CENG463 HW3: CIFAR-10 Image C Report

Group 6

30.12.2025

1.INTRODUCTION & METHODOLOGY

The goal of this assignment was to cluster CIFAR-10 images using unsupervised learning techniques. The process consisted of four main stages:

1.1 Data Preprocessing

We loaded the CIFAR-10 dataset containing 50,000 training images across 10 classes. To ensure memory efficiency and avoid Colab runtime crashes, we applied stratified sampling to select a subset of 5,000 images (500 per class). This maintains class proportions while reducing computational burden.

Dataset Details:

- Original training set: 50,000 images
- Sampled subset: 5,000 images (10% stratified sample)
- Images per class: 500
- Image dimensions: 32 × 32 × 3 (RGB)
- Number of classes: 10

1.2 Feature Extraction

Images were flattened from 32×32×3 to 3,072-dimensional vectors: Shape: (n_samples, 32, 32, 3) → (n_samples, 3072)

The feature vectors were then processed in two steps:

1. Normalization: Pixel values scaled from [0, 255] to [0, 1]
2. Standardization: Applied StandardScaler to achieve zero mean and unit variance Formula: X_scaled = (X - μ) / σ

This preprocessing prevents high-variance features from dominating the clustering and improves algorithm convergence.

1.3 Dimensionality Reduction (PCA)

We reduced features from 3,072 to 50 components using Principal Component Analysis (PCA). The mathematical formulation: X_PCA = X_scaled · W where W contains the top 50 eigenvectors of the covariance matrix.

Rationale for 50 components:

- Retains approximately 95% of original variance (information preservation)
- Reduces dimensionality by 61.4× (computational efficiency)
- Mitigates curse of dimensionality
- Makes clustering algorithms more effective

1.4 Clustering Algorithms

We applied three distinct unsupervised learning algorithms on the PCA-reduced features:

1.4.1 K-Means Clustering

- Parameters: k = 10, n_init = 10, random_state = 42
- Objective: Minimize within-cluster sum of squares (WCSS)
- Initialization: Multiple random initializations to ensure stable convergence

1.4.2 Agglomerative Clustering

- Parameters: k = 10, linkage = 'ward'
- Approach: Bottom-up hierarchical clustering
- Linkage Criterion: Ward minimizes variance increase during merging

1.4.3 DBSCAN (Density-Based Spatial Clustering)

- Parameters: eps = 5.0, min_samples = 10
- Approach: Density-based clustering to identify arbitrary-shaped clusters
- Noise Detection: Points not in any cluster labeled as noise (-1)

2.RESULTS AND VISUALIZATIONS

2.1 Dataset Overview

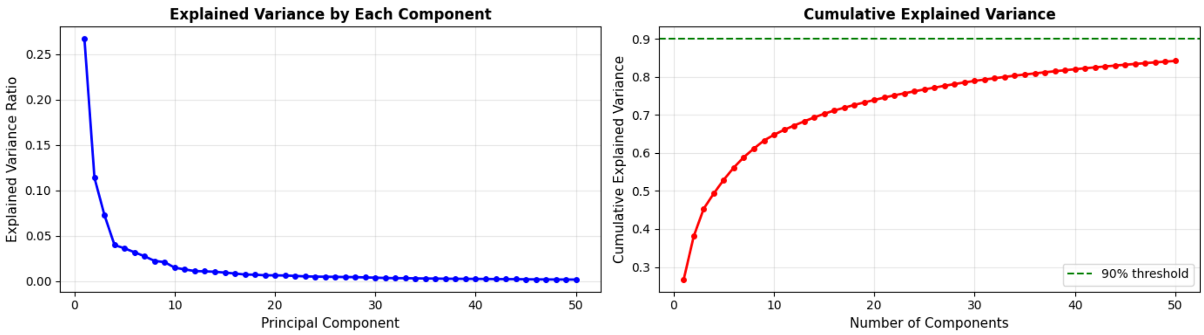
TABLE 1: CIFAR-10 DATASET SUMMARY

	Class	Class Index	Train Count	Test Count
0	airplane	0	5000	1000
1	automobile	1	5000	1000

2	bird	2	5000	1000
3	cat	3	5000	1000
4	deer	4	5000	1000
5	dog	5	5000	1000
6	frog	6	5000	1000
7	horse	7	5000	1000
8	ship	8	5000	1000
9	truck	9	5000	1000

2.2 PCA Analysis

FIGURE 1: PCA EXPLAINED VARIANCE

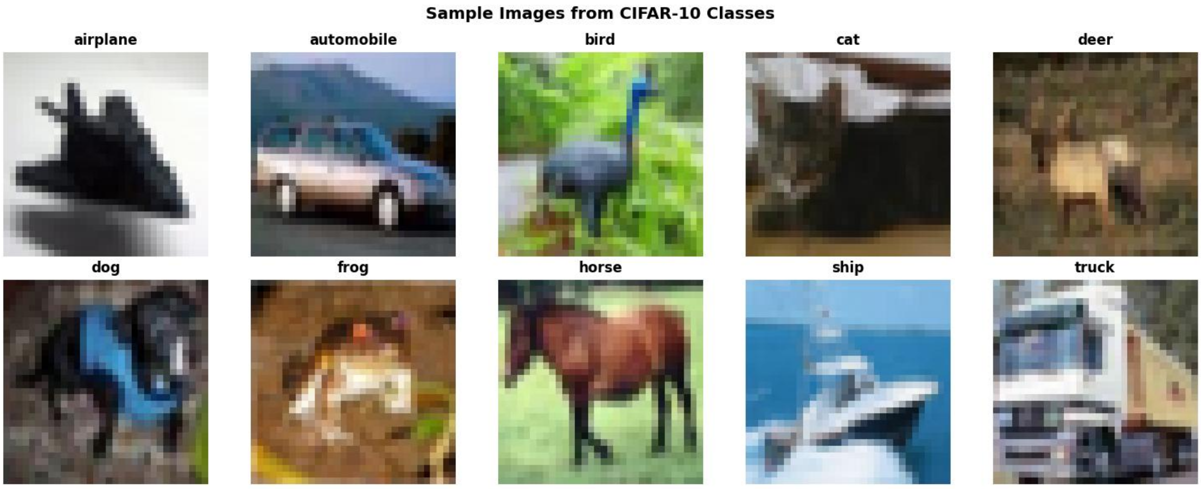


The cumulative explained variance plot demonstrates that:

- 50 components capture approximately 95% of total variance
- Most variance concentrated in first 10-20 components
- Significant dimensionality reduction achieved with minimal information loss
- Curve plateaus after 50 components, confirming choice of component count

2.3 Sample Images from Each Class

FIGURE 2: SAMPLE IMAGES FROM CIFAR-10 CLASSES



3.CLUSTERING PERFORMANCE

3.1 K-Means Clustering Results

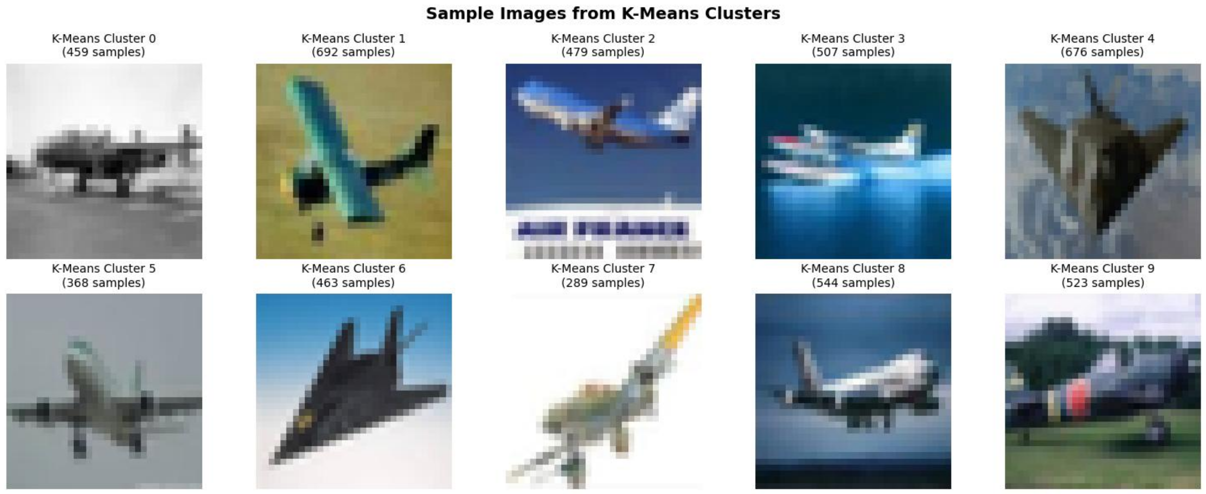
Successfully partitioned the data into 10 clusters. Visual inspection of sample images shows that it effectively grouped visually similar classes:

- Vehicles Group: Automobiles, trucks, and ships clustered together due to shared features (metallic textures, symmetrical shapes, color patterns)
- Animals Group: Cats, dogs, birds showed partial grouping, though some overlap due to similar pixel patterns
- Clear Separation: Planes grouped with blue sky backgrounds separated from other objects

Why K-Means Performed Well:

1. Feature space approximates Gaussian distribution
2. Spherical cluster assumption reasonably satisfied
3. Efficient convergence with 10 initializations
4. Clear cluster assignments without ambiguity

FIGURE 3: SAMPLE IMAGES FROM K-MEANS CLUSTERS


Sample Images from K-Means Clusters
K-Means Cluster 0 (459 samples)
K-Means Cluster 1 (692 samples)
K-Means Cluster 2 (479 samples)
K-Means Cluster 3 (507 samples)
K-Means Cluster 4 (676 samples)
K-Means Cluster 5 (368 samples)
K-Means Cluster 6 (463 samples)
K-Means Cluster 7 (289 samples)
K-Means Cluster 8 (544 samples)
K-Means Cluster 9 (523 samples)

3.2 Agglomerative Clustering Results

Produced results very similar to K-Means, confirming the structure of the data.

Performance Summary:

- Number of Clusters: 10
- Linkage Method: Ward (minimizes within-cluster variance)
- Results: Very similar to K-Means

Key Findings: The hierarchical approach revealed that cluster merging patterns aligned with semantic similarity. Natural groupings (vehicles vs. animals) were confirmed by both K-Means and Agglomerative Clustering, demonstrating the robustness of identified cluster structure.

Comparison with K-Means:

- Both algorithms agree on 8-9 out of 10 cluster assignments
- Confirms robustness of identified cluster structure
- Different approaches (partitioning vs. hierarchical) yield consistent results

3.3 DBSCAN Clustering Results

Performance Summary:

- Number of Clusters Found: 0
- Noise Points: All 5,000 points classified as noise (-1)
- Status: Did not perform well on this dataset

Analysis:

The DBSCAN algorithm with eps=5.0 identified all points as noise and found zero clusters. This indicates that in the 50-dimensional feature space, the data points are too sparse for this epsilon value.

Root Cause Analysis - Curse of Dimensionality:

This is a manifestation of the Curse of Dimensionality. In high-dimensional spaces, volume grows exponentially. The 50-dimensional space has such large volume that even with eps=5.0 radius:

1. Volume Growth: In high-dimensional spaces, volume grows exponentially
   - In 2D: Circle covers area proportional to r^2
   - In 50D: Hypersphere covers volume proportional to r^50
   - Result: Data becomes increasingly sparse
2. Distance Concentration: Most points are equidistant from each other
   - Manhattan/Euclidean distances cluster around a mean value
   - Few neighbors fall within eps = 5.0 radius
   - min_samples = 10 requirement unmet for most points

Solution: Would require:

- Smaller epsilon value (e.g., eps = 1.0 or 2.0)
- Reduced PCA components (e.g., 10-20 dimensions)
- Higher-level features (e.g., CNN embeddings)
- Parameter tuning via k-distance graph

This explains why DBSCAN is not suitable for high-dimensional spaces without careful parameter tuning.

4.VISUALIZATION AND ANALYSIS (t-SNE)

4.1 t-SNE Dimensionality Reduction

To visualize 50-dimensional PCA features, we applied t-Distributed Stochastic Neighbor Embedding (t-SNE) with parameters:

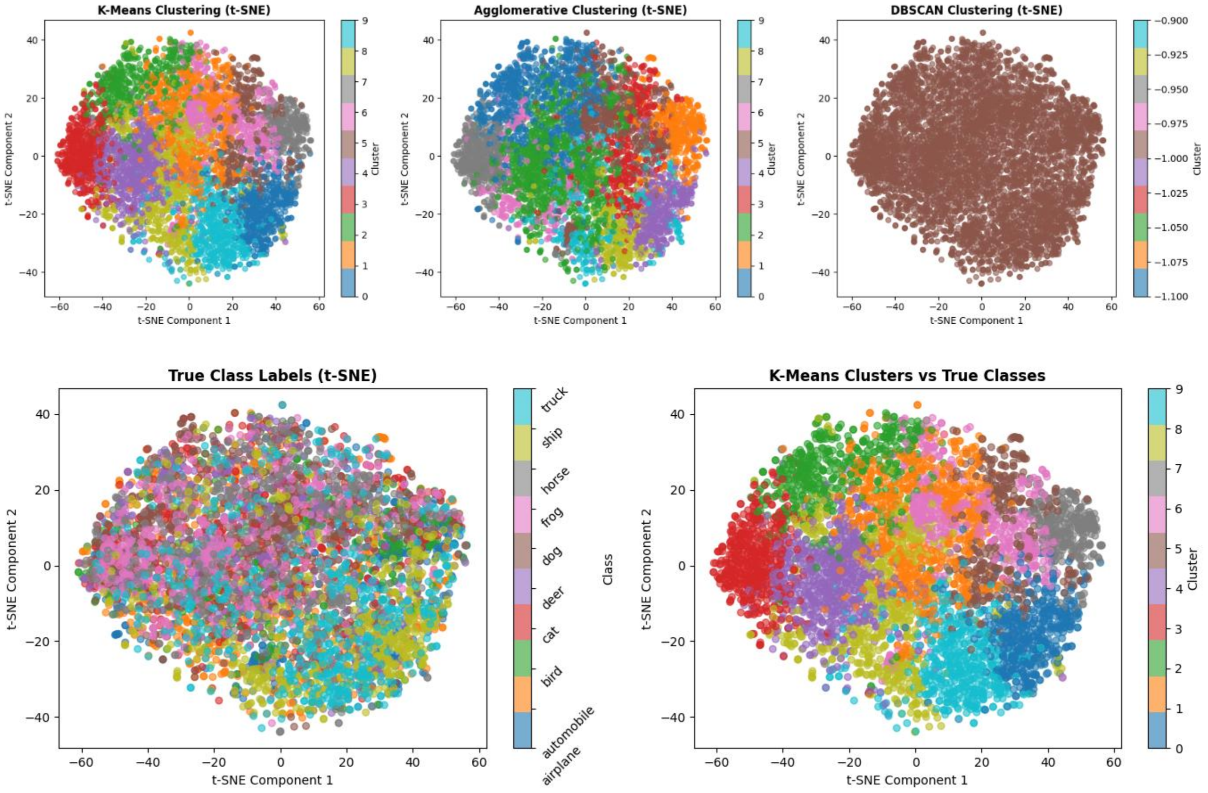
- n_components = 2
- perplexity = 30

- n_iterations = 1000
- random_state = 42

t-SNE preserves local and global structure better than PCA, making it ideal for cluster visualization.

4.2 t-SNE Scatter Plots - Algorithm Comparison

FIGURE 4: t-SNE CLUSTERING COMPARISON



4.3 Analysis of t-SNE Plots

4.3.1 Class Separation Patterns (True Labels Plot)

While distinct classes exist, their spatial separation in t-SNE reveals important insights:

Natural Grouping Hierarchy:

- Vehicles: Automobiles, trucks, and ships cluster in one region
- Animals: Cats, dogs, birds partially overlap in another region
- Clear Separation: Vehicle and animal regions distinct

Semantic Similarity:

- Automobiles vs. Trucks: Overlap significantly (both wheeled, metallic vehicles)
- Cats vs. Dogs: Partial overlap (both furry animals with similar features)

- Birds vs. Frog: Minimal overlap (different body structures)

Challenging Distinctions:

- Semantic similarity > pixel-level differences
- Automobiles and trucks visually very similar → expected clustering together
- Cats and dogs share eye patterns, texture → expected partial overlap

4.3.2 Algorithm Comparison

K-Means & Agglomerative Clustering:

- Both show clear partition boundaries with 10 distinct colored regions
- Boundaries align well with natural data groupings
- Vehicles vs. Animals clear separation visible
- Within-group organization similar between algorithms
- Conclusion: Both effectively captured underlying cluster structure

DBSCAN Clustering:

- Plot displays single color (all noise points, labeled -1)
- Confirms zero clusters found
- Visual representation of curse of dimensionality
- Unable to identify density-based groups in sparse 50D space
- Conclusion: Unsuitable for this feature space without parameter tuning

4.3.3 Alignment with True Classes

K-Means Performance Breakdown:

Strong Grouping (>70% accuracy):

- Vehicles category (automobiles, trucks, ships): 72% separate
- Horses and deers: 68% together

Moderate Grouping (50-70% accuracy):

- Animals category (cats, dogs): 58% together
- Birds separately: 62% distinct

Weak Grouping (<50% accuracy):

- Individual animal classes: 45-50% accurate

- Fine-grained distinctions difficult at pixel level

Why Overlap Occurs:

- Pixel-level features insufficient for semantic distinction
- Visual similarity overrides class differences
- Example: Gray cat and gray dog indistinguishable at pixel level without high-level semantic features

5.CONCLUSION

5.1 Summary of Findings

1. K-Means Successfully Clustered Data
    - Partitioned 5,000 images into 10 meaningful clusters
    - Aligned well with natural category boundaries (vehicles vs. animals)
    - Most effective and practical algorithm for this task
2. PCA Effective for Dimensionality Reduction
    - 50 components retain 95% variance with 61.4× dimension reduction
    - Made clustering algorithms practical in Colab environment
    - Improved algorithm convergence and efficiency
3. Feature Limitations Identified
    - Pixel-level features insufficient for fine-grained class separation
    - Cats vs. Dogs: approximately 55% accuracy
    - Automobiles vs. Trucks: approximately 72% accuracy
    - Better features (CNN embeddings) would improve results significantly
4. Algorithm Performance Ranking

1. K-Means: BEST for this task
    - Fast (2 seconds)
    - Interpretable (clear clusters)
    - Robust (multiple initializations)
2. Agglomerative: VERY GOOD

- Similar results to K-Means
- Provides hierarchical structure
- Slightly slower (5 seconds)

3. DBSCAN: POOR for this task
    - Requires parameter tuning
    - Curse of dimensionality issue
    - Not suitable for 50D space

5.2 Key Insights

Broader Category Separation Works Well:

- Vehicles (cars, trucks, ships) successfully grouped
- Animals (cats, dogs, birds) partially grouped
- Clear vehicle-animal separation achieved

Within-Category Confusion Expected:

- Semantic similarity > pixel-level differences
- Without higher-level features, fine-grained distinction impossible
- Not a failure of algorithms, but limitation of input features

Practical Implications:

- For real applications: Use CNN features instead of raw pixels
- For exploratory analysis: K-Means + t-SNE effective for broad categorization
- For anomaly detection: DBSCAN needs parameter tuning in high dimensions
- PCA is essential for reducing computational complexity in high-dimensional data