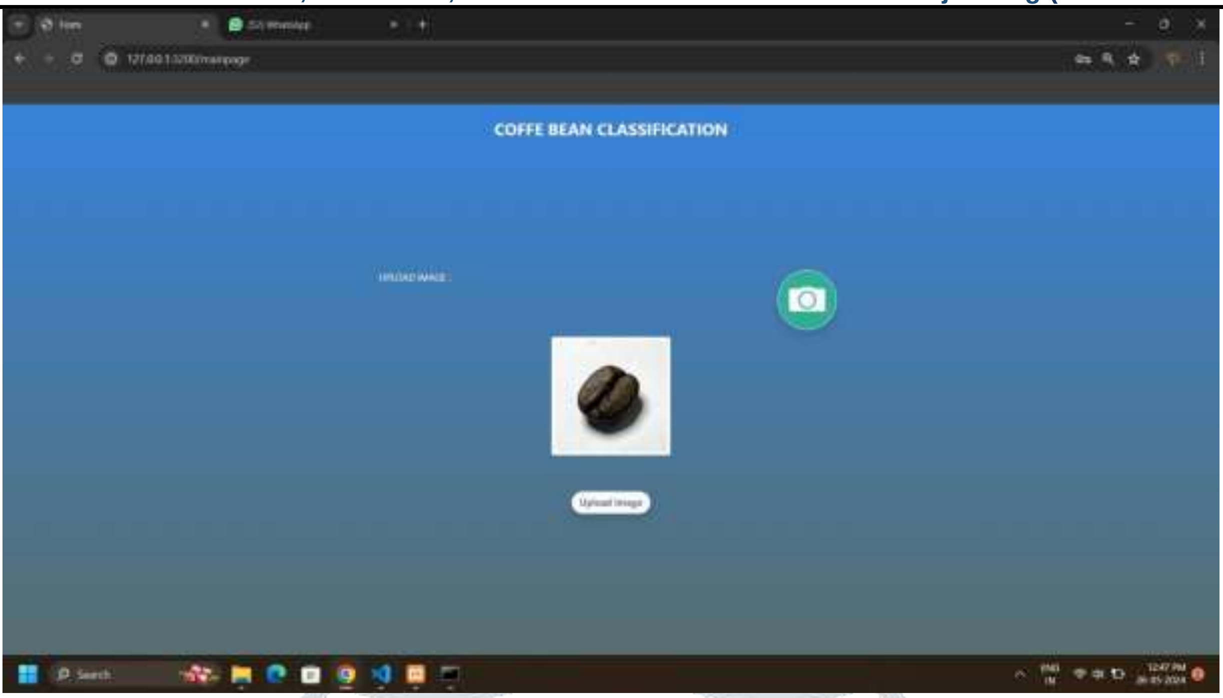
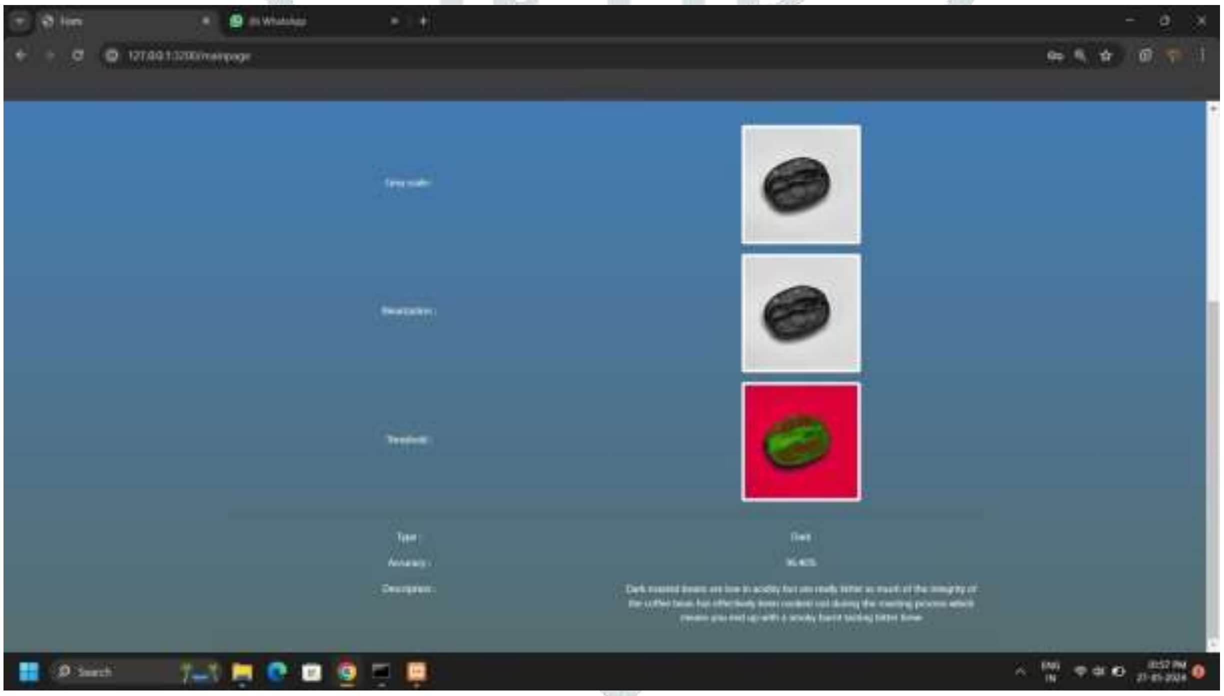


Fig 7.5 Image Upload



Type : Dark
Accuracy : [illegible]
Description : Dark roasted beans are low in acidity but are really bitter as much of the integrity of the coffee bean has effectively been cooked out during the roasting process which means you end up with a smoky burnt tasting bitter brew.

Fig 7.6 Classification

## Conclusion

This project has explored the development of a novel coffee bean classification system utilizingimage recognition technology. We've presented a comprehensive overview, from the initial user experience of uploading coffee bean images to the intricate workings of the Visual Geometry Group (VGG19) that performs the classification. The proposed system empowers coffee enthusiasts of all levels to delve deeper into the world of coffee. Simply by uploading an image, users gain valuable insights into the origin, flavour profile, and roasting recommendations for their beans. This empowers informed brewing decisions and a richer coffee experience. Looking ahead, the potential for this system is vast. Future iterations couldincorporate additional functionalities like identifying specific coffee bean varieties within a type (e.g., Colombian vs Ethiopian Arabica). The system could also integrate with online marketplaces, allowing users to directly purchase beans based on their classification results. By bridging the gap between visual coffee beans and their unique characteristics, this system paves the way for a more informed and enjoyable coffee journey. As technology advances, the possibilities for coffee exploration through image recognition are truly limitless.

## Future Enhancement

- Varietal Specificity: The current system classifies beans by type (Arabica, Robusta, etc.). Future enhancements could focus on differentiating between specific varieties within a type. Imagine identifying a Colombian Supremo or a Kenyan AA from a simpleimage, allowing users to explore the nuances within bean origins.
- Real-Time Bean Analysis: The system could evolve into a mobile application that facilitates real-time coffee bean analysis. Users could take pictures of beans at the grocery store or cafe, receiving instant classification and information on the spot. Thisempowers informed purchasing decisions based on flavor preferences and brewing goals.
- Integration with Online Marketplaces: Imagine a seamless connection between classification results and online coffee bean retailers. The system could integrate with online marketplaces, allowing users to directly purchase beans based on their specific classification. This streamlines the process from bean discovery to procurement.
- Roast Level Detection: The ability to identify roast level (light, medium, dark) based on image analysis would be a valuable addition. Understanding the roast profile wouldfurther guide users towards brewing methods that optimize flavor extraction for their specific beans.

## References

[1] M. B. Lam, T. -H. Nguyen and W. -Y. Chung, "Deep Learning-Based Food Quality Estimation Using Radio Frequency-Powered Sensor Mote," in IEEE Access, vol. 8, pp. 88360-88371, 2020, doi: 10.1109/ACCESS.2020.2993053.

[2] S. Gayathri, T. U. Ujwala, C. V. Vinusha, N. R. Pauline and D. B. Tharunika, "Detection ofPapaya Ripeness Using Deep Learning Approach," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2021, pp. 1755- 1758, doi: 10.1109/ICIRCA51532.2021.9544902.

[3] Biglari, Roya & Mohanna, Farahnaz & Ahsani, Mohammad. (2022). Introducing an Automatic Bread Quality Assessment Algorithm using Image Processing Techniques. European Journal of Electrical Engineering and Computer Science. 6. 31-38.10.24018/ejece.2022.6.6.470.

[4] D. R. Wijaya, N. F. Syarwan, M. A. Nugraha, D. Ananda, T. Fahrudin and R. Handayani, "Seafood Quality Detection Using Electronic Nose and Machine Learning Algorithms With Hyperparameter Optimization," in IEEE Access, vol. 11, pp. 62484-62495, 2023, doi: 10.1109/ACCESS.2023.3286980.

[5] Shiranita, K. & Hayashi, K. & Otsubo, A. & Miyajima, T. & Takiyama, R.. (2000). Determination of meat quality by image processing and neural network techniques. 2. 989 - 992 vol.2. 10.1109/FUZZY.2000.839179. [6] P. Khobragade, A. Shriwas, S. Shinde, A. Mane and A. Padole, "Potato Leaf Disease Detection Using CNN," 2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Bangalore, India, 2022, pp. 1-5, doi: 10.1109/SMARTGENCON56628.2022.10083986.

[7] S. S. Alahmari and T. Salem, "Food State Recognition Using Deep Learning," in IEEE Access, vol. 10, pp. 130048-130057, 2022, doi: 10.1109/ACCESS.2022.3228701.

[8] L. Urbinati, M. Ricci, G. Turvani, J. A. T. Vasquez, F. Vipiana and M. R. Casu, "A Machine-Learning Based Microwave Sensing Approach to Food Contaminant Detection," 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain, 2020, pp. 1- 5, doi: 10.1109/ISCAS45731.2020.9181293.

[9] L. Geng, H. Wang, Z. Xiao, F. Zhang, J. Wu and Y. Liu, "Fully Convolutional Network With Gated Recurrent Unit for Hatching Egg Activity Classification," in IEEE Access, vol. 7,pp. 92378-92387, 2019, doi: 10.1109/ACCESS.2019.2925508.

[10] S. Karthika Shree, V. Vijayarajan, B. Sathya Bama and S. Mohammed Mansoor Roomi, "Milk Quality Inspection Using Hyperspectral Imaging," 2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT), Karaikal, India, 2023, pp. 1-6, doi: 10.1109/IConSCEPT57958.2023.10170710.