|  |
| --- |
| Mohamed Hesham  Fall 2025 |

|  |
| --- |
| Misr International University |
| Fruits Recognition |
| Mohamed Hesham – 2023/00428 |

A plate of fruit on a black background

AI-generated content may be incorrect.

# Introduction

Automated fruit recognition represents a significant challenge in computer vision with practical applications spanning automated retail systems, agricultural quality assessment, and dietary monitoring solutions. This project implements a comprehensive fruit identification system leveraging image processing techniques combined with machine learning algorithms. The primary objectives include effective image preprocessing, meaningful feature extraction, robust model training, and deployment of a user-friendly application capable of real-time fruit classification.

# Technical Methodology

## Dataset

The project utilizes the **Fruits-360 dataset** available on Kaggle:

* <https://www.kaggle.com/datasets/moltean/fruits>

## Dataset Overview

The dataset consists of fruit images categorized into distinct classes, organized within separate directories for **training** and **testing** purposes. This structured organization facilitates systematic model development and evaluation. The dataset was selected for its diversity, consistency, and well-defined categorization, making it ideal for developing and validating the recognition model.

A red pepper with green stem

AI-generated content may be incorrect.

# Pipeline Diagram

The implementation follows a structured processing pipeline as illustrated below:

**A close-up of a black background

AI-generated content may be incorrect.**

# Implementation Details

## Image Acquisition & Loading

A screen shot of a computer program

AI-generated content may be incorrect.

The dataset structure consists of training and testing directories, each containing class-specific subfolders. System validation confirms directory existence before processing.

Directory Structure:

* + Training path: /dataset/fruits-360/Training
  + Testing path: /dataset/fruits-360/Test

Custom functions were developed to load images and corresponding labels, converting images into NumPy arrays for computational processing. A dedicated **Fruit** class was implemented to encapsulate image data and labels, facilitating organized data management

## Color Space Conversion (HSV)

Images undergo conversion from BGR (OpenCV default) to **HSV** color space. This transformation facilitates more effective segmentation by separating color information from intensity.

HSV Components:

1. **Hue**: Dominant color (0-360° circular scale)
2. **Saturation**: Color purity/intensity
3. **Value**: Brightness component

### Conversion Implementation & Visualization

A screen shot of a computer

AI-generated content may be incorrect.

A close up of a fruit

AI-generated content may be incorrect.A graph with a purple line

AI-generated content may be incorrect.

## 3. Image Segmentation (Thresholding)

Region-based segmentation that isolates fruit regions from the background using **saturation** channel thresholding. The white backgrounds in the dataset have near-zero saturation values, while fruit regions show higher saturation, enabling effective separation.

**Threshold** value: 15 (chosen through trial & error), results in a binary mask where:

* fruit = white (255)
* background = black (0)

### Thresholding Implementation & Visualization:

A screen shot of a computer code

AI-generated content may be incorrect.

A comparison of a white and black image

AI-generated content may be incorrect.

A comparison of a white circle

AI-generated content may be incorrect.

A comparison of a moon and a binary moon

AI-generated content may be incorrect.

## Image Description & Feature Extraction

Four distinctive features were extracted from segmented fruit regions:

1. **Area**: Pixel count of fruit region (region-based descriptor)
2. **Average** **Hue**: Dominant color using circular mean calculation
3. **Average** **Saturation**: Mean color purity
4. **Average** **Value**: Mean brightness intensity.

### Feature Extraction Implementation

A screen shot of a computer program

AI-generated content may be incorrect.

### Process All Fruits (Segmentation & Feature Extraction)

A screenshot of a computer program

AI-generated content may be incorrect.A screenshot of a graph

AI-generated content may be incorrect.

## Data Preparation and Preprocessing

Extracted features undergo **standardization** using Min-Max scaling to ensure uniform value ranges (0-1), optimizing them for machine learning algorithms. Label **encoding** transforms categorical fruit names into numerical representations.

Preprocessing Steps:

1. Feature normalization using **MinMaxScaler**
2. Label encoding using **LabelEncoder**
3. Data splitting into training and testing sets.

### Preprocessing Implementation

A screen shot of a computer program

AI-generated content may be incorrect.

### Preprocessing Results (Before vs. After)

A screenshot of a computer

AI-generated content may be incorrect.

## Object Recognition & Classification

### Model Training and Selection

Multiple machine learning algorithms were evaluated:

* K-Nearest Neighbors (**KNN**): Implemented with Euclidean distance metric
* **Random Forest**: Ensemble method with multiple decision trees

### Training & Testing Results

Models were trained on normalized feature vectors with encoded labels, employing separate training and testing datasets.

**Random Forest** achieved **92.04%** accuracy on the **testing** data. Thus, the model `*fruit\_model.joblib*` is saved with its scaler & encoder for deployment (website implementation).

### Visualization Of Predictions

A group of fruit on a white background

AI-generated content may be incorrect.

### Confusion Matrix Visualization & Implementation

A screen shot of a computer program

AI-generated content may be incorrect.

A graph of a training

AI-generated content may be incorrect.

# Web Application Deployment

### A screenshot of a computer AI-generated content may be incorrect.Flask-Based Implementation

The trained model was integrated into a **Flask** web application providing real-time fruit recognition capabilities.

Application Features:

* User-friendly image upload interface
* Real-time prediction with confidence scoring
* Responsive design for various devices
* Error handling for invalid inputs

### **Home Section**

A screenshot of a website

AI-generated content may be incorrect.

### **Visit & Project Sections**

A screenshot of a fruit chart

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

### **Prediction Section (Try it yourself!)**

A screenshot of a fruit

AI-generated content may be incorrect.