Report on Comparative Image Histogram Analysis and 3D Clustering Using Gaussian Mixture Models

Introduction

This report covers two main tasks. The first task involves comparing the histograms of two similar images using various metrics. The second task involves creating a 3D dataset, applying Gaussian Mixture Modeling (GMM) clustering, and visualizing the clustering process using Expectation-Maximization algorithm.

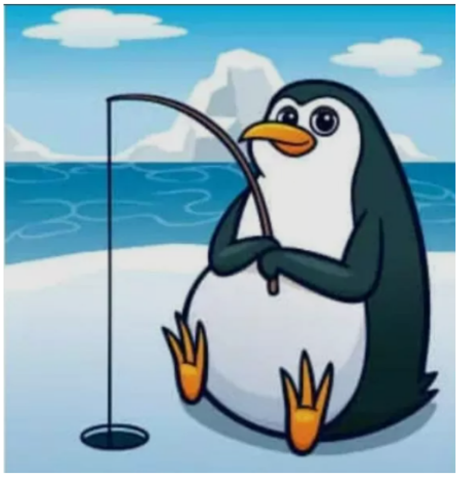
Task 1: Image Histogram Comparison

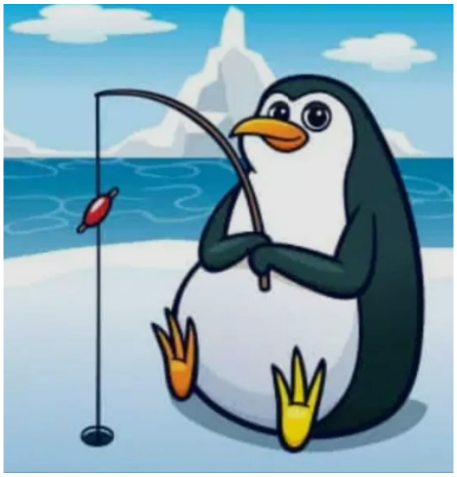
Objective

1. **Prepare Two Similar Images:** Prepare two images that are visually similar for histogram comparison.
2. **Draw Histograms:** Draw and plot histograms for each image to visualize their pixel intensity distributions.
3. **Compare Distributions:** Compare the distributions using Kolmogorov-Smirnov (KS) test, Cross-Entropy, KL-Divergence, and JS-Divergence.

Question 1: Prepare Two Images That Are Fairly Similar

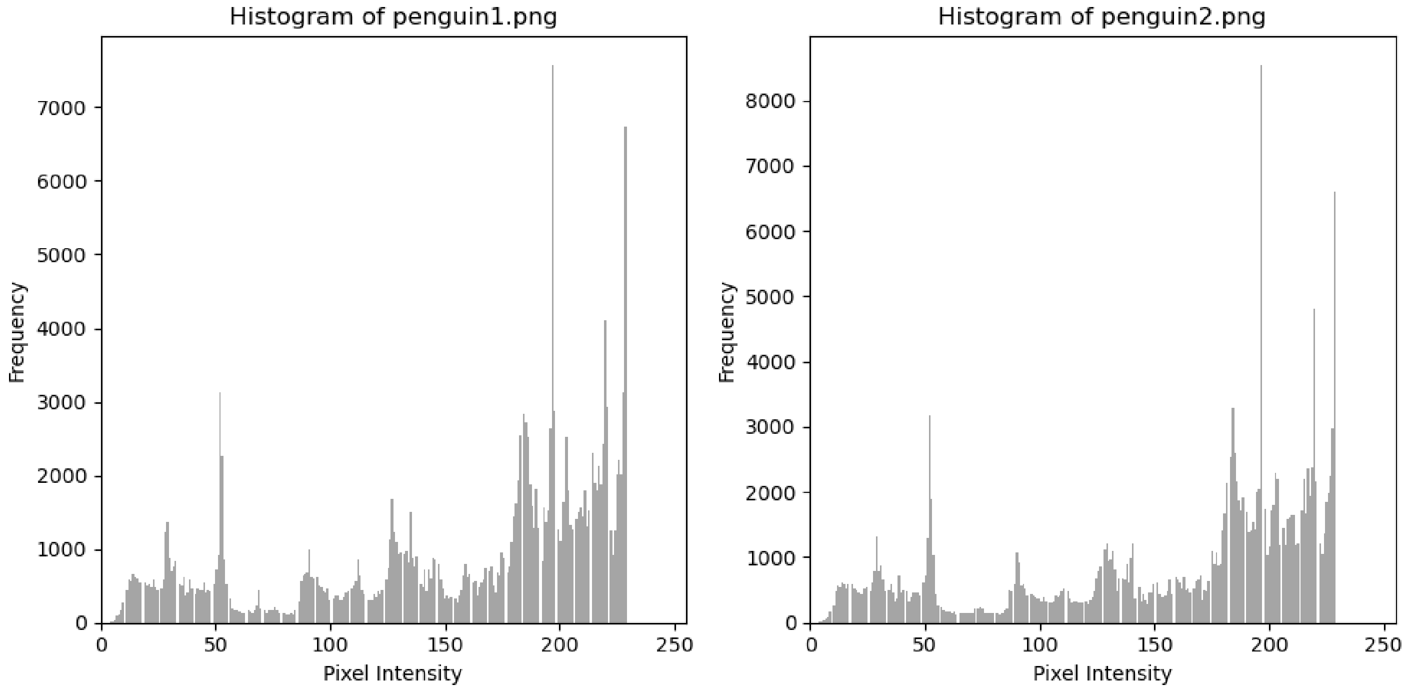
For this task, two images named `penguin1.png` and `penguin2.png` were selected. Both images are visually similar, making them suitable for comparison. These images appear alike; however, there are still a few subtle differences such as the color of the legs, the shape of the mountain, the size of the hole, and the presence of food on the hook. The images are displayed below:





Question 2: Draw Histograms of Each Image

Histograms for each image were plotted to show the distribution of pixel intensities. The histograms are represented as follows:



The histograms for `penguin1.png` and `penguin2.png` reveal the distribution of pixel intensities in grayscale. The x-axis represents pixel intensity values, and the y-axis represents the frequency of each intensity.

Question 3: Compare Distributions Between Both Images

The distributions of pixel intensities between the two images were compared using four different metrics: Kolmogorov-Smirnov (KS) Statistic, Cross-Entropy, KL-Divergence, and JS-Divergence. Each of these metrics provides unique insights into the similarity or difference between the two histograms.

1. Kolmogorov-Smirnov (KS) Statistic and p-value

- **Statistic**: 0.04296875
- **p-value**: 0.9726060102257182

Explanation: The Kolmogorov-Smirnov (KS) Statistic measures the maximum difference between the empirical cumulative distribution functions (CDFs) of two distributions. In this case, it calculates the largest vertical distance between the cumulative distributions of pixel intensities for the two images.

A low KS statistic of 0.04296875 indicates a minimal discrepancy between the cumulative distributions of the two images, suggesting that the histograms are quite similar.

The p-value of 0.9726060102257182 tests the null hypothesis that the two distributions are identical. A high p-value (close to 1) means that there is no significant difference between the distributions, reinforcing the interpretation that the two images have similar pixel intensity distributions.

2. Cross-Entropy

- **Value**: 5.06742227809366

Explanation: Cross-Entropy measures the difference between two probability distributions. It quantifies the amount of information required to describe one histogram using the probability distribution of the other histogram.

In this case, a Cross-Entropy value of 5.06742227809366 suggests a moderate level of similarity between the histograms. Lower cross-entropy values would indicate a higher similarity, meaning that less information is needed to describe one histogram using the other. This value suggests that while there is some difference, it is not substantial.

3. KL-Divergence

- **Value**: 0.026940525960668642

Explanation: KL-Divergence, or Kullback-Leibler Divergence, measures how one probability distribution diverges from a second, reference distribution. It calculates the average number of extra bits required to encode samples from one distribution using a code optimized for the other distribution.

A KL-Divergence value of 0.026940525960668642 indicates a small divergence between the two histograms. This low value suggests that the two distributions are quite similar. KL-Divergence is not symmetric, but in this context, the low value supports the idea that the histograms are similar.

4. JS-Divergence

- **Value**: 0.006448779847129416

Explanation: JS-Divergence, or Jensen-Shannon Divergence, is a symmetric measure that captures the similarity between two probability distributions. It is computed as the average of KL-Divergence between each distribution and the average of the two distributions.

A JS-Divergence value of 0.006448779847129416 is very low, indicating that the distributions are highly similar. JS-Divergence provides a more balanced view than KL-Divergence because it is symmetric and bounded between 0 and 1. The low value reinforces the conclusion that the pixel intensity distributions of the two images are very similar.

Summary

Overall, the metrics computed suggest a high degree of similarity between the distributions of pixel intensities in the two images. The low values of the KS Statistic, Cross-Entropy, KL-Divergence, and JS-Divergence all indicate that the histograms of the images are quite similar, confirming that the images are visually comparable with only minor differences.

Task 2: Gaussian Mixture Modeling (GMM)

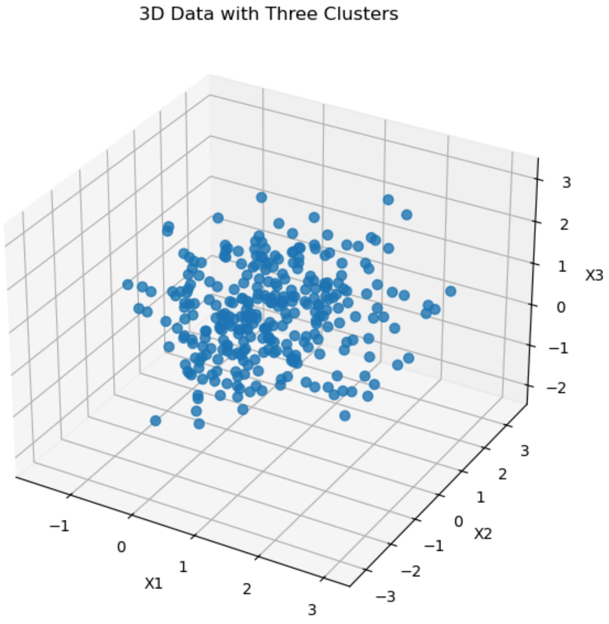
Objective

1. **Create a Random Dataset:** Generate a 3D dataset with three distinct clusters.
2. **Apply GMM Clustering:** Use Expectation-Maximization algorithms to cluster the data using Gaussian Mixture Modeling.
3. **Visualize GMM Steps:** Visualize the clustering process at different steps of the Expectation-Maximization algorithm.

Question 1: Create a Random Dataset in 3D Space with Three Clusters

A random dataset was created in 3D space, containing three distinct clusters. The clusters were generated with the following parameters:

- **Number of Samples:** 300
- **Cluster Centers:** [(0, 0, 0), (1, 1, 1), (1, -1, 1)]
- **Cluster Standard Deviation:** 0.8



Question 2: Applying Gaussian Mixture Modeling (GMM)

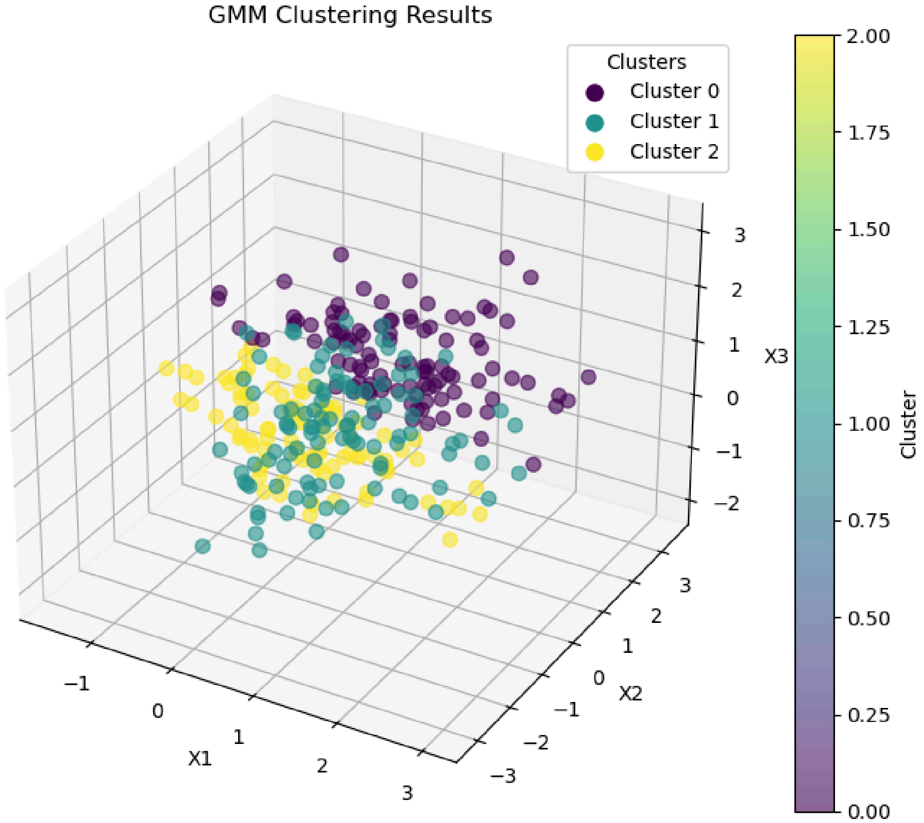
Gaussian Mixture Modeling (GMM) is applied to cluster the synthetic dataset. GMM is a probabilistic model that assumes the data is generated from a mixture of several Gaussian distributions with unknown parameters.

`GaussianMixture` **Class**: Initializes the GMM model with the number of clusters.

`fit` **Method**: Fits the model to the data.

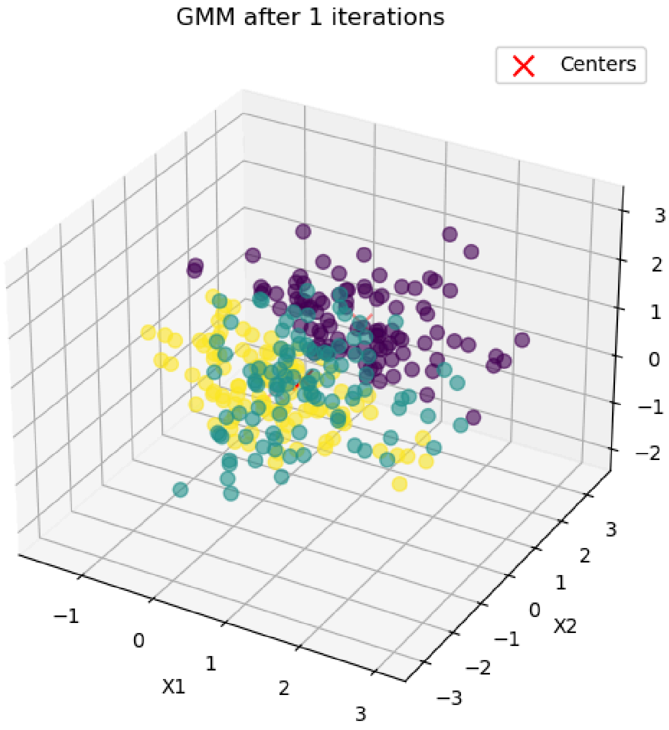
`predict` **Method**: Assigns cluster labels to each data point based on the fitted model.

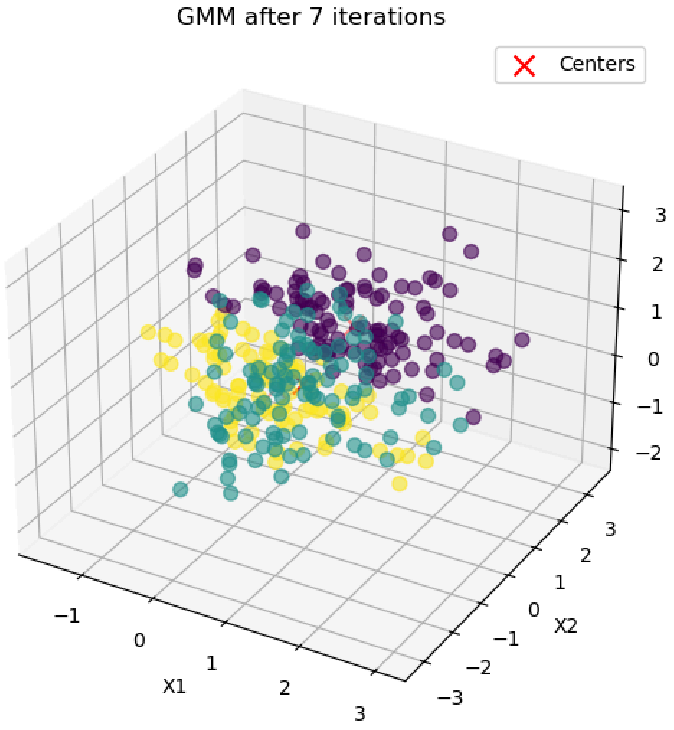
The 3D scatter plot color-coded by cluster labels shows the clustering result from GMM.



Question 3: Visualizing Expectation-Maximization (EM) Steps

To understand the GMM clustering process, it is crucial to visualize intermediate steps of the Expectation-Maximization (EM) algorithm. EM alternates between estimating the parameters of the Gaussian components (Expectation step) and assigning data points to these components. Clusters after 1 iteration and 7 iterations are shown below:





Summary

Task 2 involved clustering a synthetic 3D dataset with three clusters using Gaussian Mixture Modeling (GMM). The dataset was created and clustered with GMM, which was visualized through 3D scatter plots. Additionally, the Expectation-Maximization (EM) steps were visualized to illustrate the iterative clustering process. The task effectively demonstrated GMM's clustering capabilities and the evolution of clustering results over iterations.