  
Mini Project Report

Machine Learning

Field : Ingénierie des Systèmes d’Information

|  |
| --- |
| 🍎 Object Recognition Using Fruits 360 Dataset |

Written by:

Imane Himmi & Lamyae Rjafallah

## 1. Project Objective

The primary goal of this mini-project is to build a model that can automatically recognize and classify images of fruits. Using a machine learning approach, we trained and evaluated two different models to determine how accurately they can identify various types of fruits from the **Fruits 360** image dataset.

This project serves as a practical demonstration of visual object recognition using traditional machine learning techniques, with potential applications in fields like:

* Automated food sorting,
* Inventory management,
* Retail checkout systems,
* Computer vision in robotics.

## 2. Dataset Source

We used the **Fruits 360** dataset, a publicly available dataset tailored for image classification tasks. It contains over **90,000 images** of fruits, taken in a clean and consistent background setting.

### Key Characteristics:

* Image size : **100x100 pixels**
* Format : **RGB** (3 color channels)
* Categories: Over **130 fruit types**
* Folder-based labeling: Each fruit type is stored in a dedicated folder (e.g., Banana, Apple Braeburn), making labeling straightforward.

## 3. Exploratory Data Analysis (EDA)

During initial exploration, we observed the following:

* All images are **centered and cropped**, which simplifies preprocessing.
* The dataset is **balanced**, with roughly equal numbers of images per class.
* Image backgrounds and lighting are **standardized**, reducing visual noise.
* Some classes (e.g., different varieties of apples) are **visually similar**, which can challenge classification models.

## 4. Preprocessing Steps

Before feeding images to the machine learning models, we applied the following preprocessing:

### a. Image Flattening

Each 100x100x3 image was flattened into a **1D vector of 30,000 features**, allowing traditional ML models to process the pixel data numerically.

### b. Normalization

Pixel values (ranging from 0–255) were **scaled between 0 and 1**, standardizing the input range.

### c. Label Encoding

Fruit labels (e.g., "Apple Granny Smith") were **numerically encoded** so that models could interpret them as categorical classes.

### Visualization Insight (Intuition):

Think of each image as a point in a 30,000-dimensional space. Fruits of the same type (e.g., different bananas) cluster close together, while visually different fruits are spaced farther apart.

## 5. Algorithms Used

We applied **two models** to classify the fruit images:

### 1. K-Nearest Neighbors (KNN)

KNN is a **lazy learning algorithm** that classifies an image by comparing it to other images in the training set. It does not try to understand the data ahead of time. Instead, when it sees a new image, it checks which other images are closest to it (based on pixel values), and assigns it the most common label among those "neighbors".

* It’s like recognizing a fruit by looking around and finding the closest matching fruits you already know.

### 2. Ridge Classifier (Linear Regression for Classification)

Ridge Classifier is based on **linear regression**, where the algorithm tries to draw a line (or plane) to separate classes. It assumes that the relationship between image pixels and fruit labels is linear which often is not true for image data.

It is like trying to draw a straight line to separate apples and oranges on a graph, which works only if the fruits look very different.

## 6. Results Summary

| **Metric** | **KNN** | **RidgeClassifier** |
| --- | --- | --- |
| Accuracy | ✅ 0.9952 (~99.5%) | ⚠ 0.7103 (~71%) |
| Macro F1-score | ✅ 1.00 | ⚠ 0.70 |
| Weighted F1-score | ✅ 1.00 | ⚠ 0.70 |
| Class Consistency | Very Homogeneous | Uneven; some classes weak |
| Training Time | ⚠ Long | ✅ Fast |
| Generalization Ability | Good, but expensive | Moderate to weak |

## 7. Interpretation of Results

### KNN:

* **Excellent performance**: The model correctly classified nearly all fruit images.
* **Perfect F1-score**: The precision and recall were very close to 1.00, meaning almost every fruit was classified without error.
* **High computational cost**: Because KNN compares each new image to all training images; it takes longer, especially as the dataset grows.

### Ridge Classifier:

* Lower performance**: The model made many mistakes on fruits that look alike.**
* Poor generalization**: Struggled with complex image patterns, as it assumes simple linear boundaries between classes.**
* Faster**: Great for testing but not for production in image classification.**

## 8. Conclusion

This project shows that **not all machine learning algorithms perform equally well on image data**. The **K-Nearest Neighbors algorithm significantly outperformed** Ridge Classifier on the Fruits 360 dataset because it does not rely on strict assumptions about data structure. It simply finds the most similar images in memory.

However, KNN's performance comes with a cost: **slow prediction times and high memory usage**. For real-time applications or larger datasets, more scalable models like **Convolutional Neural Networks (CNNs)** might be a better fit.

Still, for the scope of this mini-project, KNN proved to be a powerful and intuitive tool for **fruit image classification**.