Project Documentation: Healthy vs Rotten Fruit Classification Web App

# 1. Project Overview

This web application enables users to upload images of fruits and receive predictions on whether the fruit is healthy or rotten. The application is built using Flask and incorporates a pre-trained machine learning model developed with Keras/TensorFlow.

# 2. Directory Structure

.  
├── static/  
│ ├── assets/ # Frontend static resources (CSS, JS, images)  
│ ├── forms/ # Optional static forms  
│ └── uploads/ # Stores uploaded fruit images for classification  
│  
├── templates/  
│ ├── blog-single.html # Blog single post template  
│ ├── blog.html # Blog overview template  
│ ├── index.html # Main homepage with upload form  
│ └── portfolio-details.html # Extra details or result display  
│  
├── app.py # Flask application script  
├── healthy\_vs\_rotten.h5 # Trained deep learning model (Keras format)  
├── ipython.html # Optional notebook export/reference  
├── Readme.txt # Basic project information

# 3. File Descriptions

- app.py: Core of the Flask application. Handles routing, file uploads, model loading, and result rendering.  
- healthy\_vs\_rotten.h5: Keras/TensorFlow model for binary classification (Healthy vs Rotten).  
- templates/: Contains HTML templates for rendering web pages.  
- static/: Hosts all static content including uploaded images.  
- Readme.txt: Brief information about the project.

# 4. Functional Workflow

1. User lands on the index.html page.  
2. Uploads an image of a fruit.  
3. The image is saved in the static/uploads/ directory.  
4. The image is preprocessed and fed to the model.  
5. The model outputs a prediction (healthy or rotten).  
6. Result is shown on a webpage using a template.

# 5. Requirements

pip install flask tensorflow pillow

# 6. Running the App

python app.py  
Navigate to: http://127.0.0.1:5000/

# 7. Future Enhancements

- Add frontend validation and error messages  
- Use advanced UI features and image processing  
- Extend model to classify fruit types

# 8. Credits

Developed as a mini-project using Python, Flask, TensorFlow, and HTML/CSS.