Project Report

1. INTRODUCTION

1.1 Project Overview

The “Smart Sorting” project is a machine learning-based system that aims to automate the identification and classification of rotten and healthy fruits and vegetables using transfer learning. This project utilizes pre-trained deep learning models to efficiently analyse and categorize images of produce, making the sorting process faster, more accurate, and less labour - intensive. The system is trained using image datasets of various fruits and vegetables in both healthy and rotten conditions. By leveraging transfer learning, the model benefits from the knowledge of a previously trained network, significantly reducing training time and improving accuracy even with limited data.

This application can be particularly beneficial in agricultural industries, supermarkets, and food distribution centres, where visual inspection is often done manually. The web interface allows users to upload images and get instant predictions, enabling real-time decision-making. Overall, the project brings together computer vision, deep learning, and practical implementation through a user-friendly web interface to solve a real-world problem that affects food quality and safety.

1.2 Purpose

The primary purpose of the “Smart Sorting” project is to automate the process of identifying and sorting rotten fruits and vegetables using advanced machine learning techniques. Manual inspection of produce is often time-consuming, inconsistent, and prone to human error. This project addresses that challenge by using a trained deep learning model that can analyse images of fruits and vegetables and accurately classify them as healthy or rotten. By implementing transfer learning, the system leverages pre-trained models, making the training process faster and more efficient while achieving high accuracy.

This solution is designed to support industries like agriculture, food retail, and supply chain management, where rapid and accurate sorting of produce is crucial for maintaining quality standards and reducing waste. The project also promotes food safety by ensuring that only fresh produce reaches consumers. Additionally, the web-based interface allows for easy accessibility, so users can upload images and get immediate feedback, making the technology practical and scalable for real-world applications.

2. IDEATION PHASE

2.1 Problem Statement

2.2 Empathy Map Canvas

2.3 Brainstorming

3. REQUIREMENT ANALYSIS

3.1 Customer Journey map

3.2 Solution Requirement

3.3 Data Flow Diagram

3.4 Technology Stack

4. **PROJECT DESIGN**

4.1 Problem Solution Fit
4.2 Proposed Solution
4.3 Solution Architecture

5. **PROJECT PLANNING & SCHEDULING**

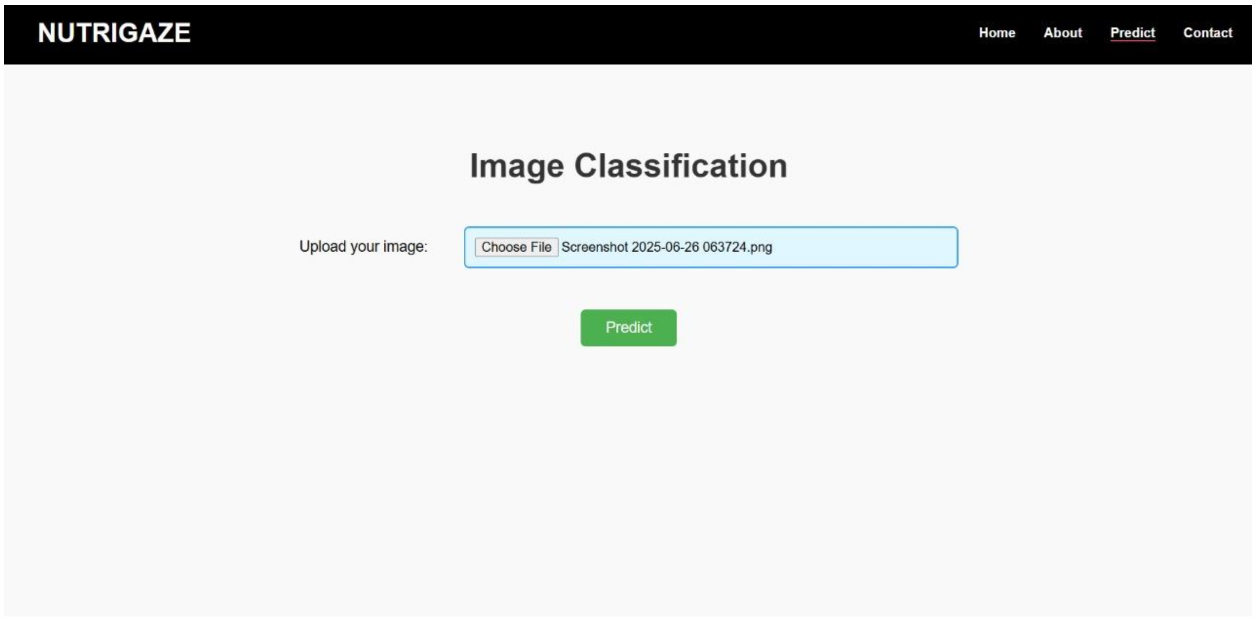
5.1 Project Planning

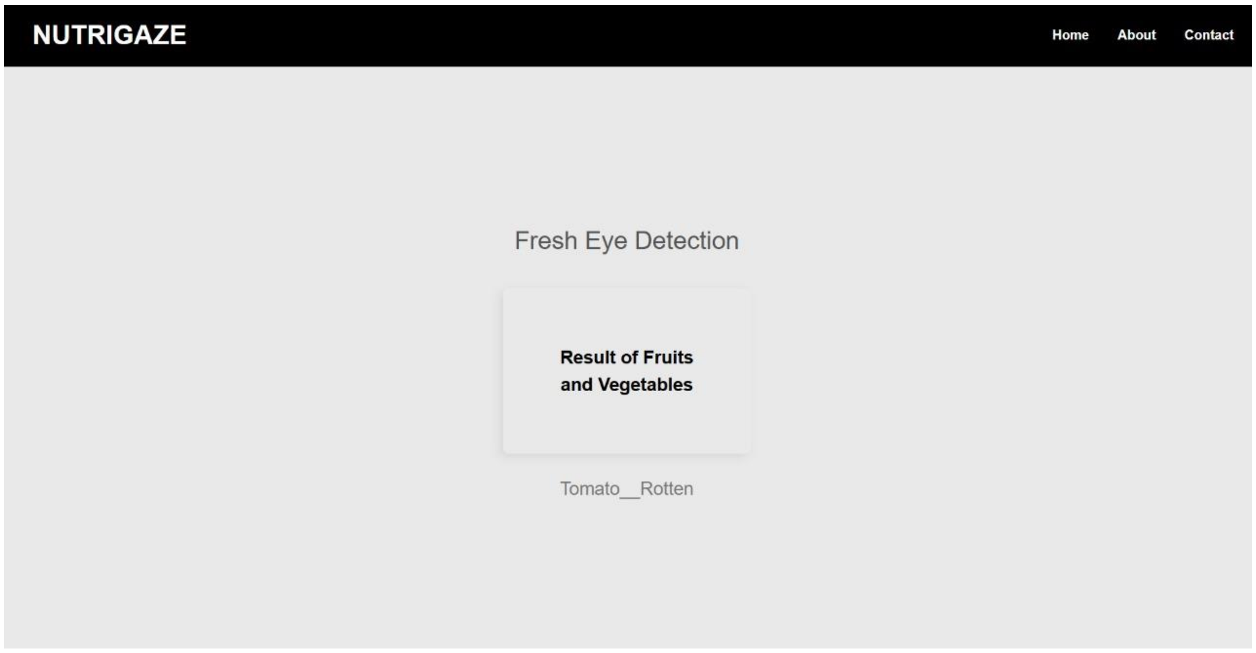
6. **FUNCTIONAL AND PERFORMANCE TESTING**

6.1 Performance Testing

7. **RESULTS**

7.1 Output Screenshots





Fresh Eye Detection

Result of Fruits and Vegetables

Tomato__Rotten

8. ADVANTAGES & DISADVANTAGES

Advantages:

1. **Efficient Quality Control:** Automates the process of sorting fruits and vegetables, reducing human error and ensuring consistent quality.
2. **Time and Labor Saving:** Minimizes manual effort and speeds up the sorting process, increasing productivity in agricultural supply chains.

Disadvantages:

1. **High Initial Setup Cost:** Requires good-quality cameras, computing infrastructure, and trained models, which can be costly to implement initially.
2. **Data Dependency:** Performance heavily relies on a large and diverse dataset; poor data quality can reduce accuracy.

9. CONCLUSION

The Smart Sorting project effectively demonstrates how transfer learning can be applied to automate the identification of rotten fruits and vegetables. By utilizing pre-trained models, the system achieves high accuracy in classification, ensuring reliable and efficient sorting.

This not only reduces manual labour but also improves the overall quality control process in agricultural and food retail sectors. With its strong performance and potential for scalability, the solution paves the way for real-world implementation, contributing to reduced food waste and enhanced productivity in supply chains.

10. FUTURE SCOPE

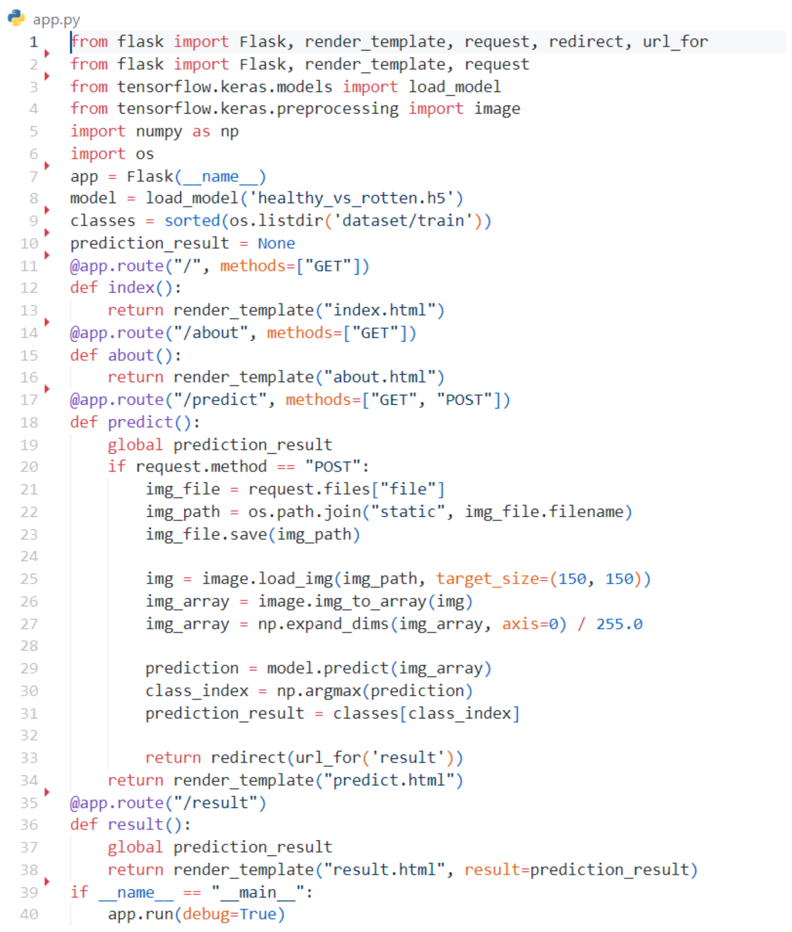
In the future, the Smart Sorting system can be improved to work in real-time with sorting machines in warehouses and supermarkets. By connecting it to cameras and robotic arms, the system can automatically detect and separate rotten fruits and vegetables quickly and accurately. It can also be trained to recognize different levels of freshness or signs of diseases, helping in better quality control.

This project can also be developed as a mobile app where farmers or vendors use their phone cameras to check the quality of produce instantly. With cloud support, data can be stored and used to improve the model over time. Expanding the model to support more types of fruits and vegetables will make it more useful in different regions and seasons, helping reduce food waste and improve food quality.

11. APPENDIX

Source Codes:

app.py



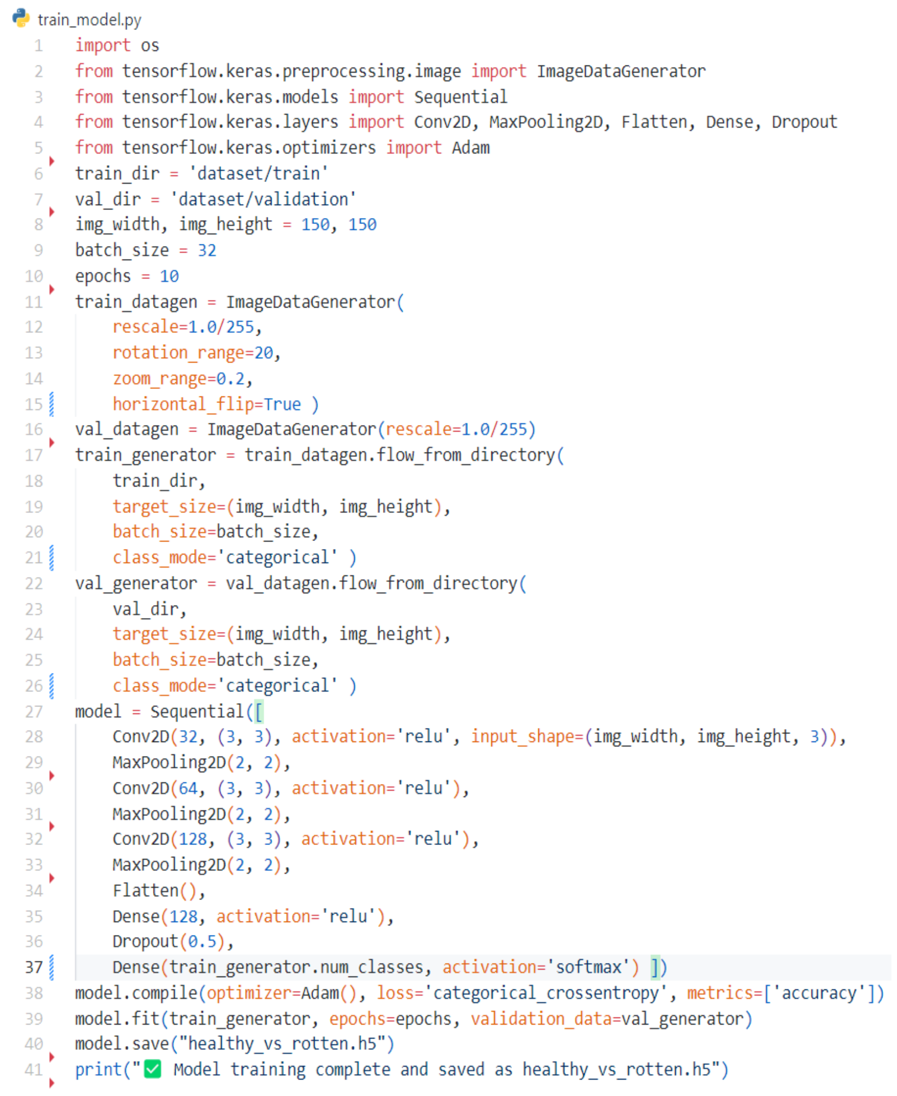
```python
from flask import Flask, render_template, request, redirect, url_for
from flask import Flask, render_template, request
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
import os
app = Flask(__name__)
model = load_model('healthy_vs_rotten.h5')
classes = sorted(os.listdir('dataset/train'))
prediction_result = None
@app.route("/", methods=["GET"])
def index():
    return render_template("index.html")
@app.route("/about", methods=["GET"])
def about():
    return render_template("about.html")
@app.route("/predict", methods=["GET", "POST"])
def predict():
    global prediction_result
    if request.method == "POST":
        img_file = request.files["file"]
        img_path = os.path.join("static", img_file.filename)
        img_file.save(img_path)

        img = image.load_img(img_path, target_size=(150, 150))
        img_array = image.img_to_array(img)
        img_array = np.expand_dims(img_array, axis=0) / 255.0

        prediction = model.predict(img_array)
        class_index = np.argmax(prediction)
        prediction_result = classes[class_index]

        return redirect(url_for('result'))
    return render_template("predict.html")
@app.route("/result")
def result():
    global prediction_result
    return render_template("result.html", result=prediction_result)
if __name__ == "__main__":
    app.run(debug=True)
```

train_model.py



train_model.py

```python
import os
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.optimizers import Adam
train_dir = 'dataset/train'
val_dir = 'dataset/validation'
img_width, img_height = 150, 150
batch_size = 32
epochs = 10
train_datagen = ImageDataGenerator(
    rescale=1.0/255,
    rotation_range=20,
    zoom_range=0.2,
    horizontal_flip=True )
val_datagen = ImageDataGenerator(rescale=1.0/255)
train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='categorical' )
val_generator = val_datagen.flow_from_directory(
    val_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='categorical' )
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(img_width, img_height, 3)),
    MaxPooling2D(2, 2),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(train_generator.num_classes, activation='softmax') ])
model.compile(optimizer=Adam(), loss='categorical_crossentropy', metrics=['accuracy'])
model.fit(train_generator, epochs=epochs, validation_data=val_generator)
model.save("healthy_vs_rotten.h5")
print("✅ Model training complete and saved as healthy_vs_rotten.h5")
```

Dataset Link:

Fruit and Vegetable Diseases Dataset
Source: [Kaggle] ([https://www.kaggle.com/datasets/muhammad0subhan/fruit-and-vegetable-disease-healthy-vs-rotten])

GitHub & Project Demo Link:

GitHub:

https://github.com/HaraKusuma/smart_sort_project

Project Demo Link:

https://drive.google.com/file/d/1EdyTN7qSOf0lnrPTPA6VxoklnPtevYBZ/view?usp=drivesdk