CMPE 462 - Machine Learning: Assignment 1

Asaf Kanlipicak, Ali Sonmez, Saur

November 28, 2025

1 Introduction

This report outlines our work on developing a multi-class classification model for fruit and vegetables. For the project, we needed to create a custom dataset consisting of five distinct classes: banana, carrot, cucumber, mandarin, and tomato. Two primary goals of the project was to understand the mechanics of classification algorithms by building one from scratch and facing with the difficulties of data collection.

In the following sections, we describe how we collected and processed our data (combining images, text, and numerical attributes), detail our custom implementation, and compare its performance against the standard Scikit-learn library.

2 Dataset

Examples

Our dataset includes **5922** images in total. We have **301** real photographs and **5621** generated images.

Table 1: Distribution of Generated and Photograph Images per Class

Class	Generated	Photograph	Total
Banana	1000	100	1100
Carrot	1173	50	1223
Cucumber	1000	50	1050
Mandarin	1000	51	1051
Tomato	1448	50	1498
Total	**5621**	**301**	**5922**

We have separated 0.8 of the generated images as our **training** set and 0.2 of it as our **validation** set. To see if a model which is trained on generated images can predict real photographs, we have selected all the photographs as our **test** set. Below are a few examples.

Figure 1: A generated banana

Table 2: Example Feature Vector (Class: Banana)

Feature Group	Attributes	Values
Grayscale Histogram	gray_000 . . . gray_005 . . . gray_063	2.0, 2.0, 13.0, 76.0, 77.0, 51.0 . . . . . . 205.0
Color Statistics	blue_mean, blue_std green_mean, green_std red_mean, red_std	73.93, 67.23 133.87, 84.41 163.57, 100.53
Physical Attributes	weight, size	130.93 g, 17.86 cm
Text Description	description	"soft yellow"
Target Label	**class**	**banana**

Figure 2: A generated carrot

Table 3: Example Feature Vector (Class: Carrot)

Feature Group	Attributes	Values
Grayscale Histogram	gray_000 ... gray_005	221.0, 217.0, 208.0, 193.0, 206.0, 202.0 ...
	... gray_063	... 172.0
Color Statistics	blue_mean, blue_std green_mean, green_std red_mean, red_std	172.12, 57.35 178.13, 48.76 190.24, 35.87
Physical Attributes	weight, size	46.06 g, 11.84 cm
Text Description	description	"orange temperate"
Target Label	**class**	**carrot**
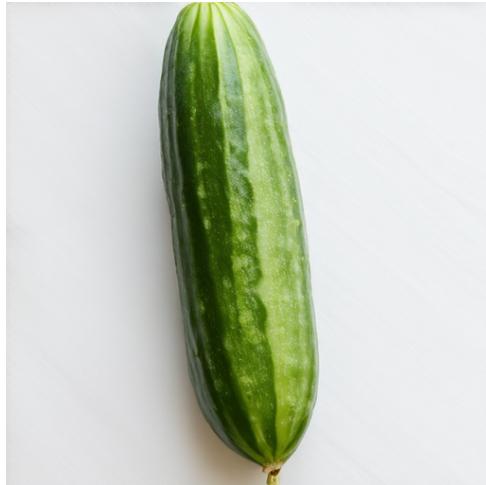
Figure 3: A generated cucumber

Table 4: Example Feature Vector (Class: Cucumber)

Feature Group	Attributes	Values
Grayscale Histogram	gray_000 ... gray_005	175.0, 168.0, 163.0, 159.0, 160.0, 154.0 ...
	... gray_063	... 190.0
Color Statistics	blue_mean, blue_std green_mean, green_std red_mean, red_std	149.94, 82.32 173.52, 66.73 175.36, 70.31
Physical Attributes	weight, size	287.30 g, 25.95 cm
Text Description	description	"temperate long"
Target Label	**class**	**cucumber**

Figure 4: A generated mandarin

Table 5: Example Feature Vector (Class: Mandarin)

Feature Group	Attributes	Values
Grayscale Histogram	gray_000 ... gray_005	186.0, 204.0, 225.0, 234.0, 230.0, 200.0 ...
	... gray_063	... 110.0
Color Statistics	blue_mean, blue_std	164.11, 85.86
	green_mean, green_std	190.25, 53.10
	red_mean, red_std	212.07, 44.73
Physical Attributes	weight, size	76.19 g, 7.87 cm
Text Description	description	"sour soft"
Target Label	**class**	**mandarin**

Figure 5: A generated tomato

Table 6: Example Feature Vector (Class: Tomato)

Feature Group	Attributes	Values
Grayscale Histogram	gray_000 . . . gray_005	135.0, 144.0, 122.0, 147.0, 218.0, 159.0 . . .
	. . . gray_063	. . . 50.0
Color Statistics	blue_mean, blue_std green_mean, green_std red_mean, red_std	100.99, 72.33 112.19, 72.33 130.97, 63.93
Physical Attributes	weight, size	108.11 g, 4.12 cm
Text Description	description	"soft sour"
Target Label	**class**	**tomato**

Data Collection

To obtain a large dataset we employed two techniques:

- **Generation:** The majority of the data was generated using **Stability AI's Stable Diffusion 3.5 Medium** model. Our prompts are visible in the generation notebook.

- **Manual Photography:** Additionally, we manually took photographs of the actual fruits and vegetables. These are used as the test set.

Preprocessing Pipeline

We wrote a preprocessing script (`image_processing.ipynb`) to standardize the raw inputs before feature extraction. Our pipeline is:

1. **Resizing:** All images, regardless of their original resolution, are resized to a uniform dimension of 512×512 **pixels**.

2. **Randomization:** A fixed seed (`SEED = 462`) was used to shuffle the dataset for reproducability concerns.

Dataset Splitting Strategy

We separated the data as follows:

- **Training & Validation Sets:** Derived from the **generated** images. The generated data was shuffled and split with an **80/20 ratio**:

    - **80%** allocated to the Training set.

    - **20%** allocated to the Validation set.

- **Test Set:** Composed only of the **real-world photographs**.

With the help of this splitting, we can test an interesting case in which a model is trained using generated images, but tested on the real ones.

Feature Extraction

We have **73** features in total, 64 of which is the grayscale representation of the image. The other 8 is related to colors, physical attributes, and semantic information.

Grayscale Histogram ($8 \times 8 = 64$ features)

We convert each image into a 8×8 grayscale image and give the value of each pixel as a feature to our model. This helps distinguishing between objects with similar colors but different surface textures (e.g., the smooth surface of a tomato vs. the rougher texture of a carrot).

How did we extract: We made use of OpenCV to convert the image into grayscale and resize it to 8×8.

Color Statistics (RGB Mean and Standard Deviation)

Color is the most intuitive discriminator for this specific dataset.

- **Mean:** Captures the dominant color of the object. This is crucial to separate the objects with different colors such as banana and mandarin.

- **Standard Deviation:** Captures the color variance. Even though we are not sure, we think that standard deviation might help the model learn some texture specific relation. (On the surface of a cucumber, there might be frequent dark spots.)

How did we extract: We made use of OpenCV to read the color value of each pixel in an image. Afterwards, it was just simple math to calculate the features.

Physical Attributes (Weight and Size)

Weight and size are important features since these help the model to learn, hopefully, the geometric attributes of an object.

- These features define the geometric and physical boundaries of the classes (A cucumber is significantly longer/larger than a mandarin).

How did we extract: This might be the trickiest one to extract. We used Gaussian distributions with different mean and variance values for each class to generate weights and sizes. For example, the average weight of a banana is 120 g, with a variance of 16 g² in the code. These values were derived by asking to ChatGPT since there are multiple answers on the Internet, but we want one.

Semantic Features (Text Description)

We have selected a set of words to define different classes. However, we paid attention not to make them easily separable. For example, if an object can have the word "tropical" in its text feature, then, another one also can.

- We aim to make the model learn the relation between a *bag of words* and a class. With combination of words, it is feasible to separate the objects that share some common words.

How did we extract: We created a vocabulary in which we aggregated all unique descriptors (e.g., colors, textures, climates) across all classes. We then represented the text description of each sample as a binary feature vector, where each dimension corresponds to the presence (1) or absence (0) of a specific word from this vocabulary. This resulted in a representation whcih encodes the attributes defined in our vocabulary.

To integrate the multi-modal data, we concatenated the normalized feature vectors from the visual, physical, and semantic domains into a single unified vector (in a csv file).