AI BASED TOOL FOR FRUITS-VEGETABLES CLASSIFICATION

MINI PROJECT REPORT

Submitted by

ABISHEK .S (202109005)

HARIHARANVIGNESH K (202109021)

MOHAMED ASLAM K (202109034)

In partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

MEPCO SCHLENK ENGINEERING COLLEGE,
SIVAKASI-626123

(An Autonomous Institution)

ANNA UNIVERSITY:CHENNAI 600 025

OCTOBER 2023

MEPCO SCHLENK ENGINEERING COLLEGE, SIVAKASI
AUTONOMOUS

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

BONAFIDE CERTIFICATE

Certified that this project report "**AI BASED TOOL FOR FRUITS-VEGETABLE CLASSIFICATION**" is the bonafide work of **MOHAMED ASLAM.K (202109034), ABISHEK.S (202109005), HARIHARANVIGNESH K (202109021)** who carried out the project work under my supervision.

SIGNATURE

Ms.L.Prasika, BE, ME, (PhD)
Assistant Professor
Artificial Intelligence and Data Science
Mepco Schlenk Engineering College,
Sivakasi- 626 005,Virudhunagar

SIGNATURE

Dr.J.Angela Jennifa Sujana,
Professor & Head of the Department
Artificial Intelligence and Data Science
Mepco Schlenk Engineering College,
Sivakasi- 626 005,Virudhunagar

The project report submitted for the viva voce held on …….

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

First and foremost, we offer our heartfelt praise and gratitude to "**The Almighty,**" the Lord of all creation, who, through His abundant grace, has sustained and enabled us to successfully undertake this project.

We take immense pleasure in extending our sincere thanks to our esteemed management members, who serve as the cornerstone of our college.

We wish to express our deep appreciation to the respected Principal,

Dr. S. Arivazhagan, M.E., Ph.D., for providing the necessary facilities for our mini project.

We are profoundly thankful to our Head of the Department, **Dr. J. Angela Jennifa Sujana, M.E., Ph.D.,** Professor (SG), and Head of the Department of Artificial Intelligence and Data Science, for her invaluable guidance and unwavering support throughout our mini project.

Our gratitude also goes out to our guide, **Ms. L. Prasika, B.E., M.E., (PhD),** an Assistant Professor in the Department of Artificial Intelligence and Data Science, for her valuable guidance. It is a great privilege to acknowledge her contribution.

We extend our heartfelt thanks and deep appreciation to all the faculty members of the Department of Artificial Intelligence and Data Science for their generous assistance during the course of our mini project.

Last but not least, we are grateful to our parents and friends for their constant support during the execution of our mini project.

TABLE OF CONTENTS

ABSTRACT

This project integrates two crucial components: a sophisticated deep learning image classifier for fruits and vegetables and an intuitive web application. The initial Python code, powered by TensorFlow, constructs a robust image classification model. It systematically prepares a diverse dataset of fruits and vegetables, designs a model architecture rooted in MobileNetV2, and meticulously trains it using advanced data augmentation techniques. The code is equipped with a range of functions for dataset management, model development, training, and real-time image predictions.

In parallel, the second component comprises a dynamic Streamlit web application, which harnesses the pretrained image classification model. This application empowers users to effortlessly upload images for classification into fruit or vegetable categories. Furthermore, it endeavours to source additional nutritional information, such as calorie data, for recognized objects. The user-friendly interface enables users to interact with the model, providing quick and accurate category predictions. When available, nutritional details are seamlessly presented, enhancing the user experience.

By harmoniously merging these two components, this project delivers a comprehensive and user-centric solution for accurate image-based classification and a deeper understanding of fruits and vegetables. The combined system provides practical benefits for diverse users, making it a valuable tool for educational, culinary, and nutritional purposes.

CHAPTER 1

INTRODUCTION

In the age of artificial intelligence and machine learning, technological advancements have opened new horizons in the field of image recognition and classification. The precise identification of objects within images has found diverse applications across numerous domains, spanning agriculture, education, and dietary planning. Among these applications, the classification of fruits and vegetables based on images stands as an intriguing and highly practical endeavor.

This project embarks on a journey that seamlessly integrates two fundamental components: a deep learning image classifier for fruits and vegetables and an intuitive web application. By harnessing the capabilities of TensorFlow and cutting-edge neural networks, we have developed a robust image classification model with the capacity to discern an extensive array of fruits and vegetables accurately.

The initial phase of this project centers on the construction of the image classifier. It entails the meticulous curation of a diverse dataset comprising various fruits and vegetables, the formulation of a model architecture rooted in MobileNetV2, and the refinement of its performance through the application of advanced data augmentation techniques. This segment of the project forms the cornerstone for precise image recognition and classification.

Complementing this formidable classifier is a dynamic Streamlit web application, designed to place this technology directly in the hands of end-users. This user-friendly application facilitates the straightforward uploading of images for instant classification into fruit or vegetable categories. Moving beyond mere classification, the application takes a step further by providing additional nutritional insights, such as calorie content, linked to the identified objects.

This project represents the fusion of state-of-the-art technology and user-centric design. It empowers educational institutions, culinary aficionados, and health-conscious individuals alike by simplifying the identification of fruits and vegetables while enhancing nutritional awareness. The application's dynamic nature ensures adaptability to evolving needs and allows for future enhancements.

In the following sections, we delve deeper into the technical intricacies of the image classification model and the user-friendly application, showcasing how these components seamlessly merge to provide a comprehensive and user-centric solution for accurate image-based classification and a deeper understanding of fruits and vegetables.

1.1.SCOPE OF THE PROJECT

The scope of this project encompasses the development of a robust image classification model for fruits and vegetables, seamlessly integrated with a user-friendly web application. The project is designed to enable accurate real-time classification of a wide range of produce, providing users with an intuitive platform for image uploads and category predictions. Beyond classification, the application strives to offer nutritional insights, making it a valuable resource for educational, culinary, and dietary planning. Its scalability ensures the potential for future enhancements, and it represents an innovative convergence of technology and user-centric design.

1.2.OBJECTIVE OF THE PROJECT

The primary objective of this project is to develop a robust image classification model for accurate recognition of a wide range of fruits and vegetables, seamlessly integrated with a user-friendly web application. This application will empower users to upload images and receive instant category predictions while

also offering nutritional insights, making it a versatile tool for educational, culinary, and dietary purposes. Additionally, the project is designed to be scalable, allowing for future enhancements and symbolizes the convergence of cutting-edge technology with user-centric design.

1.3.REPORT SUMMARY

This project combines a robust image classification model for fruits and vegetables with an intuitive web application. Key highlights include the development of an accurate deep learning model for image recognition, user-friendly interaction, and the provision of nutritional information. The model, driven by TensorFlow and MobileNetV2, offers real-time classification. The application, designed with Streamlit, allows easy image uploads for category predictions and nutritional insights, making it a versatile tool for education, culinary use, and dietary planning. Scalable for future enhancements, the project signifies the convergence of technology and user-centric design.

CHAPTER 2

MODULES

2.1.INTRODUCTION

In this project, several key modules are employed to facilitate various aspects of the endeavor. NumPy and Pandas serve as the foundation for data manipulation and analysis, while TensorFlow and Keras power the development of a deep learning image classification model. Matplotlib aids in data visualization, showcasing images and results. Streamlit emerges as the user-friendly interface creation tool, while PIL (Python Imaging Library) manages image processing. Additionally, Beautiful Soup (bs4) is used for web scraping, enabling the retrieval of nutritional information. These modules collectively contribute to a comprehensive solution for image classification and user interaction, highlighting the convergence of data science, machine learning, and user-centric design.

2.2 Modules

2.2.1 NumPy and Pandas:

NumPy and Pandas are fundamental libraries for data manipulation and analysis in Python. NumPy provides support for numerical operations, while Pandas is used for structured data handling, such as data frames. In this code, they play a crucial role in processing and managing data related to fruits and vegetables.

2.2.2 TensorFlow and Keras:

TensorFlow is an essential open-source machine learning framework. Keras, integrated into TensorFlow, simplifies the construction and training of neural networks. In this code, they are used to create and train the deep learning model responsible for image classification.

2.2.3 Matplotlib:

Matplotlib is a popular data visualization library in Python. It's used to display images, plots, and graphs in the Jupyter Notebook. In the code, it's employed for visualizing sample images from the dataset.

2.2.4 Streamlit:

Streamlit is a user-friendly library for creating web applications with minimal effort. In the code, it's utilized to develop the interactive web interface that allows users to upload images and receive real-time classification results.

2.2.5 PIL (Python Imaging Library):

PIL is a module within the Python Imaging Library that provides capabilities for working with images. In the code, it's used for image handling and conversion within the Streamlit application.

2.2.6 bs4 (Beautiful Soup):

Beautiful Soup is a library for web scraping and parsing HTML and XML documents. In the code, it's employed to fetch nutritional information related to recognized objects.

These modules collectively form the backbone of the code, enabling data processing, machine learning model development, user interaction through the web application, and web scraping for supplementary information.

CHAPTER 3

SYSTEM ARCHITECTURE

3.1. INTRODUCTION

- The proposed model at the core of this project represents a sophisticated and effective solution for image classification, specifically geared towards the accurate recognition of fruits and vegetables. This model is a deep learning neural network constructed using the TensorFlow framework and enhanced with the MobileNetV2 architecture. Its primary function is to precisely categorize a diverse range of produce based on input images.
- MobileNetV2 is a pre-trained neural network designed for image feature extraction and classification, making it an ideal choice for this project. The model is further refined through the application of advanced data augmentation techniques, ensuring its robustness and adaptability in real-world scenarios.
- The proposed model plays a pivotal role in empowering users to identify and classify fruits and vegetables effortlessly, whether for educational, culinary, or nutritional purposes. Its versatility and accuracy make it a valuable tool within the larger project's framework, demonstrating the potential of cutting-edge machine learning in simplifying tasks and enhancing understanding of the world around us.

3.2 MACHINE LEARNING

- **Supervised Learning:** The project employs supervised learning, where the model is trained on a labeled dataset containing images of fruits and vegetables, with corresponding class labels (e.g., "apple," "banana").

- **Deep Learning:** The core of the project relies on deep learning, specifically Convolutional Neural Networks (CNNs). CNNs are used to automatically extract relevant features from images and make accurate predictions.
- **Transfer Learning:** The project leverages transfer learning by using the pre-trained MobileNetV2 model. Transfer learning enables the model to benefit from knowledge gained during training on large image datasets, enhancing its performance.
- **Data Augmentation:** Data augmentation is applied to artificially increase the diversity of the dataset. This involves generating variations of the images (e.g., rotations, flips) to help the model generalize better and reduce overfitting.
- **Image Preprocessing:** Image preprocessing techniques are used to prepare the input data for the model. This includes resizing images to a consistent size and normalizing pixel values to ensure consistent data input.
- **Categorical Cross-Entropy Loss:** The categorical cross-entropy loss function is employed as the objective function for training the model. It quantifies the dissimilarity between predicted class probabilities and actual class labels, facilitating multi-class classification.
- **Adam Optimization:** The Adam optimization algorithm is utilized to update the model's weights during training. Adam adapts the learning rate for each parameter, which helps the model converge faster and often results in better training results.
- **Early Stopping:** The project incorporates early stopping as a form of regularization. Early stopping halts the training process when the model's performance on a validation dataset begins to degrade, preventing overfitting. These machine learning concepts collectively contribute to the development of a robust image classification model capable of accurately categorizing fruits and vegetables based on input images.

3.3. SYSTEM DESIGN

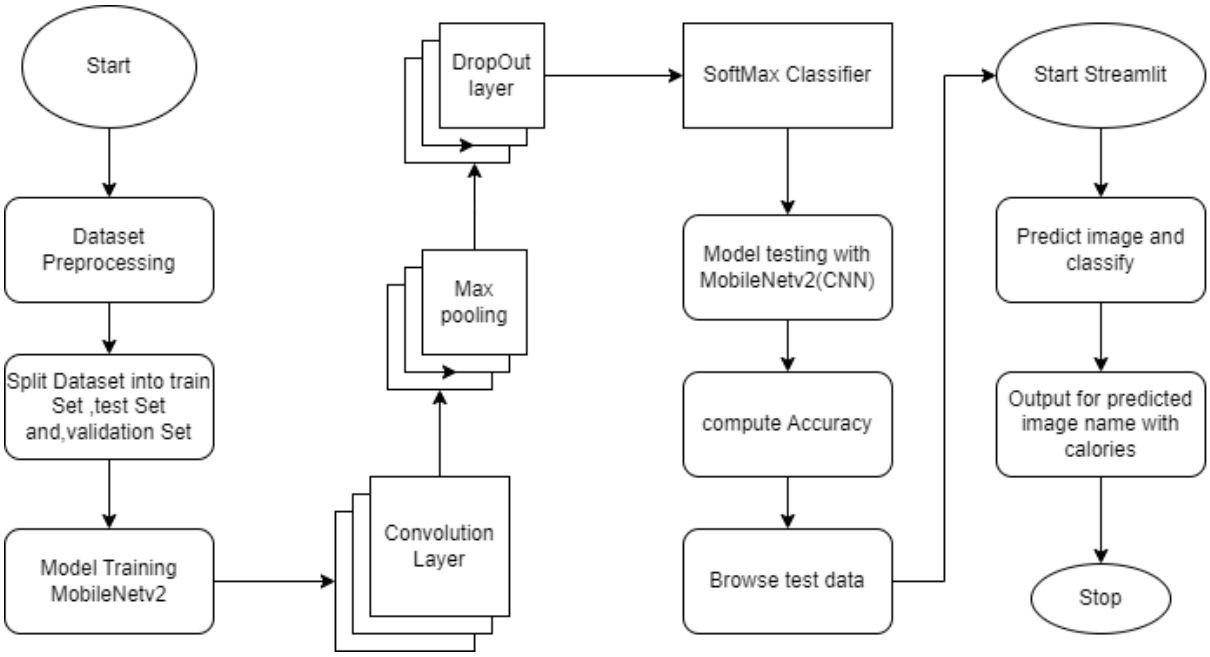

Fig -3.1 System Design

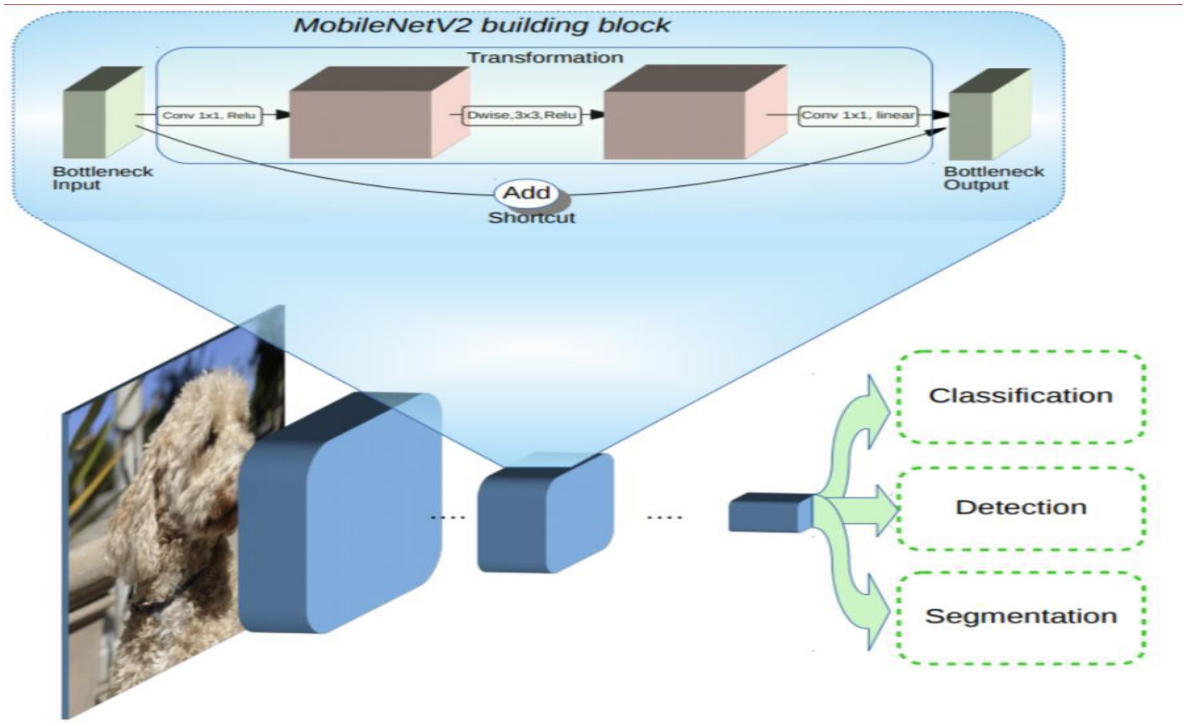

Fig -3.2 MobileNetV2 Architecture

ALGORITHM

Fruit-image-classifier.ipynb

1. **Define an image processing function:**

   Create a function image_processing that takes a list of file paths and extracts labels from the file paths.

   Generate a Pandas DataFrame with file paths and labels, shuffle the DataFrame, and return it.

2. **Process training, testing, and validation datasets:**

   Use the image_processing function to create dataframes for each dataset.

3. **Data exploration:**

   Display basic information about the training dataset, such as the number of images, the number of unique labels, and the unique labels themselves.

   Display a sample of images from the dataset.

4. **Set up image generators:**

   Create data generators for training, validation, and testing datasets.

   Use data augmentation techniques like rotation, zoom, and shifting for the training data.

5. **Build and compile the model:**

   Load a pretrained MobileNetV2 model with specified weights.

   Add custom fully connected layers for classification.

   Compile the model with an optimizer, loss function, and metrics.

6. **Train the model:**

   Train the model on the training dataset using the fit function.

7. **Make predictions:**

   Predict labels for the test dataset.

8. **Define an image classification function:**

   Create a function that takes an image location as input, preprocesses the image, and uses the model to make predictions.

Algorithm for fvclassifier.py:

1. **Define labels and categories:**

   Create dictionaries for labels and categories of fruits and vegetables.

2. **Define a function to fetch calories:**

   Create a function that fetches calorie information for a given prediction using a web search.

3. **Define a function for image processing:**

   Create a function that takes an image path, preprocesses the image, and uses the pre-trained model to make predictions.

4. **Create a Streamlit web application:**

   Create a Streamlit app to upload an image and make predictions.

   Display the uploaded image and predicted category

   Fetch and display calorie information for the predicted item.

5. **Run the Streamlit application:**

   Start the Streamlit app.

CHAPTER 4

IMPLEMENTATION

4.1 SOURCE CODE

FRUIT-IMAGE-CLASSIFIER.IPYNB

```
import numpy as np

import pandas as pd

import os

for dirname, _, filenames in os.walk('E:\Dataset'):

    for filename in filenames:

        print(os.path.join(dirname, filename))

import pandas as pd

import numpy as np

from pathlib import Path

import os.path

import matplotlib.pyplot as plt

import tensorflow as tf

from      tensorflow.keras.preprocessing.image      import
load_img,img_to_array

print(tf.__version__)

train_data=Path('E:/Dataset/train')

train_filepath = []

for subdirectory in train_data.iterdir():

    if subdirectory.is_dir():

        for file in subdirectory.glob('**/*.jpg'):
```

```
            train_filepath.append(file)
train_filepath


test_data = Path('E:/Dataset/test')
test_filepath = []
for subdirectory in test_data.iterdir():
    if subdirectory.is_dir():
        for file in subdirectory.glob('**/*.jpg'):
            test_filepath.append(file)
test_filepath


val_data=Path('E:/Dataset/validation')
val_filepath = []
for subdirectory in val_data.iterdir():
    if subdirectory.is_dir():
        for file in subdirectory.glob('**/*.jpg'):
            val_filepath.append(file)
val_filepath


def image_processing(fl_path):
    labels=[str(fl_path[i]).split("\\")[-2]for          iin
range(len(fl_path))]
    file_path   =   pd.Series([fl_path[i]   for   i   in
range(len(fl_path))], name='Filepath').astype(str)
    label = pd.Series(labels, name="Label")
```

```
    df = pd.concat([file_path, label], axis=1)
    df = df.sample(frac=1).reset_index(drop=True)
    return df

train_df=image_processing(train_filepath)
test_df=image_processing(train_filepath)
val_df=image_processing(val_filepath)

print("--Training set--")
print("Number of Images:",train_df.shape[0])
print("Number of Labels:",len(train_df.Label.unique()))
print("Labels:",train_df.Label.unique())

df_unique=train_df.copy().drop_duplicates(subset=["Label"])
.reset_index()
fig,axes=plt.subplots(nrows=6,ncols=6,figsize=(8,7),subplot
_kw={'xticks':[],'yticks':[]})
for i,ax in enumerate(axes.flat):
    ax.imshow(plt.imread(df_unique.Filepath[i]))
    ax.set_title(df_unique.Label[i],fontsize=12)
plt.tight_layout(pad=0.5)
plt.show()

train_generator                                          =
tf.keras.preprocessing.image.ImageDataGenerator(
```

```
preprocessing_function=tf.keras.applications.mobilenet_v2.prepro
cess_input)

    test_generator=tf.keras.preprocessing.image.ImageDataGenera
    tor(
    preprocessing_function=tf.keras.applications.mobilenet_v2.p
    reprocess_input)


    train_images = train_generator.flow_from_dataframe(
        dataframe=train_df,
        x_col='Filepath',
        y_col='Label',
        target_size=(224, 224),
        color_mode='rgb',
        class_mode='categorical',
        batch_size=32,
        shuffle=True,
        seed=0,
        rotation_range=30,
        zoom_range=0.15,
        width_shift_range=0.2,
        height_shift_range=0.2,
        shear_range=0.15,
        horizontal_flip=True,
        fill_mode="nearest")
    val_images = train_generator.flow_from_dataframe(
```

```
        dataframe=val_df,
        x_col='Filepath',
        y_col='Label',
        target_size=(224, 224),
        color_mode='rgb',
        class_mode='categorical',
        batch_size=32,
        shuffle=True,
        seed=0,
        rotation_range=30,
        zoom_range=0.15,
        width_shift_range=0.2,
        height_shift_range=0.2,
        shear_range=0.15,
        horizontal_flip=True,
        fill_mode="nearest")
    test_images = test_generator.flow_from_dataframe(
        dataframe=test_df,
        x_col='Filepath',
        y_col='Label',
        target_size=(224, 224),
        color_mode='rgb',
        class_mode='categorical',
        batch_size=32,
```

```
    shuffle=False)
pretrained_model = tf.keras.applications.MobileNetV2(
    input_shape=(224, 224, 3),
    include_top=False,
    weights='imagenet',
    pooling='avg')
pretrained_model.trainable = False
inputs = pretrained_model.input
x=tf.keras.layers.Dense(128,activation='relu')(pretrained_model.output)
x = tf.keras.layers.Dense(128, activation='relu')(x)
outputs = tf.keras.layers.Dense(36, activation='softmax')(x)
model = tf.keras.Model(inputs=inputs, outputs=outputs)
model.compile(optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy'])

history = model.fit(
    train_images,
    validation_data=val_images,
    batch_size = 32,
    epochs=5,
    callbacks=[
        tf.keras.callbacks.EarlyStopping(
```

```
                monitor='val_loss',
                patience=2,
                restore_best_weights=True)])
    pred = model.predict(test_images)
    pred = np.argmax(pred,axis=1)
    labels = (train_images.class_indices)
    labels = dict((v,k) for k,v in labels.items())
    pred1 = [labels[k] for k in pred]
    def output(location):
        img=load_img(location,target_size=(224,224,3))
        img=img_to_array(img)
        img=img/255
        img=np.expand_dims(img,[0])
        answer=model.predict(img)
        y_class = answer.argmax(axis=-1)
        y = " ".join(str(x) for x in y_class)
        y = int(y)
        res = labels[y]
        return res
```

FVCLASSIFIER.PY

```
    import streamlit as st
    from PIL import Image
    from keras.preprocessing.image import load_img, img_to_array
    import numpy as np
```

```
from keras.models import load_model

import requests

from bs4 import BeautifulSoup

model = load_model('C:/Users/ASUS/Desktop/AI LAB/ML PROJECT/FV.h5')

labels = {0: 'apple', 1: 'banana', 2: 'beetroot', 3: 'bell pepper', 4: 'cabbage', 5: 'capsicum', 6: 'carrot', 7: 'cauliflower', 8: 'chilli pepper', 9: 'corn', 10: 'cucumber', 11: 'eggplant', 12: 'garlic', 13: 'ginger', 14: 'grapes', 15: 'jalepeno', 16: 'kiwi', 17: 'lemon', 18: 'lettuce', 19: 'mango', 20: 'onion', 21: 'orange', 22: 'paprika', 23: 'pear', 24: 'peas', 25: 'pineapple', 26: 'pomegranate', 27: 'potato', 28: 'raddish', 29: 'soy beans', 30: 'spinach', 31: 'sweetcorn', 32: 'sweetpotato', 33: 'tomato', 34: 'turnip', 35: 'watermelon'}

fruits = ['Apple', 'Banana', 'Bello Pepper', 'Chilli Pepper', 'Grapes', 'Jalepeno', 'Kiwi', 'Lemon', 'Mango', 'Orange','Paprika', 'Pear', 'Pineapple', 'Pomegranate', 'Watermelon']

vegetables = ['Beetroot', 'Cabbage', 'Capsicum', 'Carrot', 'Cauliflower', 'Corn', 'Cucumber', 'Eggplant', 'Ginger','Lettuce', 'Onion', 'Peas', 'Potato', 'Raddish', 'Soy Beans', 'Spinach', 'Sweetcorn', 'Sweetpotato','Tomato', 'Turnip']

def fetch_calories(prediction):

    try:

        url = 'https://www.google.com/search?&q=calories in ' + prediction

        req = requests.get(url).text
```

```
        scrap = BeautifulSoup(req, 'html.parser')
        calories = scrap.find("div", class_="BNeawe iBp4i AP7Wnd").text
        return calories
    except Exception as e:
        st.error("Can't able to fetch the Calories")
        print(e)
def processed_img(img_path):
    img = load_img(img_path, target_size=(224, 224, 3))
    img = img_to_array(img)
    img = img / 255
    img = np.expand_dims(img, [0])
    answer = model.predict(img)
    y_class = answer.argmax(axis=-1)
    print(y_class)
    y = " ".join(str(x) for x in y_class)
    y = int(y)
    res = labels[y]
    print(res)
    return res.capitalize()
def run():
    st.title("Fruits🍍-Vegetable🍅 Classification")
    img_file = st.file_uploader("Choose an Image", type=["jpg", "png"])
    if img_file is not None:
```

```
            img = Image.open(img_file).resize((250, 250))
            st.image(img, use_column_width=False)
            save_image_path='C:/Users/ASUS/Desktop/AILAB/ML
PROJECT/upload_images/' + img_file.name
            with open(save_image_path, "wb") as f:
                f.write(img_file.getbuffer())
            # if st.button("Predict"):
            if img_file is not None:
                result = processed_img(save_image_path)
                print(result)
                if result in vegetables:
                    st.info('**Category : Vegetables**')
                else:
                    st.info('**Category : Fruit**')
                st.success("**Predicted : " + result + '**')
                cal = fetch_calories(result)
                if cal:
                    st.warning('**' + cal + '(100 grams)**')
run()
```

4.2 FUNCTION EXPLANATION

1. Image Processing:

Define a function image_processing(fl_path) to process image file paths and labels.

Extract labels from file paths by splitting the path and taking the second-to-last component (the fruit or vegetable name).

Create a Pandas DataFrame with two columns: 'Filepath' and 'Label'.

Shuffle the DataFrame and return it. This function essentially prepares your data for training.

2. Process Datasets:

Use the image_processing function to process your training, testing, and validation datasets.

Print out information about the training dataset, including the number of images, the number of unique labels, and the unique labels themselves.

Visualize a sample of the training dataset by displaying 36 images and their corresponding labels in a 6x6 grid using Matplotlib.

3. Data Augmentation and Image Generators:

Create image data generators for training, validation, and testing datasets using TensorFlow's ImageDataGenerator.

Apply data augmentation techniques such as rotation, zoom, shift, shear, and horizontal flip to the training data to enhance model generalization.

4. Model Building:

Load a pre-trained MobileNetV2 model from TensorFlow Hub.

Define a custom top layer with three fully connected (Dense) layers.

The model has 36 output units, representing the 36 different fruit and vegetable classes.

5. Model Compilation:

Compile the model with the Adam optimizer, categorical

cross-entropy loss, and accuracy as the metric.

6. Model Training:

Train the model using fit function with training and validation data.

The training stops early if the validation loss doesn't improve after a certain number of epochs (early stopping).

7. Model Evaluation:

Make predictions on the test dataset and convert them to class labels.

8. Prediction Function:

Define a function output(location) that takes the location of an image file as input.

The function loads and preprocesses the image, then uses the model to predict the class label and returns it. These functional explanations provide a step-by-step overview of the key components and processes in the code, from data preparation and model building to training and prediction.

fvclassifier.py

1. fetch_calories(prediction):

This function is responsible for fetching nutritional information, specifically calorie data, for a recognized object. It takes the predicted object's label as input (referred to as 'prediction') and constructs a Google search query to find the calorie information

associated with that object. It uses web scraping with Beautiful Soup to extract the calorie data from the search results and returns it.

2. processed_img(img_path):

This function is designed to process an image located at a given file path. It loads the image, converts it to an array, normalizes the pixel values, and prepares it for input to the pre-trained deep learning model. It then utilizes the loaded model to predict the category of the object in the image and returns the human-readable label of the predicted category.

3. run():

This function serves as the main entry point for the Streamlit web application. It encompasses the user interface and functionality of the application. Users can upload images, and upon pressing the "Predict" button, the processed_img function is called to classify the image. The predicted label is displayed to the user, and, if available, the calorie content of the recognized object is fetched using the fetch_calories function and presented as well.

These functions collectively form the core of the Streamlit web application, enabling users to upload images for classification and retrieve additional nutritional information related to the recognized objects.

CHAPTER 5

RESULTS

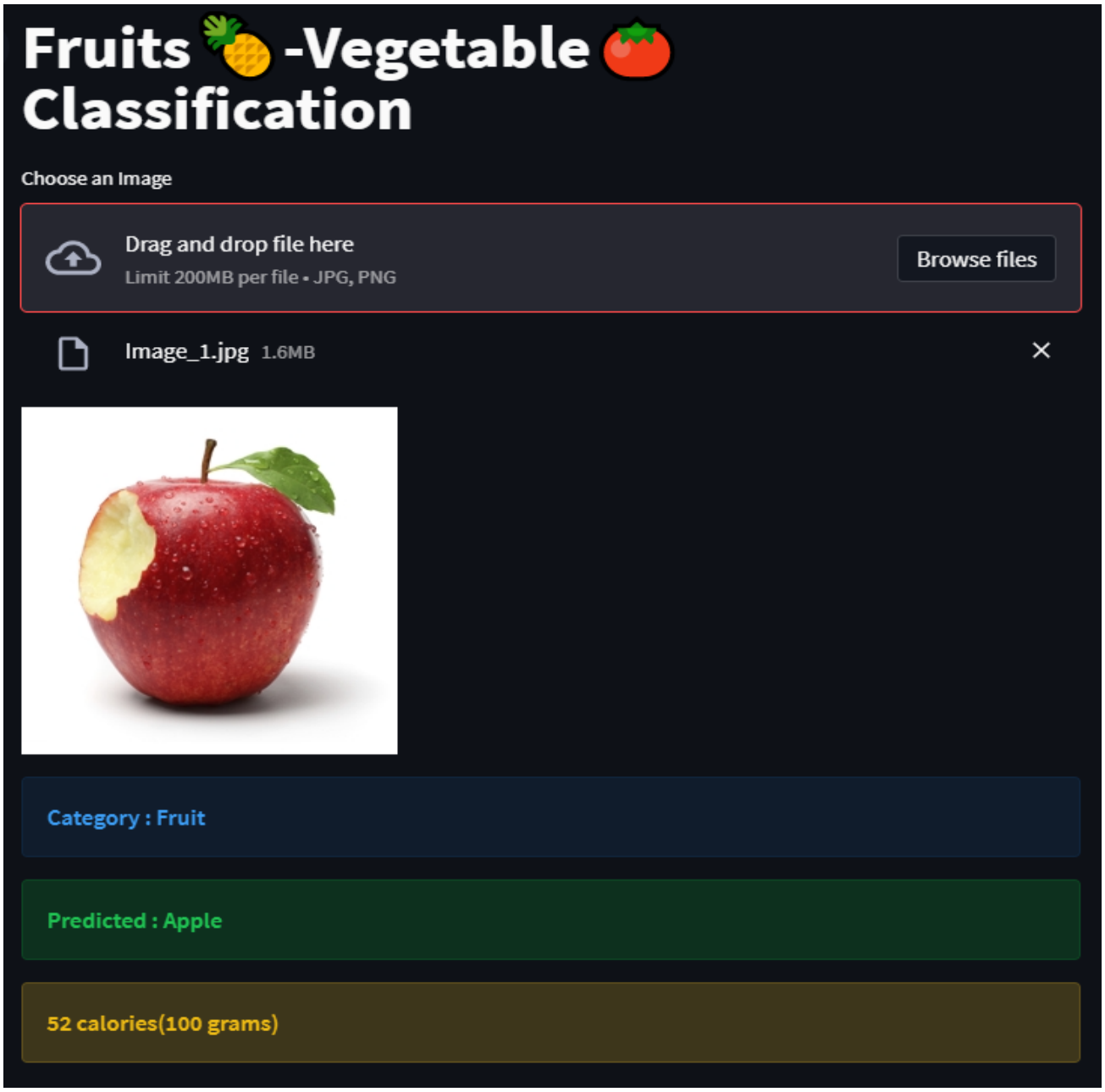

Fig-5.1 Fruit Classification with Calories

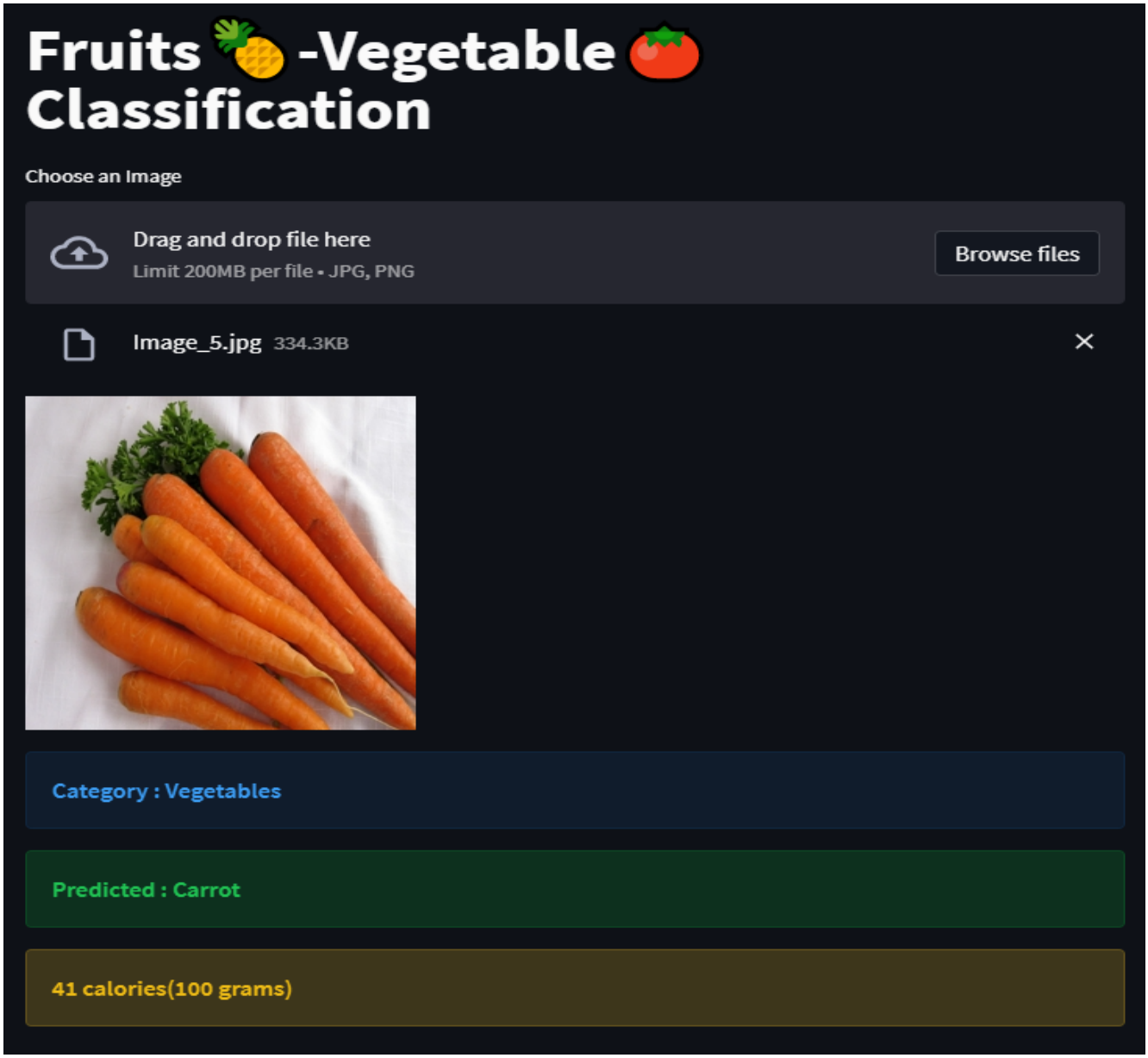

Fig-5.2 Vegetable Classification with Calories

CHAPTER 6

CONCLUSION

This project marks a harmonious convergence of cutting-edge technology, deep learning, and user-centric design. At its core, the deep learning model, powered by TensorFlow and harnessed by the MobileNetV2 architecture, showcases the remarkable capabilities of machine learning in the precise categorization of an extensive spectrum of fruits and vegetables. The utilization of advanced techniques, including data augmentation and transfer learning, bolsters the model's robustness and performance, making it a reliable tool for image classification.

Complementing the technical prowess of the model, the user-friendly web application, crafted with Streamlit, delivers a seamless and intuitive platform for users with diverse technical backgrounds. The application empowers individuals to upload images, receive instant classification results, and provides a valuable feature by offering nutritional insights. This multi-faceted approach enhances user awareness, enabling them to make more informed dietary choices.

This project extends its benefits across a spectrum of domains. In education, it stands as an interactive tool, aiding both students and educators in exploring the botanical world. Culinary enthusiasts find value in its ability to identify ingredients for recipes, while health-conscious individuals utilize it for making thoughtful dietary decisions.

With scalability ingrained into its design, the project holds the potential for future enhancements, promising to adapt and evolve to cater to evolving needs. It is a dynamic and adaptable solution that testifies to the real-world application of technology.

In summary, this project embodies the fusion of state-of-the-art technology and user-driven design. It serves as a testament to the practical benefits of artificial

intelligence and machine learning, simplifying tasks, and expanding our comprehension of the intricate world of fruits and vegetables. It stands as a valuable resource, offering practical advantages to a diverse range of users in their daily lives.