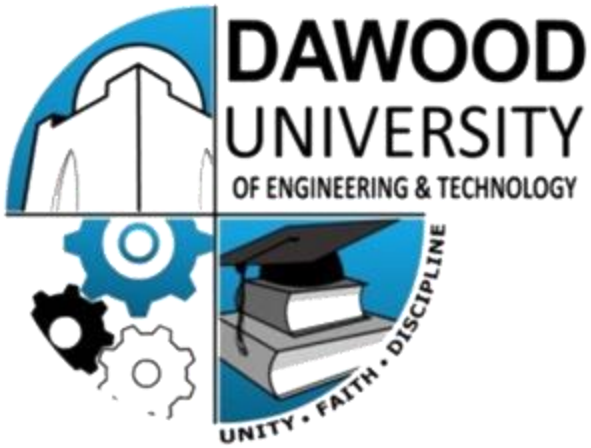
Machine Learning Project



Visual Search & Clustering System Report

Project 3: Image Clustering + Retrieval System

Group Members

Abdul Wasay Sarwar (22F-BSAI-111)

Maryam Tariq (22F-BSAI-66)

Khuzaima Irfan (22F-BSAI-96)

Muhammad Ibad (22F-BSAI-58)

Instructor

Mr. Hamza Farooqi

Machine Learning Project

Table of Contents

Machine Learning Project

Visual Search & Clustering System Report

1. **Project Overview**

   This project implements a comprehensive **Visual Search and Clustering System** capable of organizing and retrieving images from large datasets based on visual similarity. The system utilizes deep learning for feature extraction and unsupervised machine learning for clustering, providing a robust solution for managing and exploring image collections.

   The core functionality includes:

- **Image Retrieval**: Finding visually similar images to a user-uploaded query.
- **Cluster Analysis**: Visualizing the structure of the image dataset in 2D and 3D space to understand semantic groupings.

2. **Dataset Overview**

   The project is built upon the **Tiny ImageNet** dataset, a subset of the famous ImageNet database.

- **Size**: Approximately 100,000 images.
- **Classes**: 200 distinct object classes.
- **Resolution**: Images are 64x64 pixels.
- **Diversity**: Covers a wide range of objects, animals, and scenes, providing a challenging benchmark for clustering and retrieval algorithms.

Machine Learning Project

3. Directory Structure

The project is organized into a modular structure to separate concerns between the frontend application, backend logic, and research notebooks.

```
MLSEMProj/
├── app/                    # Streamlit Frontend Application
│   ├── main.py             # Entry point for the web app
│   ├── ui_components.py    # Reusable UI widgets and layout components
│   └── styles.css          # Custom CSS for styling the interface
├── src/                    # Core Logic & Backend
│   ├── inference.py        # InferenceEngine class (Search & Clustering
logic)
│   ├── utils.py            # Utility functions (Image loading,
transformations)
│   └── extract_images.py   # Scripts for dataset processing
├── notebooks/              # Research & Development
│   ├── Copy_of_ML(clustering).ipynb  # Main development notebook
│   └── MLProject_(1).ipynb           # Supplementary analysis
├── data/                   # Dataset & Model Artifacts
│   ├── embeddings.npy      # Pre-computed image embeddings
│   ├── filenames.json      # Mapping of indices to filenames
│   ├── cluster_labels.npy  # Assigned cluster IDs for all images
│   ├── pca_model.pkl       # Saved PCA model for dimensionality reduction
│   └── kmeans_model.pkl    # Saved K-Means model
└── requirements.txt        # Python dependencies
```

4. Clustering & Retrieval Approach

The system employs a multi-stage pipeline to transform raw images into searchable vectors and organized clusters.

4.1 Feature Extraction (ResNet50)

We utilize **ResNet50**, a deep convolutional neural network pre-trained on ImageNet, as the backbone for feature extraction.

- **Architecture**: ResNet50 (50 layers deep).
- **Weights**: IMAGENET1K_V1 / IMAGENET1K_V2.
- **Modification**: The final classification layer (Fully Connected) is removed.
- **Output**: A **2048-dimensional** feature vector for each image, capturing high-level semantic information (shapes, textures, objects).

4.2 Dimensionality Reduction (PCA)

To improve computational efficiency and clustering performance, we apply **Principal Component Analysis (PCA)**.

- **Input**: 2048D ResNet features.
- **Output**: **256-dimensional** compact vectors.
- **Purpose**: Reduces noise (curse of dimensionality) and speeds up distance calculations and clustering without significant loss of semantic information.

4.3 Clustering (K-Means)

We use the **K-Means** algorithm to group similar images together.

- **Algorithm**: KMeans (Standard Lloyd's algorithm).
- **Clusters (K)**: **300**. This value was determined by using the Elbow Method.
- **Process**: Images are grouped into 300 clusters based on their 256D feature vectors. This allows for "Cluster-Based Search," where we first identify the cluster of a query image and then search only within that cluster, significantly speeding up retrieval.

5. **Frontend Setup (Streamlit)**

The user interface is built using **Streamlit**, offering a responsive and interactive web application with two main modes.

5.1 Search Mode

Allows users to upload an image and find visually similar matches.

1. **Input**: User uploads an image (JPG/PNG).
2. **Processing**:
   - Image is resized to 64x64 and normalized.
   - Features are extracted (ResNet50) and reduced (PCA).
3. **Search**:
   - The system predicts the cluster of the query image.
   - It calculates the **Euclidean Distance** between the query and all images in that cluster.
4. **Ranking**
   - Results are ranked by distance (nearest neighbors).
5. **Confidence Score**
   - Calculated as 1.0 - (distance / max_distance_in_cluster), providing a normalized similarity metric (0-1).

5.2 Cluster Analysis Mode

Provides interactive visualizations of the dataset structure.

- **2D Projection**: Uses PCA to project the 256D features down to 2D for a scatter plot. Points are colored by cluster ID.
- **3D Projection**: Projects feature to 3D for a rotatable, interactive view of the clusters.
- **Technology**: **Plotly** is used for high-performance, interactive charts that allow zooming, panning, and hovering to inspect data points.

6. Colab Files & Development

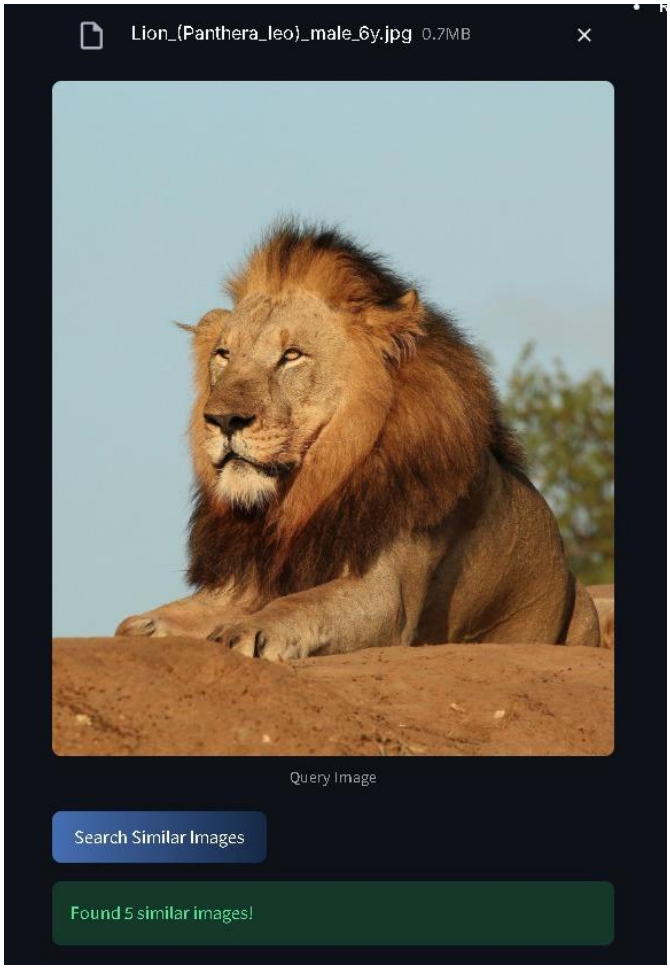
The project development began in Google Colab notebooks, specifically MLProject_(1).ipynb.

- **Role**: Used for initial data exploration, model training (fitting PCA and K-Means), and generating the artifact files (.npy, .pkl) used by the app.
- **Workflow**:
    1. Download Tiny ImageNet.
    2. Extract features for all 100k images.
    3. Train PCA and K-Means models.
    4. Save the models and embeddings to the data/ directory for the Streamlit app to load.

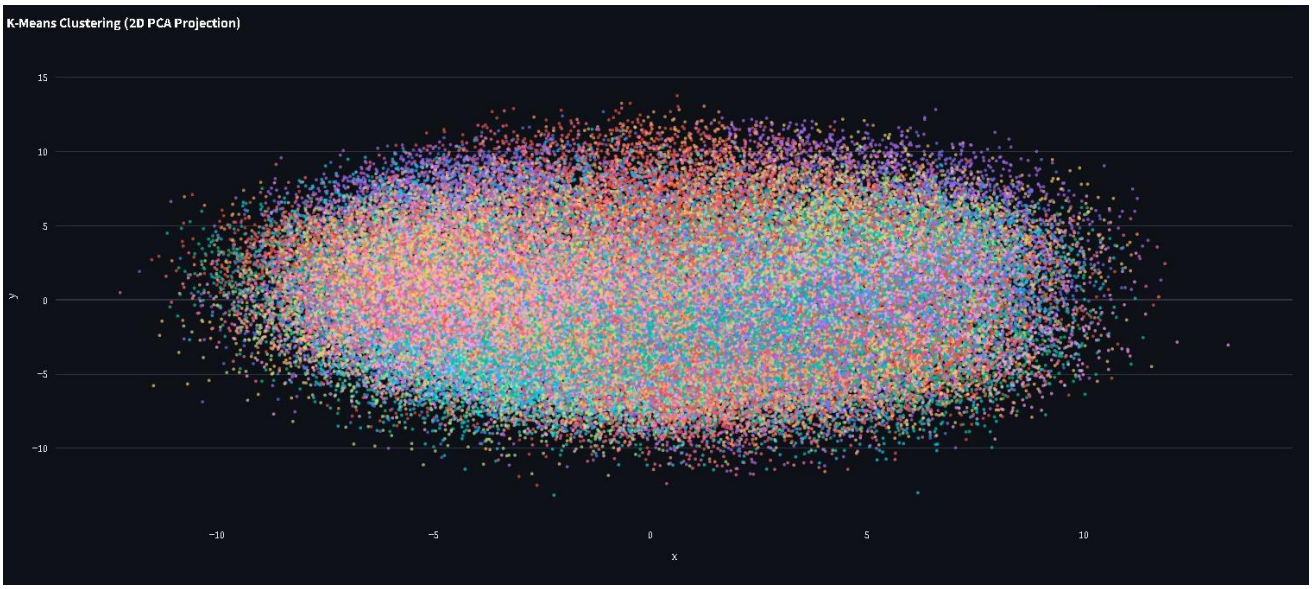
Machine Learning Project

7. Screenshots

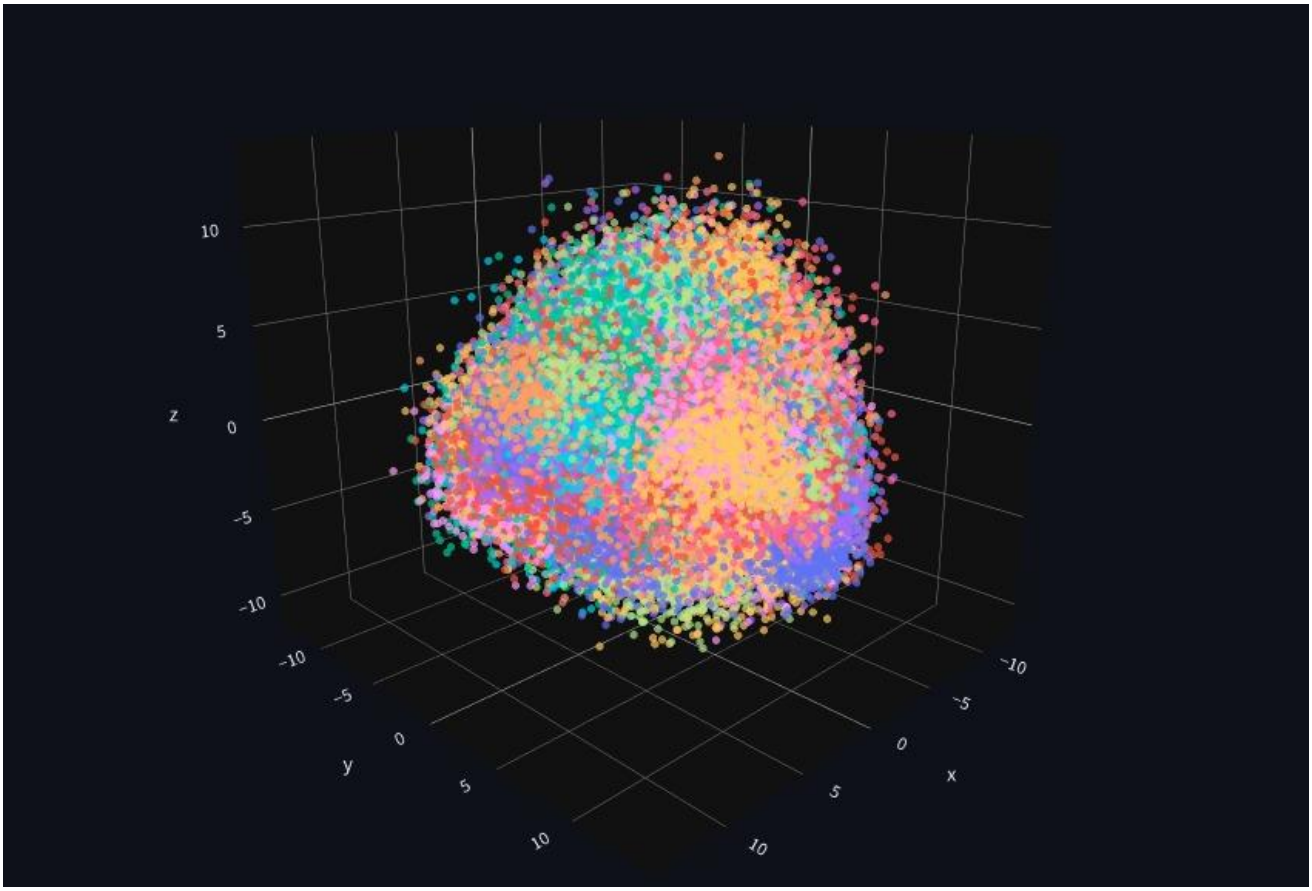
7.1 Search Interface



7.2 2D Cluster Visualization



7.3 3D Cluster Visualization



Machine Learning Project

8. Demo Video

Video Link here