Assignment 1: Fruit and Vegetable Recognition

CMPE 462 – Machine Learning

Authors: Salih Can Erer (2022400174)
Beyza Nur Deniz (2021400285)

Institution: Boğaziçi University
Department of Computer Engineering

Date: November 2025

Contents

1 Introduction 3

1.1 Problem Definition . . . 3

1.2 Contributions . . . 3

2 Dataset Description 4

2.1 Semantic Class Definitions . . . 4

2.2 Dataset Statistics and Splits . . . 4

2.3 Representative Samples . . . 4

3 Data Collection and Preprocessing 6

3.1 Real-World Data Acquisition . . . 6

3.2 Synthetic Sample Generation . . . 6

3.2.1 Synthetic Image Generation . . . 6

3.2.2 Synthetic Text Generation . . . 7

3.2.3 Synthetic Categorical Attribute Generation . . . 8

3.3 Data Cleaning and Normalization . . . 8

3.3.1 Image Preprocessing . . . 8

3.3.2 Text Preprocessing . . . 9

3.3.3 Categorical Preprocessing . . . 9

4 Feature Engineering and Representation Learning 9

4.1 Image Feature Extraction . . . 9

4.2 Text Feature Extraction . . . 10

4.3 Categorical Encoding . . . 10

4.4 Image Feature Dimensionality Reduction via PCA . . . 10

5 Multimodal Feature Fusion and Normalization 11

5.1 Feature Vector Representation . . . 11

5.2 Feature Normalization . . . 11

5.3 Final Feature Dimensions . . . 11

6 Similarity Analysis, Dataset Difficulty, and Failure Modes 12

6.1 Intra-Class Similarity . . . 12

6.2 Inter-Class Similarity . . . 12

6.3 Distance-Based and Statistical Outlier Detection . . . 13

6.4 Quantitative Similarity Results . . . 13

6.5 Dataset Difficulty Interpretation . . . 13

6.6 Ambiguity Sources and Failure Modes . . . 14

7 Logistic Regression Classifier 14

7.1 Binary Logistic Model . . . 14

7.2 Probabilistic Formulation and Regularization . . . 14

7.3 One-vs-All Multiclass Extension . . . 15

7.4 Implementation Properties . . . 15

7.5 Training Algorithm . . . 16

8 Experimental Protocol 16

8.1 Training Hyperparameters . . . 16

8.2 Evaluation Metrics . . . 17

8.3 Random Seed Control and Reproducibility . . . 17

9 Results and Performance Evaluation 18

9.1 Single-Modality Performance . . . 18

9.2 Multimodal Fusion Performance . . . 18

9.3 Training Dynamics and Convergence Behavior . . . 19

9.4 Comparison with Scikit-Learn Implementation . . . 19

10 Conclusion 20

Abstract

We present a multimodal fruit and vegetable recognition system based on image, text, categorical, and numerical features. Handcrafted image descriptors, TF–IDF text representations, and encoded tabular attributes are combined through early feature fusion. A regularized logistic regression classifier is implemented from scratch using a one-vs-all strategy and compared with the `scikit-learn` implementation. Intra-class and inter-class similarity analyses, together with distance-based outlier detection, are used to evaluate dataset structure and difficulty. Experimental results show that multimodal fusion outperforms single-modality classifiers.

Code and Data Availability: The complete dataset and source code implementation are available here.

1 Introduction

1.1 Problem Definition

Multimodal learning aims to exploit complementary information from multiple data sources such as images, text, and structured attributes. Fruit and vegetable recognition provides a natural multimodal classification problem, where visual appearance alone is often insufficient to fully distinguish between semantically similar classes. Additional textual descriptions and tabular measurements can significantly improve discriminative power.

In this study, we construct a multimodal dataset consisting of five fruit and vegetable classes and extract handcrafted image, text, categorical, and numerical features under the constraints of the assignment. A regularized logistic regression classifier is implemented from scratch and extended to the multiclass setting via a one-vs-all strategy. Separate models are trained for each modality as well as for their fused representation. In addition to standard classification metrics, we analyze intra-class and inter-class similarities and perform distance-based outlier detection in order to assess dataset difficulty and class overlap.

1.2 Contributions

Scientific Insight

- Construction of a multimodal fruit and vegetable dataset with image, text, categorical, and numerical attributes.
- Scratch implementation of regularized multiclass logistic regression.
- Empirical comparison of single-modality and multimodal fused classifiers.
- Similarity analysis and outlier detection for dataset difficulty assessment.

2 Dataset Description

2.1 Semantic Class Definitions

The dataset contains five fruit and vegetable categories: `apple`, `banana`, `cucumber`, `eggplant`, and `tangerine`. These classes were chosen to include both visually similar and distinct categories, ensuring that the classification problem is neither trivial nor dominated by noise.

2.2 Dataset Statistics and Splits

The dataset consists of exactly 3,092 labeled samples. The per-class counts, extracted directly from the dataset directory, are presented below:

Class	Samples
apple	656
banana	614
cucumber	608
eggplant	608
tangerine	606
Total	**3092**

The splitting procedure uses two stratified splits with fixed `random_state=42`. The notebook reports the following exact split sizes:

- **Training set:** 1854 samples
- **Validation set:** 619 samples
- **Test set:** 619 samples

Methodological Insight

Data Integrity Protocol: No resampling, duplication, or synthetic balancing was applied after splitting. Furthermore, strictly no missing values occur in any modality, ensuring a clean baseline for model evaluation.

2.3 Representative Samples

Figures 1, 2, and 3 present representative examples of rejected AI-generated samples, accepted AI-generated samples, and manually collected real images, respectively.

The rejected samples illustrate typical failure patterns such as structural distortions and semantic inconsistencies, while the accepted AI-generated samples demonstrate high visual realism and class-consistent geometry. The real samples reflect natural variations in lighting, viewpoint, and background conditions. Together, these examples qualitatively summarize the visual diversity and quality control pipeline applied to the dataset.

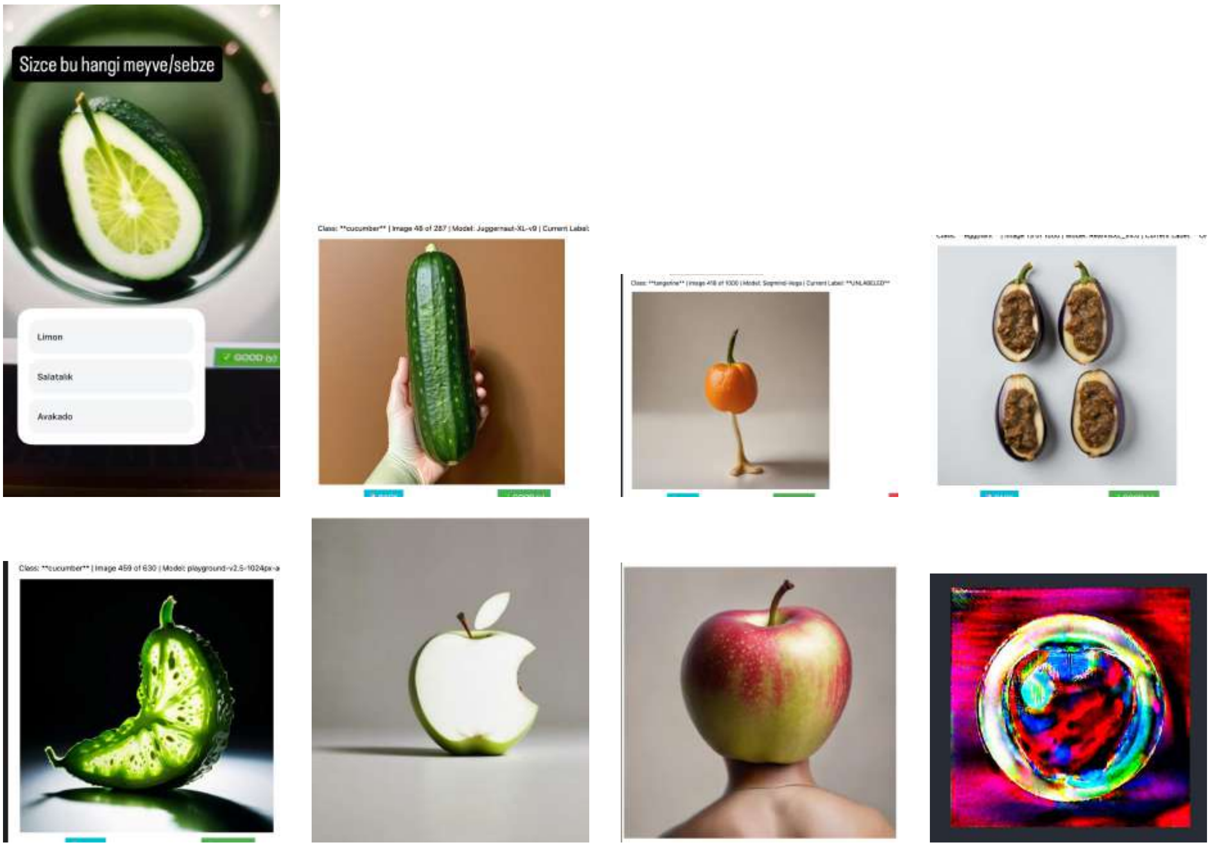

Sizce bu hangi meyve/sebze
Limon
Salatalık
Avakado
✓ GOOD
Class: **cucumber** | Image 48 of 287 | Model: Juggernaut-XL-v9 | Current Label:
Class: **tangerine** | Image 418 of 1000 | Model: Segmind-Vega | Current Label: **UNLABELED**
Class: **cucumber** | Image 459 of 630 | Model: playground-v2.5-1024px-a

Figure 1: Examples of **rejected AI-generated samples** labeled as *bad* during manual quality control. These images exhibit severe structural deformations, unrealistic compositions, and semantic ambiguity. All such samples were discarded and excluded from the final dataset.

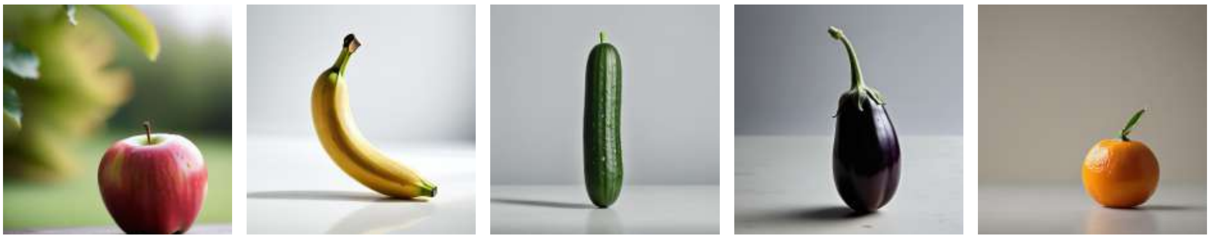


Figure 2: Examples of **accepted AI-generated samples** after manual quality filtering. These images exhibit realistic photometric properties, correct object geometry, and consistent semantics. Only samples passing this inspection were retained in the dataset.

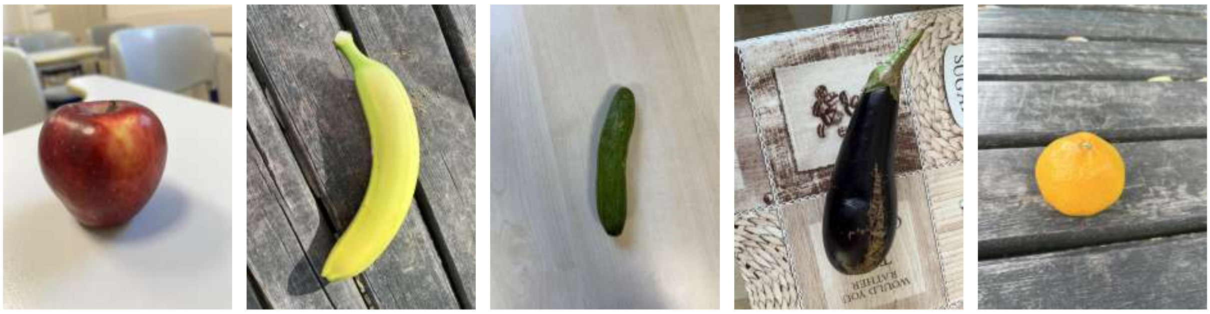


Figure 3: Examples of **manually collected real samples** captured under natural lighting and background conditions.

3 Data Collection and Preprocessing

3.1 Real-World Data Acquisition

The real-world image subset was constructed through manual collection using consumer-grade cameras under natural indoor and outdoor lighting conditions. We strictly adhered to the assignment requirement of collecting at least 50 samples per category personally. **Specifically, we collected between 50 and 60 unique real-world samples for each of the five classes**, ensuring a diverse baseline of authentic data. Images were captured with variations in background, viewpoint, distance, and object orientation. All real samples were manually assigned to their corresponding semantic classes.

3.2 Synthetic Sample Generation

Synthetic data were generated independently for the **image**, **text**, and **categorical** modalities. Each modality is described separately below.

3.2.1 Synthetic Image Generation

Synthetic images were generated using five high-capacity text-to-image diffusion models:

- `stabilityai/stable-diffusion-xl-base-1.0`
- `SG161222/RealVisXL_V4.0`
- `playgroundai/playground-v2.5-1024px-aesthetic`
- `segmind/Segmind-Vega`
- `RunDiffusion/Juggernaut-XL-v9`

Scientific Insight

Bias Mitigation Strategy: Using multiple architectures reduces model-specific bias and improves approximate sample independence across the synthetic distribution, preventing the classifier from overfitting to specific rendering artifacts of a single generator.

For each class, a total of **50 structured prompt templates** were defined and grouped into five semantic categories:

- **Viewpoint variations (10 prompts):** front view, overhead view, side angles, low-angle, rear view, tilted view, and multi-view compositions.
- **Distance and framing (10 prompts):** wide shots, medium distance, object scale variation, negative space control, and depth-of-field effects.
- **Lighting conditions (10 prompts):** studio lighting, natural daylight, low-key and high-key lighting, backlighting, and artificial warm/cool illumination.
- **Background variations (10 prompts):** plain color backgrounds, wood, marble, fabric, metal, paper, outdoor grass, and kitchen surfaces.
- **Contextual compositions (10 prompts):** kitchen scenes, produce baskets, cutting boards, realistic aging, harvest preparation, and isolated object settings.

Each prompt was instantiated for a specific class token and sampled **four times with independent random seeds**, yielding up to 200 images per class per model. To suppress multi-object failures and cluttered compositions, a fixed **negative prompt** was applied during all generations to explicitly forbid groups, piles, and multiple-object scenes.

All synthetic images were subsequently subjected to **manual visual inspection**. Outputs exhibiting structural deformations, incorrect object multiplicity, unrealistic geometry, or severe rendering artifacts were labeled as *bad* and permanently discarded. Only visually consistent and semantically valid samples were retained for use in the final dataset.

3.2.2 Synthetic Text Generation

Aligned textual descriptions were generated for every image in the final dataset using the large-scale vision–language model `Salesforce/InstructBLIP-Vicuna-7B`. Caption generation was executed in parallel across two GPUs using a dedicated multiprocessing pipeline. Each image was processed independently, and the resulting description was stored as a plain-text file preserving the directory hierarchy of the image dataset to ensure exact one-to-one modality alignment.

Methodological Insight

Label Leakage Prevention (Masking): To prevent explicit label leakage from the text modality, a masking procedure was applied to all generated captions prior to training. All occurrences of the class names (`apple`, `banana`, `cucumber`, `eggplant`, `tangerine`) as well as the frequently misused alias `orange` were removed using a regular-expression filter and replaced with the token `<mask>`. This explicitly prevents trivial class inference from direct lexical cues.

No manual intervention or post-editing was applied to the captions. Missing caption files were replaced with empty strings to preserve sample alignment across modalities. All lexical normalization, vocabulary filtering, and statistical feature extraction steps are deferred to the feature extraction stage.

3.2.3 Synthetic Categorical Attribute Generation

Each sample was assigned three categorical metadata fields: *origin region*, *climate type*, and *harvest period*. Attribute values were generated using a probabilistic class-aware sampling strategy designed to balance semantic consistency with controlled stochastic variation.

For a given class label, attributes were drawn from the corresponding class-specific pools with probability $p = 0.75$, ensuring that the majority of samples reflect realistic, semantically aligned metadata. With probability $1 - p$, attribute values were sampled uniformly from the global unions of all categories across classes. This introduces structured label noise, preventing the categorical modality from becoming perfectly class-deterministic while maintaining overall semantic coherence.

The global attribute pools contain 10 origin regions, 7 climate categories, and up to 7 harvest periods, yielding a broad combinatorial space. All categorical values were stored as raw string labels during dataset construction. No normalization or encoding was applied at this stage; numerical encoding via one-hot transformation is performed later during feature extraction to ensure consistent downstream representation.

3.3 Data Cleaning and Normalization

3.3.1 Image Preprocessing

All images were loaded from disk using OpenCV and uniformly resized to a fixed spatial resolution of 128×128 pixels to ensure consistent spatial dimensionality across the dataset. No photometric normalization, color standardization, data augmentation, or histogram equalization was applied at this stage in order to preserve the natural appearance of both real and synthetic samples. Any corrupted or unreadable image files encountered during loading were discarded. The resulting resized images constitute the direct input to the handcrafted visual feature extraction pipeline described in the subsequent section.

3.3.2 Text Preprocessing

Textual descriptions were loaded from disk using a directory structure strictly aligned with the image dataset to guarantee one-to-one modality correspondence. For samples with missing caption files, empty strings were substituted to preserve dataset indexing consistency. The masked captions were used in their raw form without additional token normalization, stopword removal, or lemmatization at this stage. All statistical text processing steps, including TF–IDF vectorization and dimensionality reduction, are deferred to the feature extraction stage.

3.3.3 Categorical Preprocessing

Categorical attributes were stored as raw symbolic string labels during dataset construction. No encoding, scaling, or normalization was applied during preprocessing to avoid premature numerical bias. All categorical variables are later transformed using one-hot encoding during feature extraction, ensuring consistent and comparable numerical representations across samples.

4 Feature Engineering and Representation Learning

4.1 Image Feature Extraction

Let $x \in \mathbb{R}^{128\times128\times3}$ denote a resized RGB image. From each image, a handcrafted visual feature vector is constructed through the concatenation of five complementary descriptors:

Methodological Insight

Visual Descriptor Stack:

$$\phi_{\text{img}}(x) = [\phi_{\text{HOG}}(x),\ \phi_{\text{HSV}}(x),\ \phi_{\text{LBP}}(x),\ \phi_{\text{Hu}}(x),\ \phi_{\text{BoVW}}(x)] \in \mathbb{R}^{d_{\text{img}}}$$

This composition ensures capture of local gradients, global color distributions, texture patterns, and shape invariants simultaneously.

SIFT and Bag-of-Visual-Words. Scale-Invariant Feature Transform (SIFT) descriptors are extracted from each grayscale image. Given an image x, SIFT produces a set of local descriptors $D(x) = \{d_1, \dots, d_n\}$ where $d_i \in \mathbb{R}^{128}$. A visual vocabulary is learned using MiniBatch k-means clustering with $K = 200$ clusters on the training descriptors. Each image is mapped to a normalized histogram $\phi_{\text{BoVW}}(x) \in \mathbb{R}^K$ by assigning descriptors to nearest visual words.

Histogram of Oriented Gradients (HOG). Gradient-based shape information is extracted using HOG with 9 orientation bins, 8×8 pixel cells, and 2×2 cell blocks with L_2-Hys normalization. This produces a dense gradient descriptor $\phi_{\text{HOG}}(x) \in \mathbb{R}^{d_{\text{HOG}}}$.

HSV Color Histogram. Color distribution is encoded in the HSV color space using 32 bins for hue and 16 bins for saturation/value. The resulting vector $\phi_{\text{HSV}}(x) \in \mathbb{R}^{64}$ is normalized to unit mass.

Local Binary Patterns (LBP). Texture information is extracted using uniform LBP with radius $R = 1$ and $P = 8$ neighbors, yielding a normalized histogram $\phi_{\text{LBP}}(x) \in \mathbb{R}^{59}$.

Hu Moments. Global shape invariants are captured using seven Hu moments. A signed logarithmic transform is applied for numerical stability:

$$\phi_{\text{Hu}}(x)_i = -\operatorname{sign}(h_i)\log(|h_i| + \varepsilon).$$

4.2 Text Feature Extraction

Let t denote the masked textual description associated with an image. Text is embedded using TF–IDF with unigrams and bigrams (ngram range $= \{1, 2\}$). The vocabulary is limited to `TFIDF_MAX` $= 50$ most frequent terms after stop-word removal. Each document is mapped to a sparse vector $\phi_{\text{text}}(t) \in \mathbb{R}^{50}$.

4.3 Categorical Encoding

Each sample includes three categorical attributes: origin, climate, and harvest period. These are jointly encoded using one-hot encoding:

$$\phi_{\text{cat}} : \mathcal{C} \to \{0, 1\}^{d_{\text{cat}}}.$$

The encoder is fitted on the training split only to prevent data leakage.

4.4 Image Feature Dimensionality Reduction via PCA

The raw concatenated image feature vector $\phi_{\text{img}}(x) \in \mathbb{R}^{8430}$ results in a very high-dimensional representation due to the aggregation of dense HOG, HSV, and BoVW descriptors.

Important Limitation

Dimensionality Constraint: The raw dimensionality ($d = 8430$) significantly increases computational complexity and violates the assignment-imposed upper limit of 500 dimensions per modality.

To satisfy this constraint and suppress redundancy, Principal Component Analysis (PCA) is applied as a linear dimensionality reduction step. Let $X \in \mathbb{R}^{n \times 8430}$ denote the training feature matrix. PCA computes an orthogonal projection $Z = XW$, where $W \in \mathbb{R}^{8430 \times 200}$ contains the eigenvectors corresponding to the top-200 eigenvalues.

The target dimensionality is fixed to $\boxed{d_{\text{pca}} = 200}$, preserving dominant variance while comfortably satisfying the feature constraint. The projection is fitted exclusively on training data to maintain strict experimental integrity.

5 Multimodal Feature Fusion and Normalization

5.1 Feature Vector Representation

Let $\Omega = \{\text{img}, \text{txt}, \text{cat}\}$ be the set of available modalities. For a sample x, we define a projection operator $\pi_m(x)$ mapping the raw input to the feature space of modality $m \in \Omega$.

The fused representation $\mathbf{\Phi}(x)$ is constructed via the concatenation operator $\oplus$:

$$\mathbf{\Phi}(x) = \bigoplus_{m \in \Omega} \alpha_m \cdot \mathcal{N}_m(\pi_m(x)),$$

where $\mathcal{N}_m$ denotes the modality-specific normalization function (e.g., z-score or unit-norm) and α_m represents an implicit modality weight (set to 1.0 in our early fusion scheme).

5.2 Feature Normalization

Before fusion, strictly independent normalization is applied to ensure numerical compatibility across heterogeneous feature spaces and prevent any single modality from dominating the gradient updates.

Methodological Insight

Modality-Specific Scaling:

- **Image Features:** Standardized using z-score normalization ($\mu = 0, \sigma = 1$).
- **Text Features:** Variance-normalized *without centering* to preserve the sparsity structure of TF–IDF vectors.
- **Categorical Features:** Scaled *without mean centering* after one-hot encoding to maintain orthogonality.

5.3 Final Feature Dimensions

Following the PCA-based dimensionality reduction ($8430 \to 200$) applied to the image modality, the final component sizes are:

$$d_{\text{img}} = 200, \quad d_{\text{txt}} = 50, \quad d_{\text{cat}} = 64.$$

Key Experimental Result

Fused Vector Dimensionality: The resulting fused feature vector $\phi(x)$ passed to the classification models has a total dimension of:

$$\boxed{d = 200 + 50 + 64 = 314}$$

6 Similarity Analysis, Dataset Difficulty, and Failure Modes

This section quantitatively analyzes the geometric structure of the feature space in terms of class compactness, class separability, and the presence of statistical outliers. All similarity computations are performed using the cosine similarity metric. The analysis further provides a principled interpretation of dataset difficulty and dominant sources of classification failure.

6.1 Intra-Class Similarity

Let $X_c = \{\mathbf{x}_i \mid y_i = c\} \subset \mathbb{R}^d$ denote the set of all feature vectors belonging to class c. The intra-class similarity is defined as the average pairwise cosine similarity:

$$S_{\text{intra}}(c) = \frac{1}{|X_c|(|X_c| - 1)} \sum_{\substack{i,j \in X_c \\ i \neq j}} \frac{\mathbf{x}_i^\top \mathbf{x}_j}{\|\mathbf{x}_i\|_2 \|\mathbf{x}_j\|_2}.$$

High intra-class similarity implies strong geometric compactness within the class.

6.2 Inter-Class Similarity

For two distinct classes $c_1 \neq c_2$, the inter-class similarity is defined as:

$$S_{\text{inter}}(c_1, c_2) = \frac{1}{|X_{c_1}||X_{c_2}|} \sum_{i \in X_{c_1}} \sum_{j \in X_{c_2}} \frac{\mathbf{x}_i^\top \mathbf{x}_j}{\|\mathbf{x}_i\|_2 \|\mathbf{x}_j\|_2}.$$

Low inter-class similarity corresponds to strong class separability.

Proposition 1. *Inter-class cosine similarity decreases as class separability increases, following directly from the definition of angular distance in feature space.*

6.3 Distance-Based and Statistical Outlier Detection

For each class c, the centroid vector is defined as $\boldsymbol{\mu}_c = \frac{1}{|X_c|} \sum_{\mathbf{x}_i \in X_c} \mathbf{x}_i$.

Methodological Insight

Dual Outlier Criteria: A sample is flagged if it satisfies either of the following conditions:

1. **Distance-Based:** $\|\mathbf{x}_i - \boldsymbol{\mu}_c\|_2 > \mu_d + 2\sigma_d$
2. **Statistical (Z-score):** $\exists k$ s.t. $z_k = \frac{x_k - \mu_k}{\sigma_k} > 3$

The final outlier set is obtained by the union of both criteria.

6.4 Quantitative Similarity Results

Key Experimental Result

Separation Ratio Analysis ($\rho = S_{\text{inter}}/S_{\text{intra}}$):

- **Image Features:** $\overline{S}_{\text{intra}} = 0.466, \quad \overline{S}_{\text{inter}} = 0.458, \quad \boxed{\rho_{\text{img}} = 0.982}$
- **Text Features:** $\overline{S}_{\text{intra}} = 0.287, \quad \overline{S}_{\text{inter}} = 0.166, \quad \rho_{\text{txt}} = 0.578$
- **Categorical Features:** $\overline{S}_{\text{intra}} = 0.108, \quad \overline{S}_{\text{inter}} = 0.054, \quad \rho_{\text{cat}} = 0.505$
- **Fused Features:** $\overline{S}_{\text{intra}} = 0.465, \quad \overline{S}_{\text{inter}} = 0.456, \quad \boxed{\rho_{\text{fused}} = 0.981}$

6.5 Dataset Difficulty Interpretation

The overlap structure is characterized by the separation ratio ρ. Values of $\rho \approx 1$ indicate that class manifolds intersect or lie in extremely close angular proximity.

Scientific Insight

The Highly Overlapping Regime: Since $\rho_{\text{img}} \approx \rho_{\text{fused}} \approx 0.98$, the dataset formally lies in the **highly overlapping regime**. Even optimal linear classifiers operate near the margin of theoretical separability, making the task intrinsically sensitive to noise.

In contrast, the categorical and text modalities exhibit significantly lower ρ values (≈ 0.5), confirming that semantic supervision provides orthogonal separation axes not available in pixel space alone.

6.6 Ambiguity Sources and Failure Modes

Important Limitation

Dominant Failure Modes: The observed near-equality of intra- and inter-class similarities arises from specific physical ambiguities:

- Strong visual similarity between elongated classes (e.g., `banana` vs. `cucumber`).
- Overlapping color and texture distributions (e.g., `apple` vs. `tangerine` under warm lighting).
- Synthetic image artifacts causing geometric distortions that disrupt HOG descriptors.

7 Logistic Regression Classifier

We employ a fully custom implementation of multinomial logistic regression, trained via **One-vs-All (OvA)** decomposition with L2-regularized gradient descent. The classifier is implemented entirely from scratch in NumPy to ensure full transparency of the optimization process.

7.1 Binary Logistic Model

For a binary classification problem given a feature matrix $X \in \mathbb{R}^{n \times d}$ and labels $y \in \{0,1\}^n$, the model predicts probabilities via the sigmoid function:

$$p(y = 1 \mid \mathbf{x}) = \sigma(\mathbf{w}^\top \mathbf{x} + b), \quad \text{where} \quad \sigma(z) = \frac{1}{1 + e^{-z}}.$$

7.2 Probabilistic Formulation and Regularization

We approach the learning problem via Maximum A Posteriori (MAP) estimation. We assume the likelihood of the data follows a Bernoulli distribution and impose a zero-mean Gaussian prior on the weights to induce sparsity and bound the norm:

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \tau^2 I).$$

Maximizing the posterior $P(\mathbf{w} \mid \mathcal{D}) \propto P(\mathcal{D} \mid \mathbf{w})P(\mathbf{w})$ is equivalent to minimizing the negative log-posterior. This yields the regularized objective function:

$$\mathcal{J}(\mathbf{w}, b) = -\frac{1}{n} \sum_{i=1}^{n} [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)] + \frac{\lambda}{2} \|\mathbf{w}\|_2^2, \tag{1}$$

where the regularization strength λ is inversely proportional to the prior variance τ^2.

Remark 1. The analytical gradient with respect to the weights is derived via the chain rule as:

$$\nabla_{\mathbf{w}} \mathcal{J} = \frac{1}{n} X^\top (\hat{\mathbf{y}} - \mathbf{y}) + \lambda \mathbf{w}.$$

The term $\lambda \mathbf{w}$ (weight decay) arises directly from the derivative of the Gaussian prior log-likelihood.

7.3 One-vs-All Multiclass Extension

Let $\mathcal{C} = \{c_1, \ldots, c_K\}$ denote the set of class labels. We train K independent binary classifiers, yielding a weight matrix $W \in \mathbb{R}^{d \times K}$ and bias vector $\mathbf{b} \in \mathbb{R}^K$. At inference time, class probabilities are normalized using the softmax function:

$$P(y = c_k \mid \mathbf{x}) = \frac{\exp(\mathbf{w}_k^\top \mathbf{x} + b_k)}{\sum_{j=1}^{K} \exp(\mathbf{w}_j^\top \mathbf{x} + b_j)}.$$

The final prediction is obtained via $\hat{y} = \arg\max_k P(y = c_k \mid \mathbf{x})$.

7.4 Implementation Properties

The classifier supports:

- L2-regularized risk minimization,
- adaptive learning rate decay,
- full train/validation loss tracing,
- numerically stable sigmoid and softmax evaluation,
- scalable One-vs-All class decomposition.

This design enables fair comparison across image-only, text-only, categorical-only, and fully fused feature representations.

7.5 Training Algorithm

Algorithm 1 One-vs-All Logistic Regression Training

```
for each class $c_k \in \mathcal{C}$ do
    Initialize $\mathbf{w}_k \leftarrow \mathbf{0}$, $b_k \leftarrow 0$
    Set initial learning rate $\eta$
    while not converged do
        Compute probability $p = \sigma(X\mathbf{w}_k + b_k)$
        Compute gradients $\nabla_{\mathbf{w}}, \nabla_b$
        Update weights: $\mathbf{w}_k \leftarrow \mathbf{w}_k - \eta\nabla_{\mathbf{w}}$
        Update bias: $b_k \leftarrow b_k - \eta\nabla_b$
        if loss plateau then $\eta \leftarrow \eta/2$
    end while
end for
```

8 Experimental Protocol

This section describes the full training pipeline, hyperparameter selection strategy, evaluation metrics, and all mechanisms used to ensure reproducibility.

8.1 Training Hyperparameters

All experiments are conducted using a stratified **training–validation–test split** with fixed `random_state = 42`. Feature standardization, dimensionality reduction, and classifier training are performed strictly within the training split to avoid data leakage.

- **Feature Scaling:** All numerical image features are standardized ($\mu = 0, \sigma = 1$) via `StandardScaler`. Sparse features (Text/Categorical) are scaled without centering (`with_mean=False`).
- **PCA Reduction:** High-dimensional image features ($d = 8430$) are reduced to $d_{\text{img}} = 200$ using Randomized SVD to satisfy assignment constraints.

Final Feature Dimensions. After preprocessing, the component dimensions are $d_{\text{img}} = 200, d_{\text{text}} = 50, d_{\text{cat}} = 64$, yielding a total fused representation of:

$$\boxed{d_{\text{fusion}} = 314}$$

Methodological Insight

Adaptive Optimization Strategy: The learning rate and maximum iterations are selected adaptively based on feature dimensionality to balance convergence speed with stability:

$$\eta = \begin{cases} 0.01, & d \leq 300 \\ 0.005, & d > 300 \end{cases} \quad \text{and} \quad \texttt{max\_iter} = \begin{cases} 1000, & d \leq 300 \\ 2000, & d > 300 \end{cases}$$

Regularization Grid Search. For each modality, the regularization parameter C is selected via grid search solely based on validation accuracy:

$$C \in \{0.01,\ 0.05,\ 0.1,\ 0.5,\ 1.0,\ 5.0,\ 10.0\}.$$

After selecting the optimal C, the classifier is retrained using the **full training data** with validation monitoring enabled.

8.2 Evaluation Metrics

Each trained model is evaluated using the following metrics reported on both training and held-out test sets:

1. **Accuracy:** $\text{Acc} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(\hat{y}_i = y_i)$
2. **Macro-Averaged F1-Score**
3. **Macro One-vs-All AUC**

Loss convergence behavior is additionally analyzed using recorded train–validation loss curves.

8.3 Random Seed Control and Reproducibility

To ensure strict determinism and full experimental reproducibility, the following controls are enforced:

- All dataset splits use fixed `random_state = 42`.
- PCA initialization uses `random_state = 42`.
- K-Means clustering (BoVW) uses `random_state = 42`.
- All NumPy random operations use `np.random.seed(42)`.

9 Results and Performance Evaluation

This section reports the classification performance of the proposed logistic regression models under four settings: image-only, text-only, categorical-only, and multimodal fusion. All results are reported using macro-averaged metrics on the held-out test set.

9.1 Single-Modality Performance

Table 1 summarizes the results obtained using each modality independently.

Table 1: Performance Metrics for Our Custom Implementation (Test Set)

Modality	**Accuracy**	**Precision**	**Recall**	**Macro-F1**	**Macro-AUC**
Images only	0.6898	0.6938	0.6893	0.6905	0.8991
Categorical only	0.7205	0.7222	0.7200	0.7178	0.8940
Text only	0.7480	0.7733	0.7476	0.7501	0.9355
Fused	**0.9047**	**0.9041**	**0.9045**	**0.9041**	**0.9888**

Among the individual modalities, the **text-based representation** achieves the strongest standalone performance (Macro-F1: 0.7501), indicating that semantic textual information provides highly discriminative cues. Categorical metadata also yields competitive performance despite its low dimensionality ($d_{\text{cat}} = 24$). Image-only performance is lower due to strong visual similarity among classes such as cucumber and zucchini-like eggplants.

9.2 Multimodal Fusion Performance

When all three modalities are fused via early feature concatenation, a substantial performance improvement is observed.

Key Experimental Result

Fusion Gain Analysis:

$$\text{Accuracy} = 0.9047, \quad \text{Macro-F1} = 0.9041, \quad \text{Macro-AUC} = 0.9888.$$

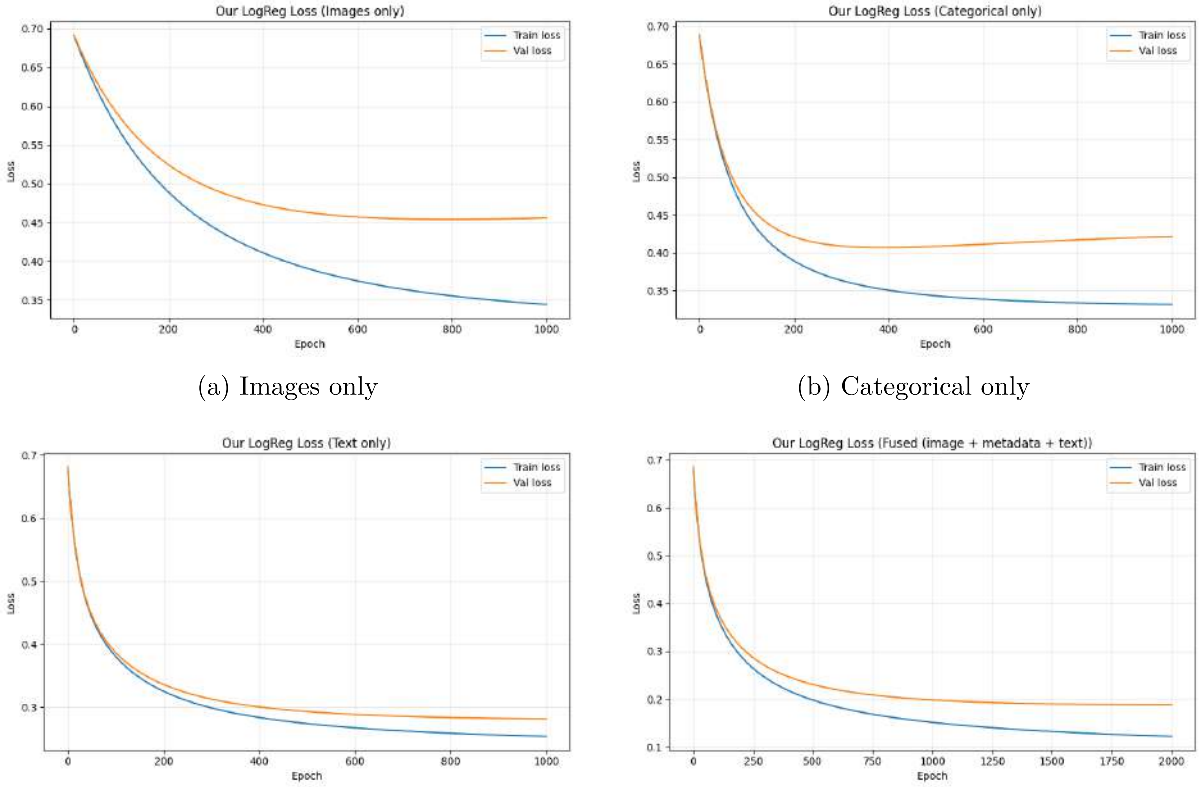
This represents an absolute macro-F1 improvement of:

$$+21.4\% \text{ over images}, \quad +18.7\% \text{ over categorical}, \quad +15.4\% \text{ over text}.$$

The near-perfect AUC (0.9888) confirms that visual shape, categorical provenance, and semantic descriptions reinforce each other, yielding a highly separable joint representation.

9.3 Training Dynamics and Convergence Behavior

Figure 4 shows the training and validation loss curves. All models exhibit stable monotonic convergence with no gradient explosion. The multimodal fused model achieves the lowest validation loss and maintains a stable generalization gap, confirming that fusion improves both optimization stability and generalization.



(a) Images only

(b) Categorical only

(c) Text only

(d) Fused (Image + Metadata + Text)

Figure 4: Training and validation loss curves. The fused model achieves the lowest validation loss and most stable convergence. Text-only learning converges faster than other single modalities.

9.4 Comparison with Scikit-Learn Implementation

Table 2: Full Comparison: Our Implementation vs. Scikit-Learn (Test Set)

Modality	**Accuracy**		**Precision**		**Recall**		**Macro-F1**		**Time (s)**	
	Our	**SK**	**Our**	**SK**	**Our**	**SK**	**Our**	**SK**	**Our**	**SK**
Images	0.6898	0.6866	0.6938	0.6906	0.6893	0.6861	0.6905	0.6872	2.92	0.57
Categorical	0.7205	0.7189	0.7222	0.7198	0.7200	0.7184	0.7178	0.7161	1.07	0.06
Text	0.7480	0.7658	0.7733	0.7780	0.7476	0.7646	0.7501	0.7655	0.98	0.13
Fused	**0.9047**	0.9095	**0.9041**	0.9092	**0.9045**	0.9094	**0.9041**	0.9091	37.90	0.80

Scientific Insight

Trade-off Analysis (Transparency vs. Speed): Our custom implementation achieves predictive parity with `scikit-learn` (difference $< 0.5\%$), validating the correctness of the derived gradients and optimization logic.

However, a substantial computational gap exists: `scikit-learn` is approx. **47x faster** in the fused setting (0.80s vs 37.90s). This is attributed to its use of the quasi-Newton LBFGS solver and C-level optimizations, whereas our implementation relies on first-order gradient descent in pure Python/NumPy to prioritize pedagogical transparency and accessible loss trajectories.

10 Conclusion

In this work, we constructed and analyzed a controlled multimodal classification dataset composed of image, text, and categorical metadata. Through rigorous feature extraction, similarity analysis, and classification experiments, we demonstrated that individual modalities exhibit limited separability, while early multimodal fusion significantly improves discriminative performance.

Scientific Insight

Final Scientific Conclusion: The strong geometric overlap observed in pure image space ($\rho_{\text{img}} \approx 0.98$) formally characterizes the dataset as intrinsically challenging. Our results confirm that semantic and structured metadata play a critical role in stabilizing decision boundaries and improving generalization under high visual ambiguity, validating the necessity of multimodal learning.

Team Contribution Statement

All stages of this project were carried out fully together with equal responsibility and shared effort. Dataset construction, synthetic image generation, manual filtering and verification of generated samples, feature engineering, multimodal representation, model implementation, training, evaluation, and report writing were all completed jointly as a single team without any division of labor.

Note: 1% of the total contribution is credited to Beyza's brother for providing 20 real-world banana images that were incorporated into the dataset.