D. Y. Patil Agriculture and Technical University, Talsande .

PROJECT NAME:

Uncovering the Hidden Treasures of the Mushroom Kingdom: A Classification Analysis

STUDENT INFORMATION:

STUDENT NAME: Juien Mirza Mujawar

CLASS: B.Tech AIML

DIV: A

ROLL NO: 17

PRN NO: 2023011042007

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Project Overview:

This project aims to classify different species of mushrooms based on their images. The classification is done using deep learning methods, specifically transfer learning using popular models Inception. The focus is on identifying the cap, gills underside of cap, and astern, which are key features for the optical recognition of mushroom species. The identified species are Boletus, Lactarius, and Russula.

The classification task involves three major categories of mushrooms: Boletus, Lactarius, and Russula. These categories encompass a wide range of species found across different regions of our planet. Boletus mushrooms are known for their distinctive cap shapes and pore-covered undersides, while Lactarius mushrooms often exhibit vibrant colors and produce a milky latex when damaged. Russula mushrooms, on the other hand, showcase diverse cap and stem characteristics and are an intriguing group to explorewhich are found in various habitats like forests, fields, and decomposing logs. Mushrooms have different shapes, sizes, and colors and are used for food, medicine, and other purposes. By leveraging deep learning techniques and transfer learning, this project aims to improve the accuracy and efficiency of mushroom species classification.

1.2 Purpose :

The purpose of this project is to develop a robust and accurate system for optical recognition and classification of mushroom species based on their visual characteristics. By leveraging deep learning techniques and transfer learning, this project aims to enhance the efficiency and accuracy of mushroom species identification. The project not only contributes to the field of mycology but also holds ecological significance by aiding in the study and conservation of mushroom species. Additionally, the system has practical applications in culinary and medicinal domains, enabling the identification of edible and medicinal mushrooms. Overall, this project serves as a comprehensive exploration of deep learning and transfer learning methods in the context of mushroom species recognition, showcasing their potential in image analysis and classification tasks.

1.3 Objectives of the Project

The key objectives of this project are:

- To understand image preprocessing and data augmentation techniques.
- To apply transfer learning using the Xception architecture.
- To build a classification model for mushroom species.
- To evaluate model performance using accuracy metrics.
- To integrate the trained model into a Flask web application.
- To provide a user-friendly interface for real-time image prediction.

1.4 Tools and Technologies Used

Google Colab:
Google Colab provides a cloud-based Jupyter notebook environment with free GPU access, making it ideal for deep learning model training.

Python Libraries:
NumPy – Used for numerical computations and array manipulation.
Pandas – Used for structured data handling (if required).
Matplotlib – Used for visualization of training results.
TensorFlow and Keras – Used for implementing deep learning models.
Flask – Used for deploying the trained model as a web application

1.5 Dataset Collection and Organization

The dataset consists of mushroom images categorized into three classes:

1. Boletus
2. Lactarius
3. Russula

The dataset is organized into training and testing folders as follows:

```
Dataset/
├── train/
│   ├── Boletus/
│   ├── Lactarius/
│   └── Russula/
└── test/
    ├── Boletus/
    ├── Lactarius/
    └── Russula/
```

This folder structure allows the ImageDataGenerator class to automatically assign labels based on directory names.

CHAPTER 2

REQUIREMENT ANALYSIS

2.1 Functional Requirement :

FR No.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	User Registration	Allow users to register an account.
FR-2	User Confirmation	Send confirmation email to verify user’s registration.
FR-3	Mushroom Classification	• Implement deep learning algorithms for mushroom analysis. • Use transfer learning techniques(inception V3, resnet50V2, Xception) • Train the models to classify mushrooms into Boletus, Lactarius, and Russula categories
FR-4	Image Processing	Develop image processing module to extract features from mushroom images.
FR-5	Prediction and Classification	Perform prediction and classification of mushrooms based on the trained models
FR-6	User Interface	Design a user friendly interface for users to interact with the system

2.2 Non-Functional Requirement :

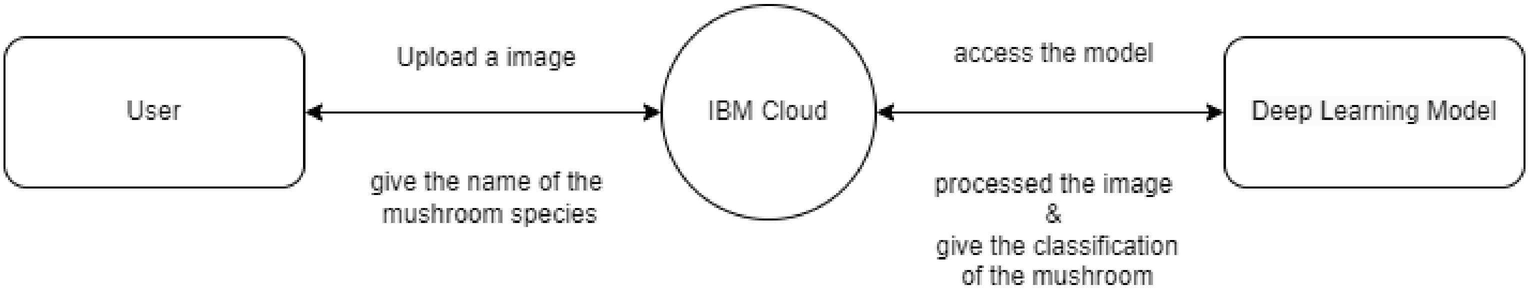
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The system should be intuitive and easy to use for both technical and non-technical users.
NFR-2	**Security**	Implement security measures to protect user data and prevent unauthorized access.
NFR-3	**Reliability**	Ensure the system is stable and reliable, with minimal downtime or errors.
NFR-4	**Performance**	The system should have fast processing and response times for image classification.
NFR-5	**Availability**	The system should be available for users to access and use at all times.
NFR-6	**Scalability**	The system should be able to handle a growing number of users and mushroom images efficiently.

CHAPTER 3

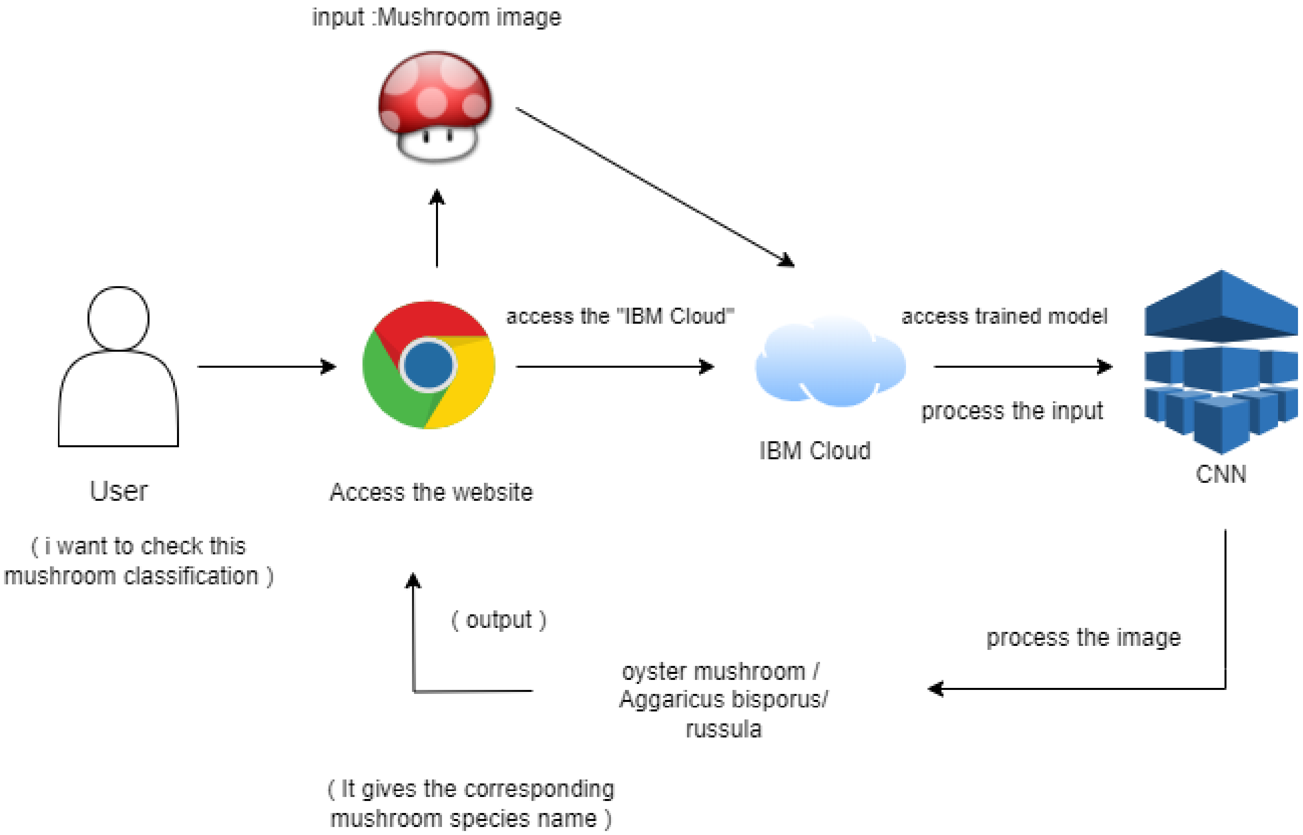
PROJECT DESIGN

3.1 Data Flow Diagrams :

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.


User
Upload a image
give the name of the mushroom species
IBM Cloud
access the model
processed the image & give the classification of the mushroom
Deep Learning Model

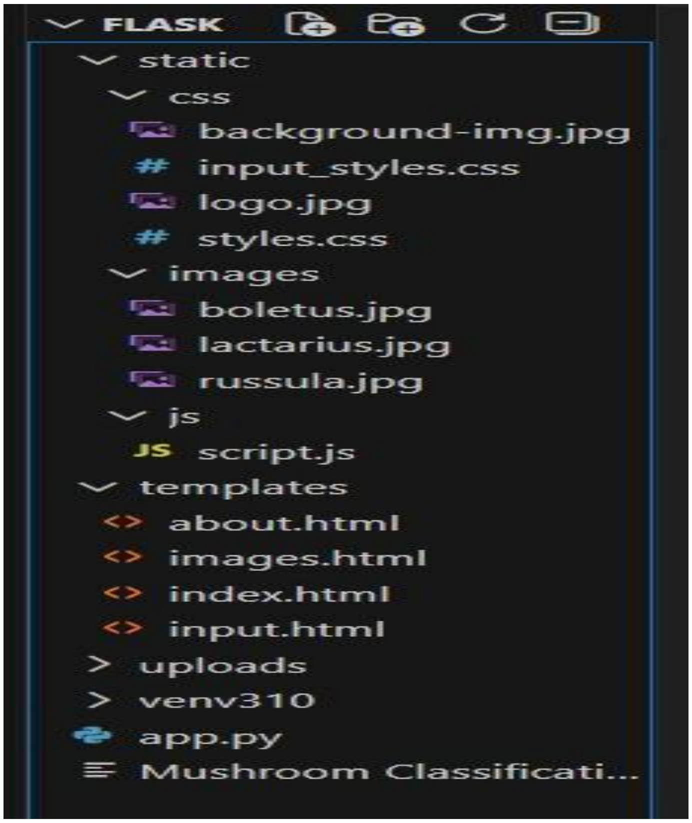
3.2 Solution & Technical Architecture :


input :Mushroom image
User
(i want to check this mushroom classification)
Access the website
access the "IBM Cloud"
IBM Cloud
access trained model
process the input
CNN
process the image
oyster mushroom / Aggaricus bisporus/ russula
(output)
(It gives the corresponding mushroom species name)

CHAPTER 4

CODING & SOLUTIONING

PROJECT STRUCTURE :



```
FLASK
  static
    css
      background-img.jpg
      input_styles.css
      logo.jpg
      styles.css
    images
      boletus.jpg
      lactarius.jpg
      russula.jpg
    js
      script.js
  templates
    about.html
    images.html
    index.html
    input.html
  uploads
  venv310
  app.py
  Mushroom Classificati...
```

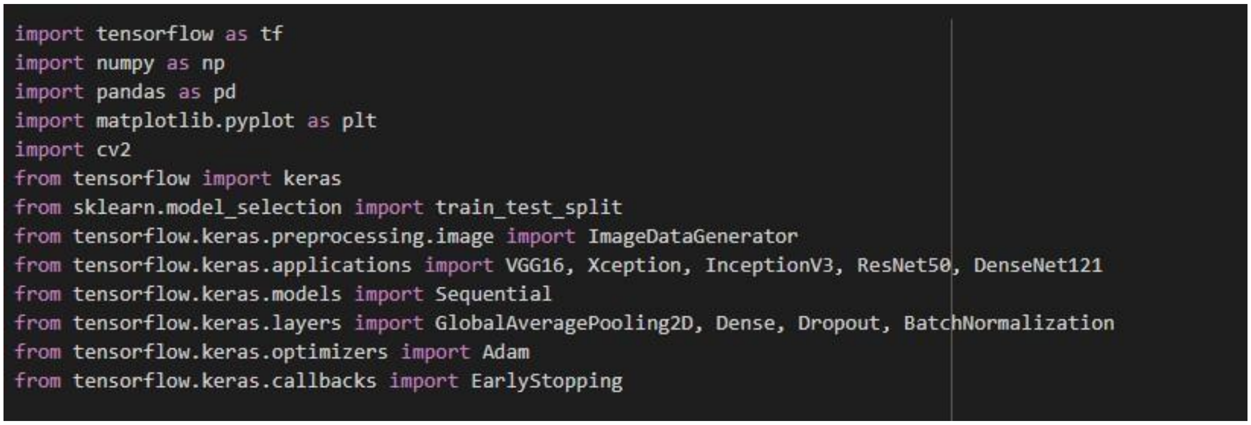
Uncovering the Hidden Treasures of the Mushroom Kingdom: A Classification Analysis Project structure, it have the static(css,js), templates(HTML),uploads,app.py file and transfer learning model.

4.1 Feature 1 :

Uncovering the Hidden Treasures of the Mushroom Kingdom: A Classification Analysis Project has function that user uploads any mushroom image it will give the classification of the mushroom, with their family name like “Boletus, Lactarius & Russula.”

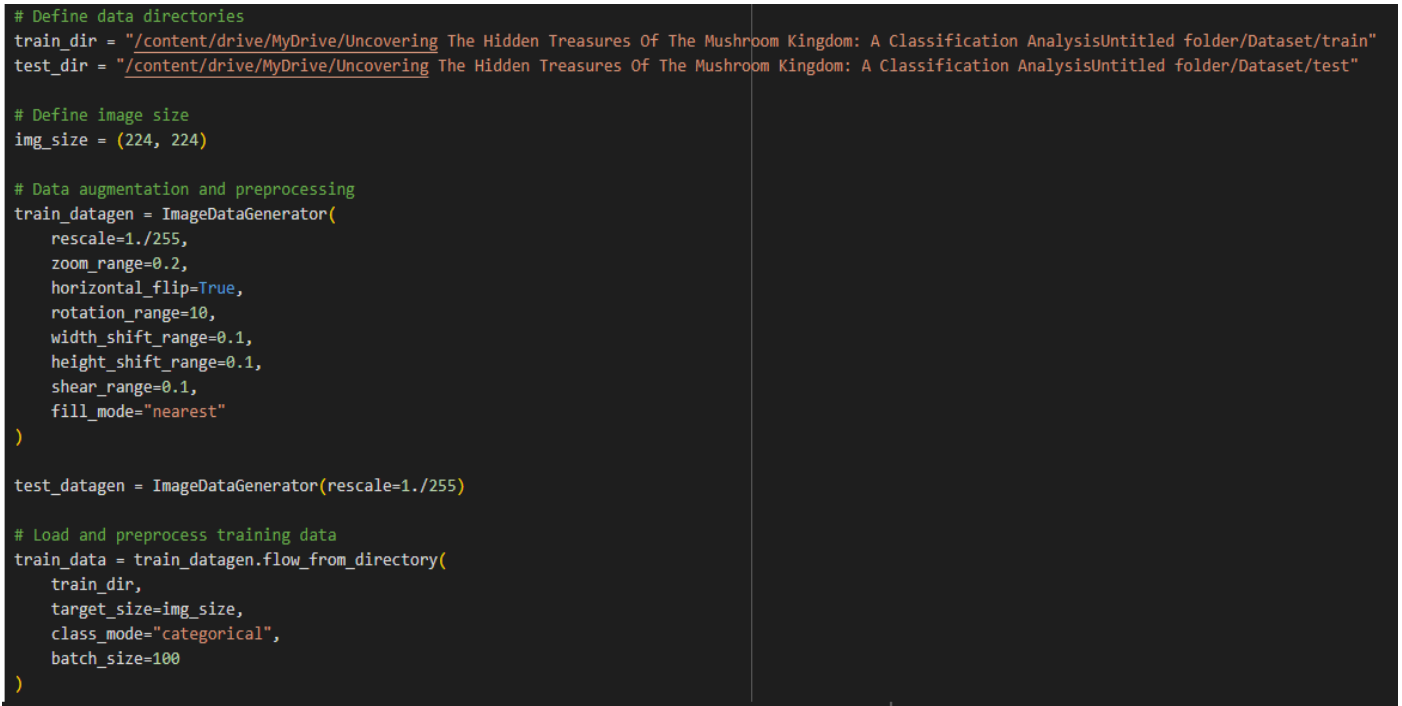
Coding :

Importing required libraries

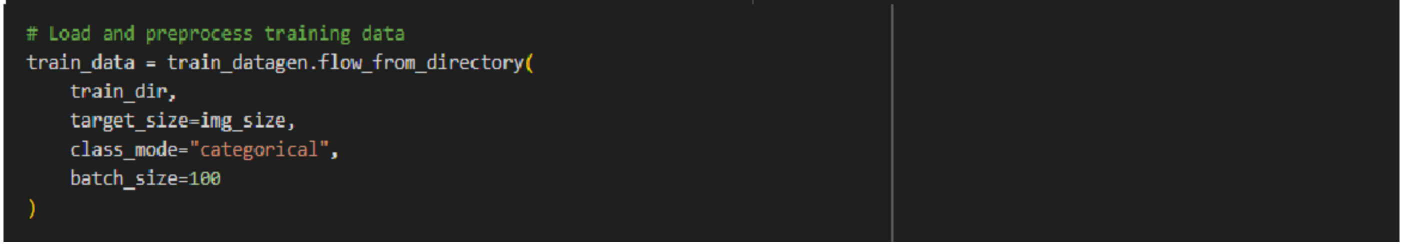


```python
import tensorflow as tf
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import cv2
from tensorflow import keras
from sklearn.model_selection import train_test_split
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import VGG16, Xception, InceptionV3, ResNet50, DenseNet121
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GlobalAveragePooling2D, Dense, Dropout, BatchNormalization
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
```

Define dataset directory and data augmentation

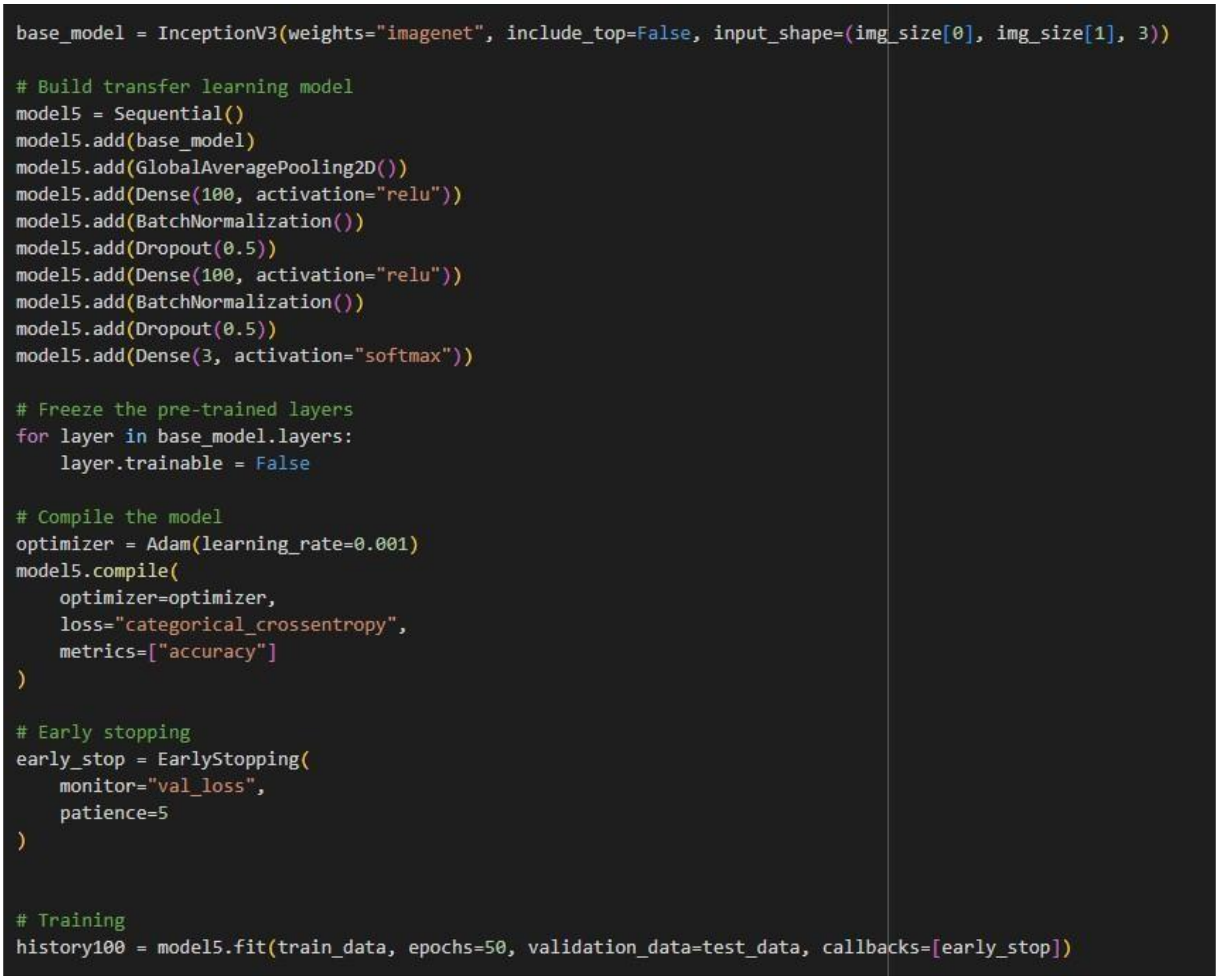


```python
# Define data directories
train_dir = "/content/drive/MyDrive/Uncovering The Hidden Treasures Of The Mushroom Kingdom: A Classification AnalysisUntitled folder/Dataset/train"
test_dir = "/content/drive/MyDrive/Uncovering The Hidden Treasures Of The Mushroom Kingdom: A Classification AnalysisUntitled folder/Dataset/test"

# Define image size
img_size = (224, 224)

# Data augmentation and preprocessing
train_datagen = ImageDataGenerator(
    rescale=1./255,
    zoom_range=0.2,
    horizontal_flip=True,
    rotation_range=10,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    fill_mode="nearest"
)

test_datagen = ImageDataGenerator(rescale=1./255)

# Load and preprocess training data
train_data = train_datagen.flow_from_directory(
    train_dir,
    target_size=img_size,
    class_mode="categorical",
    batch_size=100
)
```



```python
# Load and preprocess training data
train_data = train_datagen.flow_from_directory(
    train_dir,
    target_size=img_size,
    class_mode="categorical",
    batch_size=100
)
```

Model building with InceptionV3 pretrained model, adding layer & Training the model :



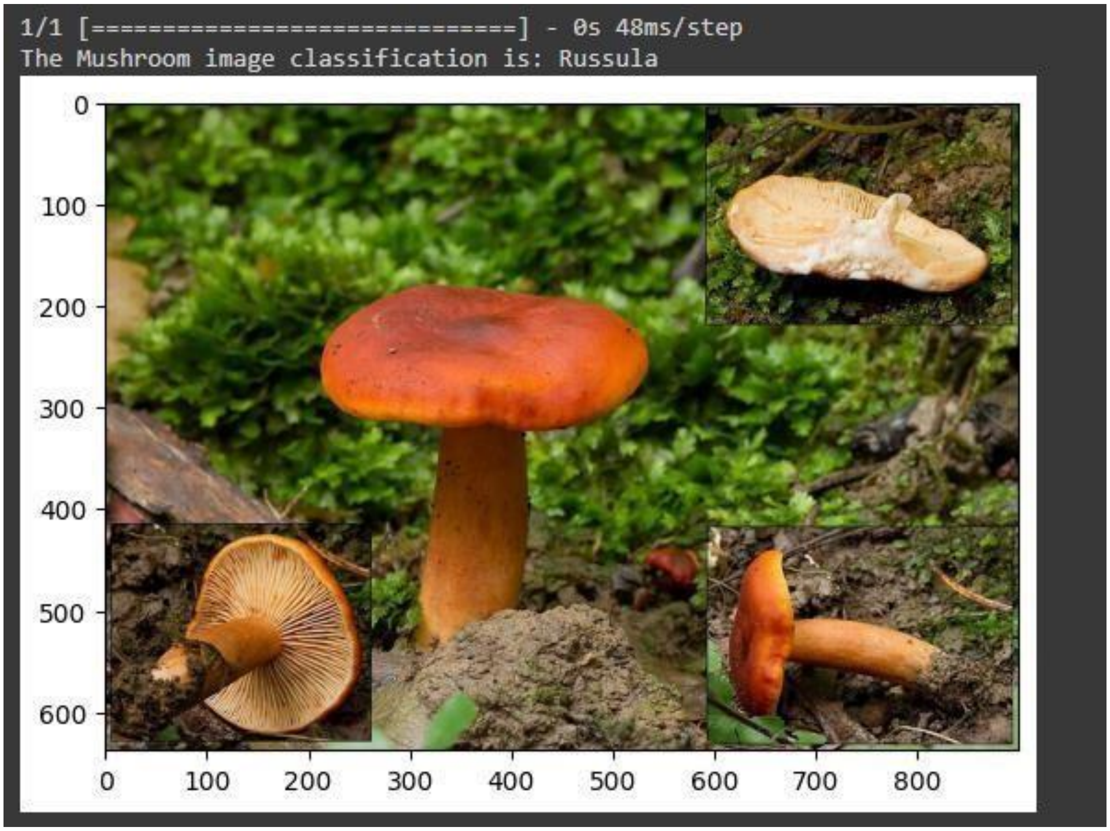
```
base_model = InceptionV3(weights="imagenet", include_top=False, input_shape=(img_size[0], img_size[1], 3))

# Build transfer learning model
model5 = Sequential()
model5.add(base_model)
model5.add(GlobalAveragePooling2D())
model5.add(Dense(100, activation="relu"))
model5.add(BatchNormalization())
model5.add(Dropout(0.5))
model5.add(Dense(100, activation="relu"))
model5.add(BatchNormalization())
model5.add(Dropout(0.5))
model5.add(Dense(3, activation="softmax"))

# Freeze the pre-trained layers
for layer in base_model.layers:
    layer.trainable = False

# Compile the model
optimizer = Adam(learning_rate=0.001)
model5.compile(
    optimizer=optimizer,
    loss="categorical_crossentropy",
    metrics=["accuracy"]
)

# Early stopping
early_stop = EarlyStopping(
    monitor="val_loss",
    patience=5
)


# Training
history100 = model5.fit(train_data, epochs=50, validation_data=test_data, callbacks=[early_stop])
```

This model is build with pretrained model “InceptionV3” and adding the layers like 3 dense layers , GlobalAveragePooling2D. A Dense layer with 100 units and ReLU activation function is added. BatchNormalization layer is added to normalize the activations of the previous layer.

Dropout layer is added to prevent overfitting by randomly dropping 50% of the connections.

A Dense layer with 3 units and softmax activation function is added as the final output layer.

Predicting the input image :



```
1/1 [==============================] - 0s 48ms/step
The Mushroom image classification is: Russula
```

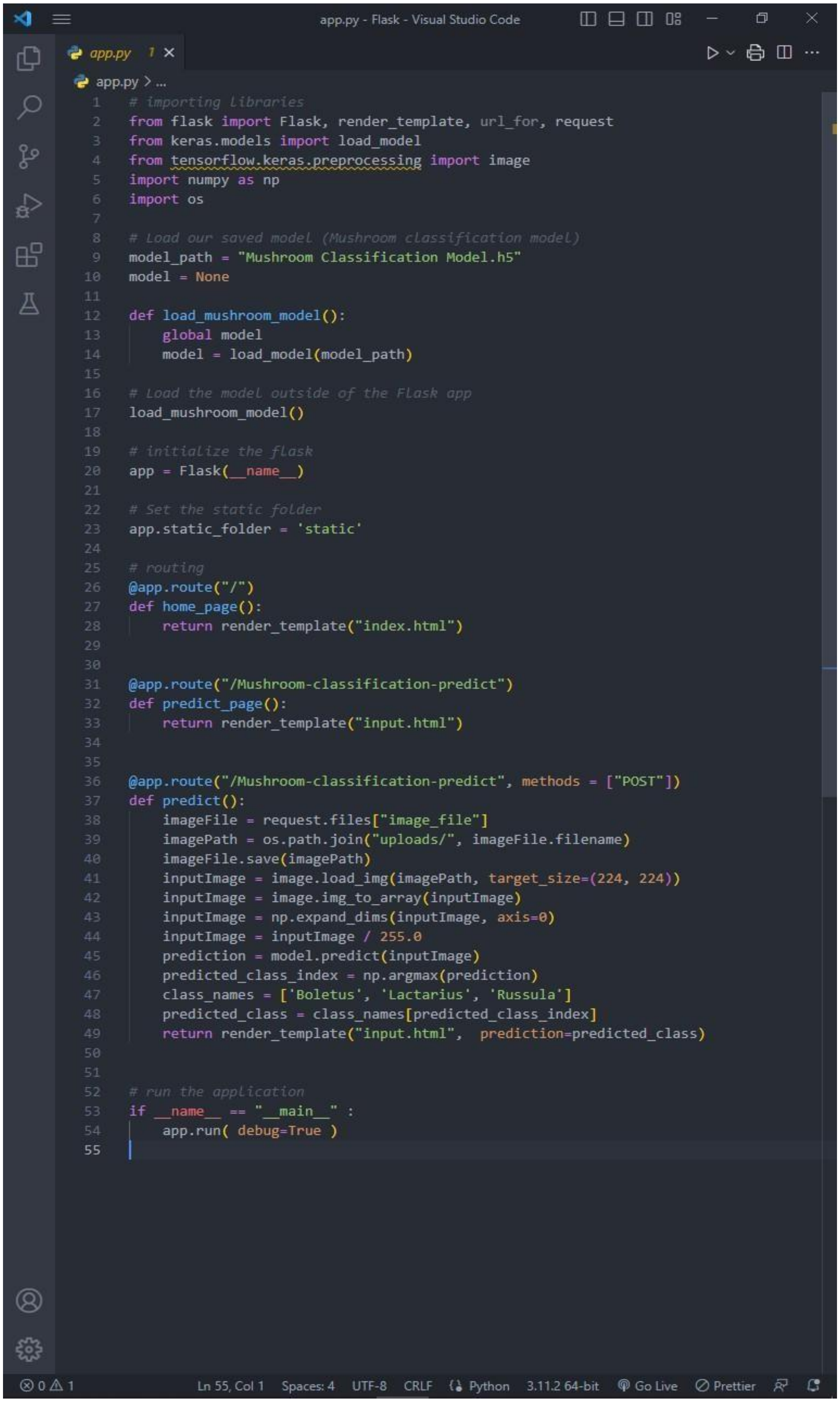
If user upload image, the model will classify the images then process the image and give the mushroom classification (Output).

4.1 Feature 2 :

In Existing projects it doesn't show the upload file, but our project overcome it .We integrated Machine learning to flask. Flask is a webFramework development.

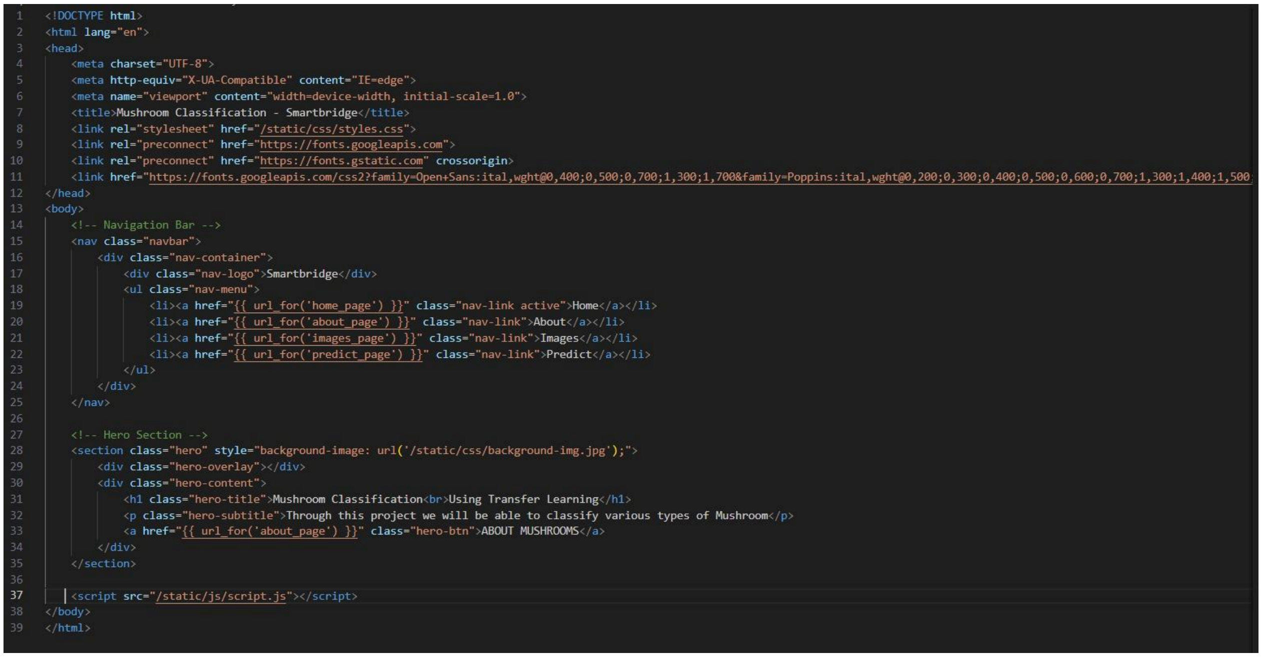
Uncovering the Hidden Treasures of the Mushroom Kingdom: A Classification Analysis Project structure, it have the static(css,js), templates(HTML),uploads,app.py file and transfer learning model.

Flask App :



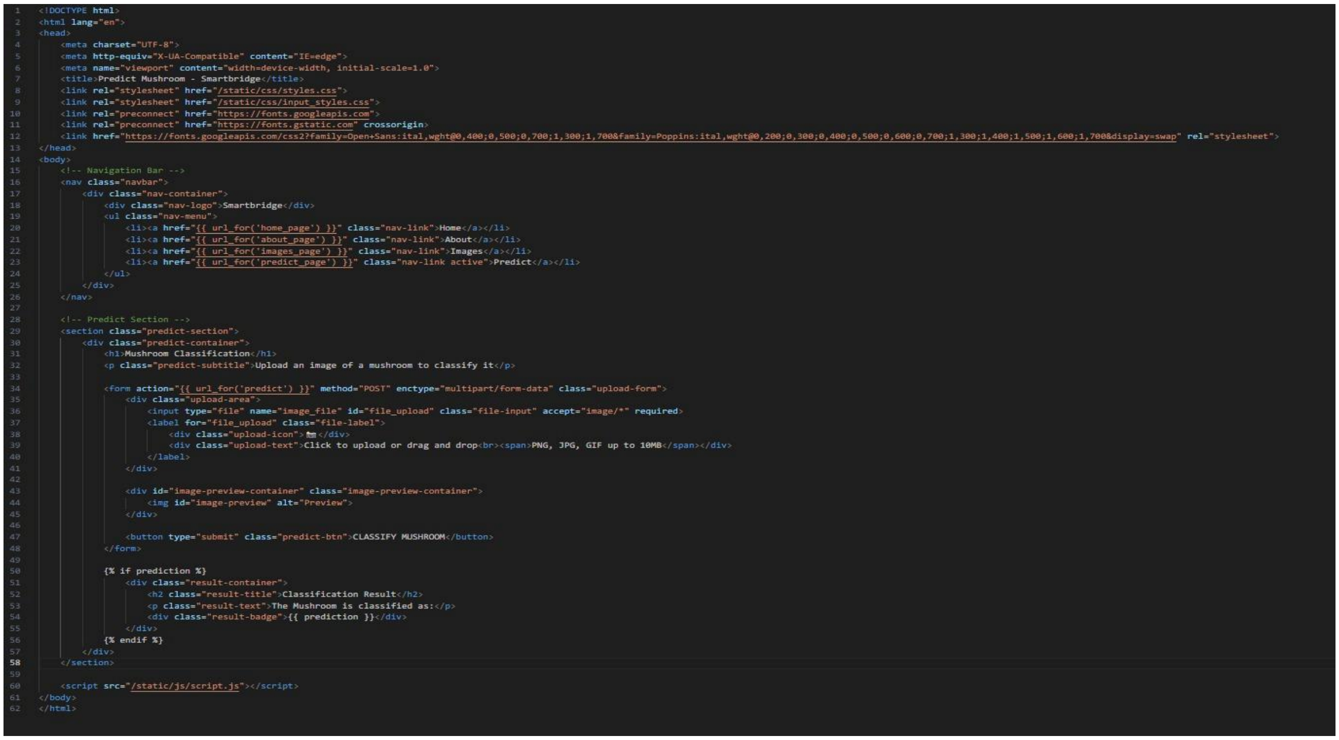
```python
# importing libraries
from flask import Flask, render_template, url_for, request
from keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
import os

# Load our saved model (Mushroom classification model)
model_path = "Mushroom Classification Model.h5"
model = None

def load_mushroom_model():
    global model
    model = load_model(model_path)

# Load the model outside of the Flask app
load_mushroom_model()

# initialize the flask
app = Flask(__name__)

# Set the static folder
app.static_folder = 'static'

# routing
@app.route("/")
def home_page():
    return render_template("index.html")


@app.route("/Mushroom-classification-predict")
def predict_page():
    return render_template("input.html")


@app.route("/Mushroom-classification-predict", methods = ["POST"])
def predict():
    imageFile = request.files["image_file"]
    imagePath = os.path.join("uploads/", imageFile.filename)
    imageFile.save(imagePath)
    inputImage = image.load_img(imagePath, target_size=(224, 224))
    inputImage = image.img_to_array(inputImage)
    inputImage = np.expand_dims(inputImage, axis=0)
    inputImage = inputImage / 255.0
    prediction = model.predict(inputImage)
    predicted_class_index = np.argmax(prediction)
    class_names = ['Boletus', 'Lactarius', 'Russula']
    predicted_class = class_names[predicted_class_index]
    return render_template("input.html",  prediction=predicted_class)


# run the application
if __name__ == "__main__" :
    app.run( debug=True )

```

index.html :



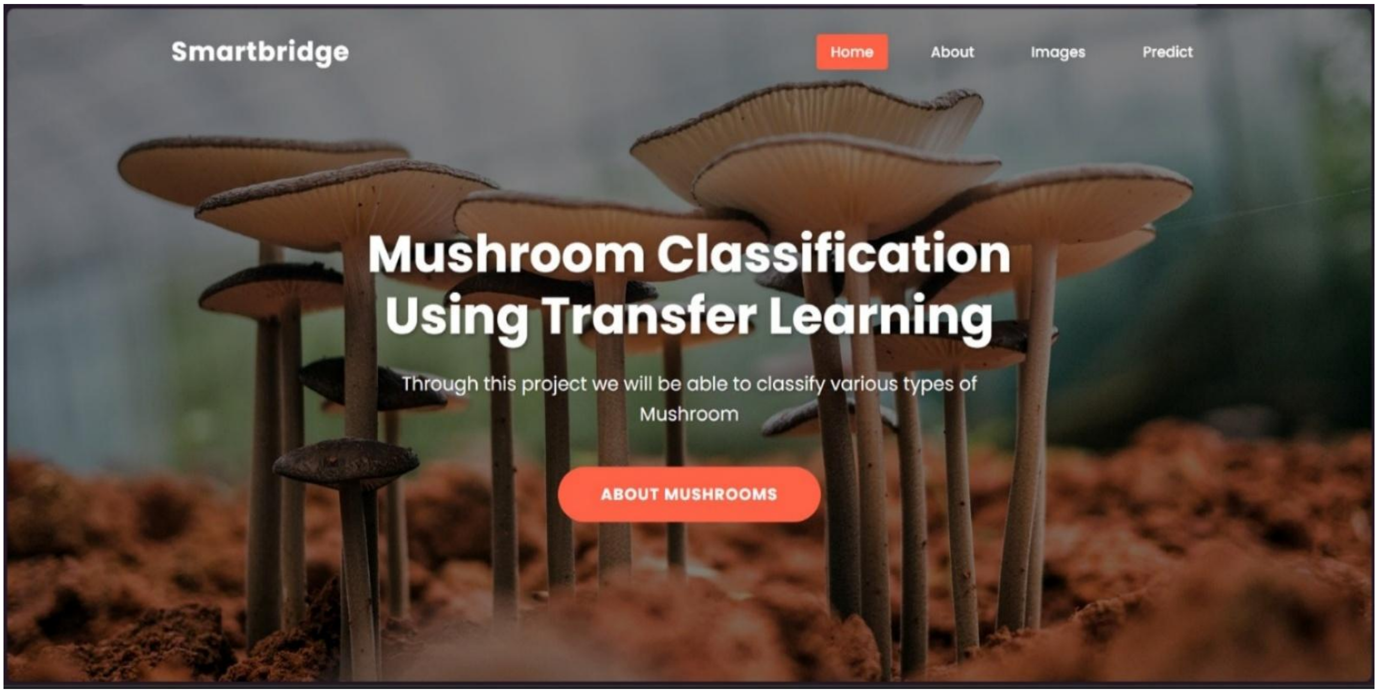
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Mushroom Classification - Smartbridge</title>
    <link rel="stylesheet" href="/static/css/styles.css">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Open+Sans:ital,wght@0,400;0,500;0,700;1,300;1,700&family=Poppins:ital,wght@0,200;0,300;0,400;0,500;0,600;0,700;1,300;1,400;1,500
</head>
<body>
    <!-- Navigation Bar -->
    <nav class="navbar">
        <div class="nav-container">
            <div class="nav-logo">Smartbridge</div>
            <ul class="nav-menu">
                <li><a href="{{ url_for('home_page') }}" class="nav-link active">Home</a></li>
                <li><a href="{{ url_for('about_page') }}" class="nav-link">About</a></li>
                <li><a href="{{ url_for('images_page') }}" class="nav-link">Images</a></li>
                <li><a href="{{ url_for('predict_page') }}" class="nav-link">Predict</a></li>
            </ul>
        </div>
    </nav>

    <!-- Hero Section -->
    <section class="hero" style="background-image: url('/static/css/background-img.jpg');">
        <div class="hero-overlay"></div>
        <div class="hero-content">
            <h1 class="hero-title">Mushroom Classification<br>Using Transfer Learning</h1>
            <p class="hero-subtitle">Through this project we will be able to classify various types of Mushroom</p>
            <a href="{{ url_for('about_page') }}" class="hero-btn">ABOUT MUSHROOMS</a>
        </div>
    </section>

    <script src="/static/js/script.js"></script>
</body>
</html>
```

input.html :



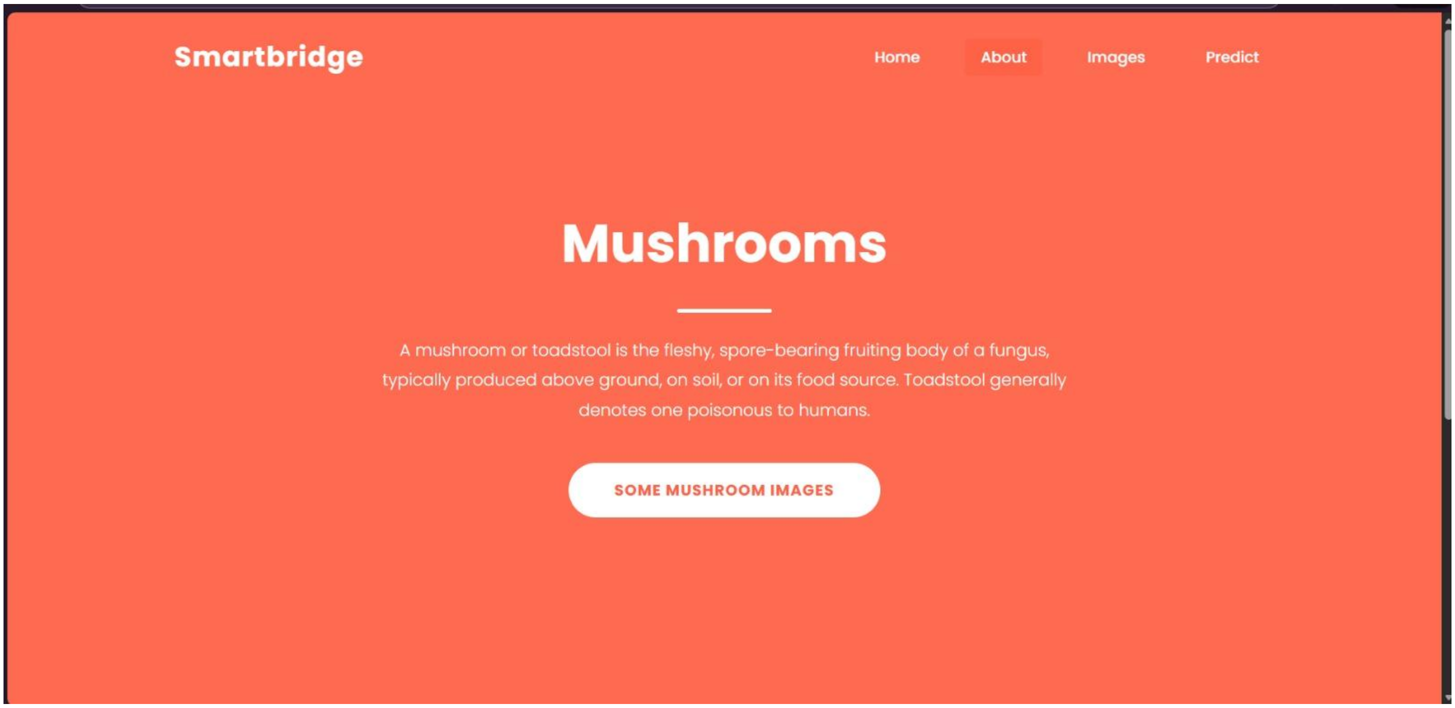
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Predict Mushroom - Smartbridge</title>
    <link rel="stylesheet" href="/static/css/styles.css">
    <link rel="stylesheet" href="/static/css/input_styles.css">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Open+Sans:ital,wght@0,400;0,500;0,700;1,300;1,700&family=Poppins:ital,wght@0,200;0,300;0,400;0,500;0,600;0,700;1,300;1,400;1,500;1,600;1,700&display=swap" rel="stylesheet">
</head>
<body>
    <!-- Navigation Bar -->
    <nav class="navbar">
        <div class="nav-container">
            <div class="nav-logo">Smartbridge</div>
            <ul class="nav-menu">
                <li><a href="{{ url_for('home_page') }}" class="nav-link">Home</a></li>
                <li><a href="{{ url_for('about_page') }}" class="nav-link">About</a></li>
                <li><a href="{{ url_for('images_page') }}" class="nav-link">Images</a></li>
                <li><a href="{{ url_for('predict_page') }}" class="nav-link active">Predict</a></li>
            </ul>
        </div>
    </nav>

    <!-- Predict Section -->
    <section class="predict-section">
        <div class="predict-container">
            <h1>Mushroom Classification</h1>
            <p class="predict-subtitle">Upload an image of a mushroom to classify it</p>

            <form action="{{ url_for('predict') }}" method="POST" enctype="multipart/form-data" class="upload-form">
                <div class="upload-area">
                    <input type="file" name="image_file" id="file_upload" class="file-input" accept="image/*" required>
                    <label for="file_upload" class="file-label">
                        <div class="upload-icon">📁</div>
                        <div class="upload-text">Click to upload or drag and drop<br><span>PNG, JPG, GIF up to 10MB</span></div>
                    </label>
                </div>

                <div id="image-preview-container" class="image-preview-container">
                    <img id="image-preview" alt="Preview">
                </div>

                <button type="submit" class="predict-btn">CLASSIFY MUSHROOM</button>
            </form>

            {% if prediction %}
                <div class="result-container">
                    <h2 class="result-title">Classification Result</h2>
                    <p class="result-text">The Mushroom is classified as:</p>
                    <div class="result-badge">{{ prediction }}</div>
                </div>
            {% endif %}
        </div>
    </section>

    <script src="/static/js/script.js"></script>
</body>
</html>
```

TESTING :



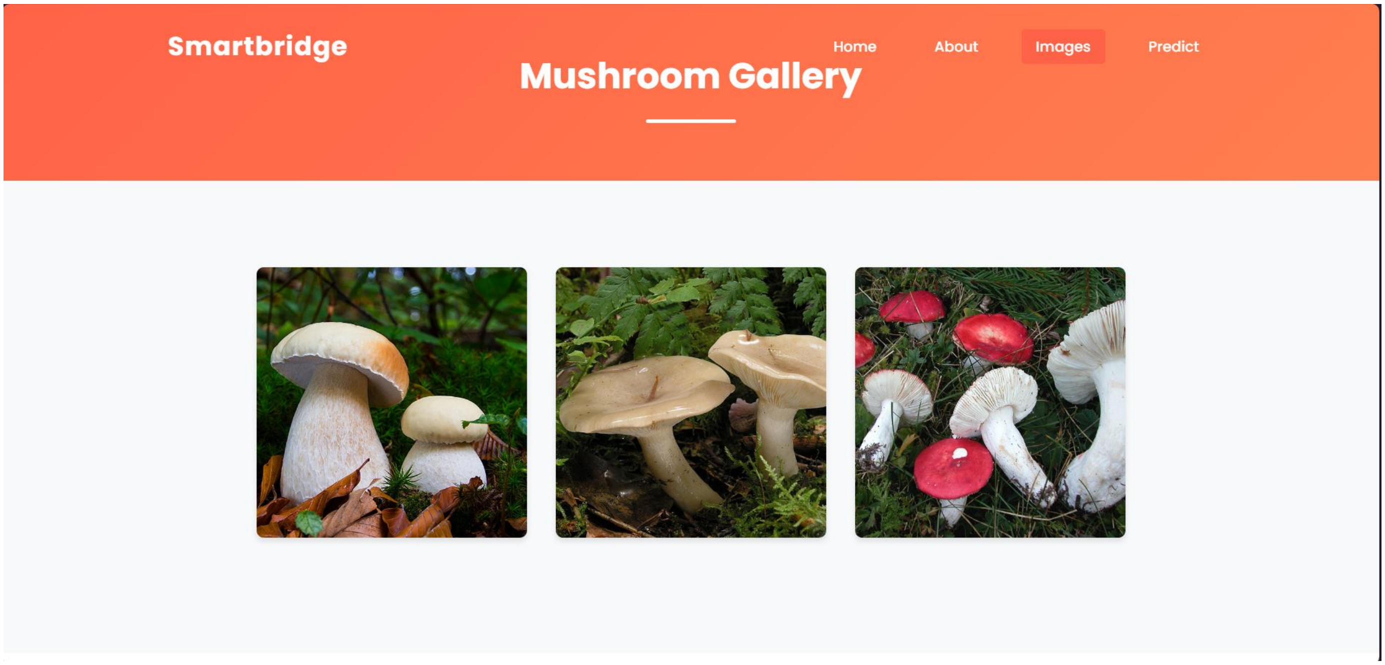
Mushroom Classification Using Transfer Learning

Through this project we will be able to classify various types of Mushroom

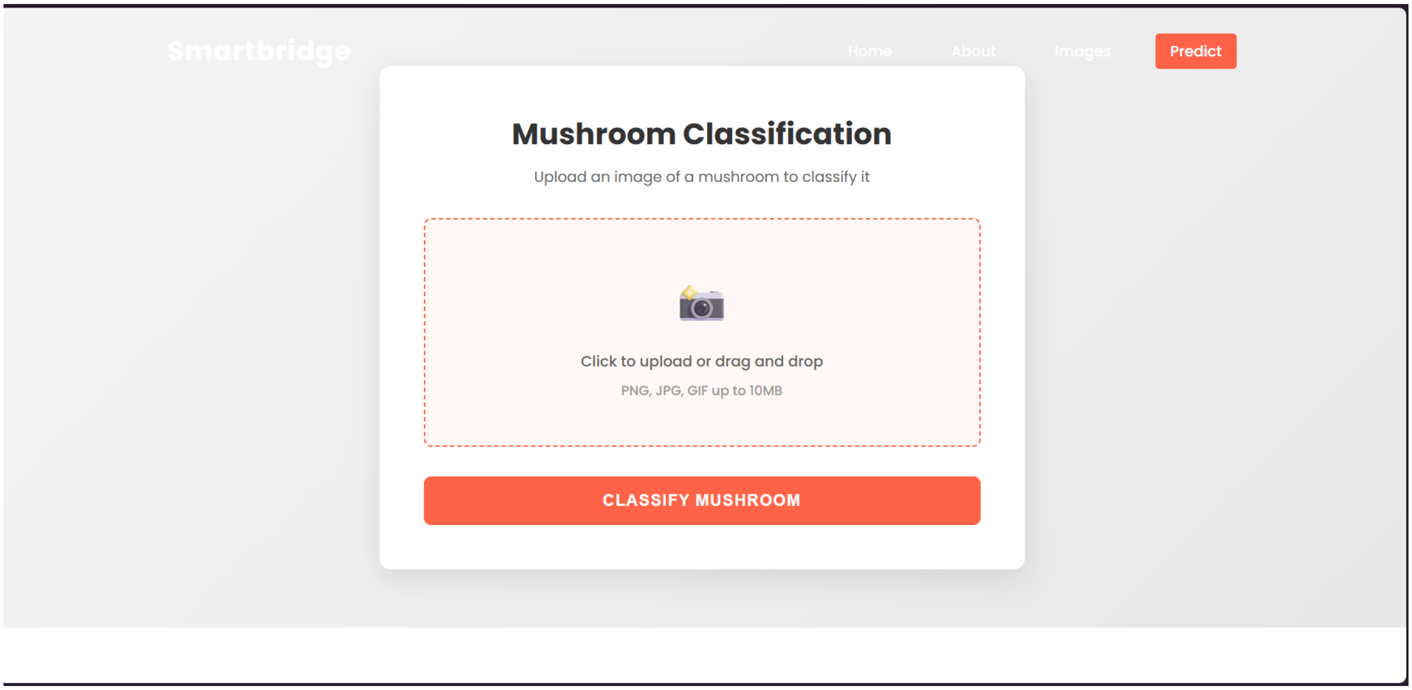


Mushrooms

A mushroom or toadstool is the fleshy, spore-bearing fruiting body of a fungus, typically produced above ground, on soil, or on its food source. Toadstool generally denotes one poisonous to humans.



Mushroom Gallery



Mushroom Classification

Upload an image of a mushroom to classify it

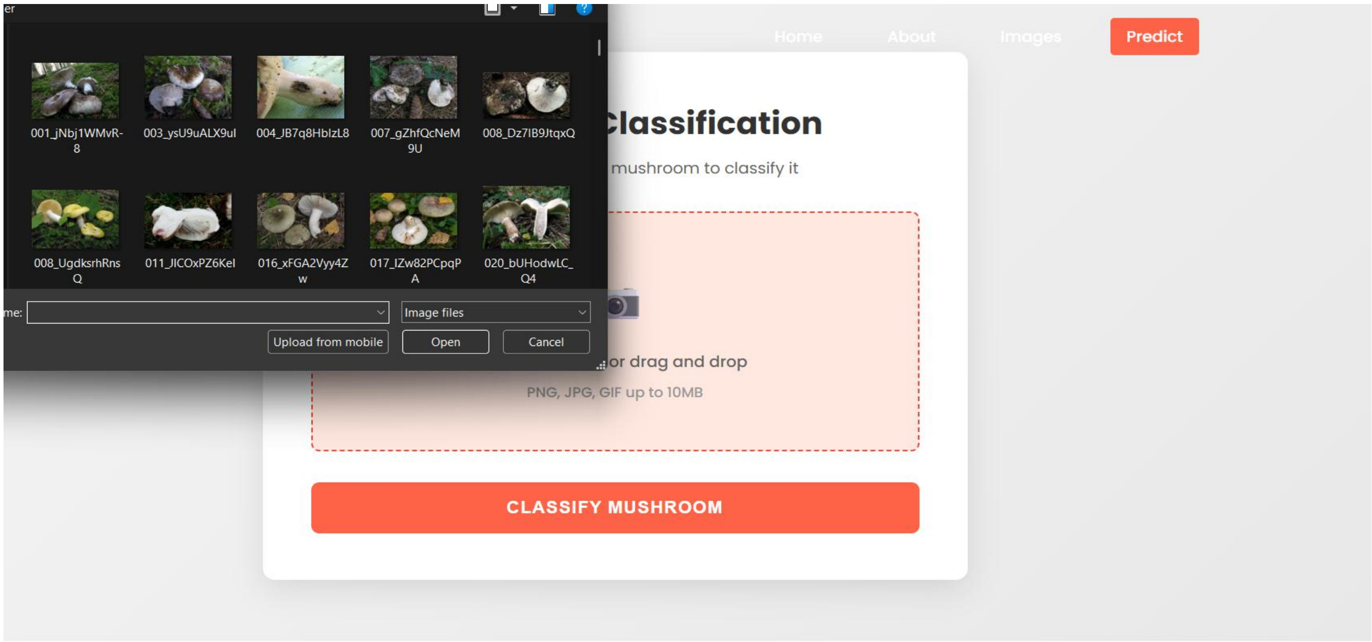
📷

Click to upload or drag and drop

PNG, JPG, GIF up to 10MB

The interface of the Uncovering the Hidden Treasures of the Mushroom Kingdom: In UI you can see the buttons like Home, About, Images, Predict.

SELECT IMAGE :



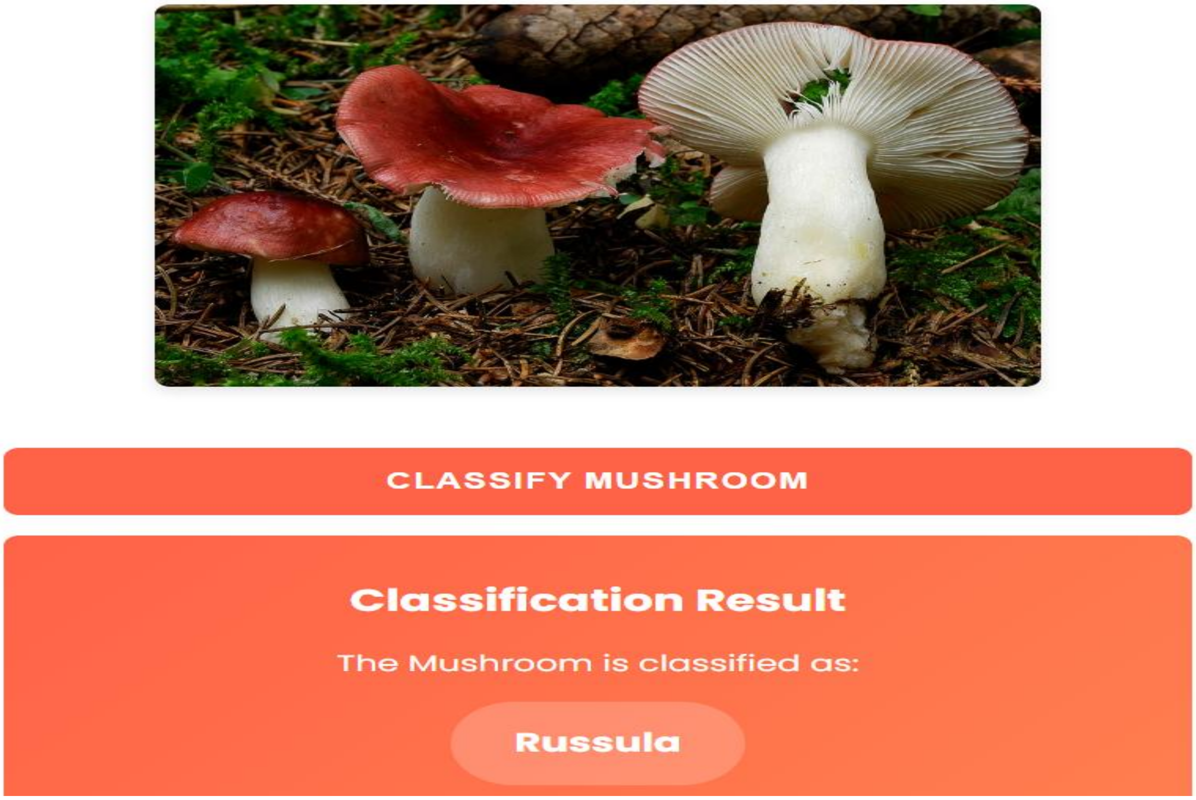
Classification

mushroom to classify it

or drag and drop

PNG, JPG, GIF up to 10MB

OUTPUT PAGE :



Classification Result

The Mushroom is classified as:

Russula

CHAPTER 5

ADVANTAGES & DISADVANTAGES

ADVANTAGES :

- Good User – Friendly interface
- Less complexity
- Optical recognition
- High-performance classifiers: Deep-learning methods (InceptionV3)
- Wide range of mushrooms image classification
- High speed mushroom image classification
- Useful for all kind of peoples

DISADVANTAGES :

- Less accessibility
- Limited scope (Classifies only 3 kinds of mushrooms only).

CHAPTER 6

CONCLUSION

In conclusion, the project focused on the optical recognition and classification of various mushroom species using deep-learning methods. By leveraging transfer learning techniques and Inception V3 model, the project aimed to achieve high-performance classification accuracy. The classification of mushrooms has a wide range of applications, including food, medicine, conservation, and ecological research. By accurately identifying mushroom species based on their physical features, The project also contributes to the advancement of mycology as a scientific discipline and enthusiasts.

CHAPTER 7

FUTURE SCOPE

❖ Future Enhancements

Future improvements may include:

- Expanding the dataset with more mushroom species.
- Fine-tuning deeper layers of the Xception model.
- Deploying the application on cloud platforms.
- Improving UI design for better user experience.
- Integrating mobile compatibility.

Overall, the project successfully fulfills its objectives and demonstrates a complete machine learning pipeline from data collection to deployment.

CHAPTER 8

APPENDIX

GitHub & Project Video Demo Link :

Git Repo Link : juien27/Uncovering-the-Hidden-Treasures-of-the-Mushroom-Kingdom-A-Classification-Analysis
Project Demo Link :

(Demonstration video)