Advance Machine Learning and Neural Networks
Assignment 1

Task 1:

1. Prepare two images that are fairly similar.
2. Draw a histogram of each image in the previous task is somehow the distribution of the image. Plot the histogram of each image.
3. Compare the distributions between both images with Kolmogorov- Smirnov, Cross-Entropy, KL-Divergence, and JS-Divergence.

For this task, I used a Python-based approach to generate and compare two images.

1. Image Preparation

I started by creating two images using a tool called *Stable Diffusion* that generates images from text descriptions. The first image was a landscape with mountains, trees, a house, and a river at sunset. I made slight changes to the second image by overlaying lines and waves using a Python library called *Pillow*, which resulted in a similar but not identical image.

2. Grayscale Conversion and Histogram Plotting

To focus on the structure and brightness of the images without the complexity of colours, I converted both images to grayscale. I then generated histograms, which display how frequently each shade of grey appears in the image. This helps to understand the distribution of brightness in both images.

3. Statistical Comparison

I compared the two histograms using four methods:

a) **Kolmogorov-Smirnov (KS) Test**: This test checks for differences between two distributions. The result of 0.09 indicated that the images are quite similar.

b) **Cross-Entropy**: This measures how surprising it is to see one image if we expect it to look like the other. The score of 5.38 indicated some noticeable differences between the images.

c) **KL-Divergence**: This quantifies how one image's histogram differs from the other. In this case, the result was infinity, meaning the method was not suitable due to large differences.

d) **JS-Divergence**: This is a more stable version of KL-Divergence, and it gave a small result (0.01), confirming that the images are slightly different but still very similar.

4. Conclusion

The analysis showed that the second image was only slightly altered from the first. The small KS and JS-Divergence values confirmed the high similarity, while Cross-Entropy highlighted some noticeable changes.

Advance Machine Learning and Neural Networks
Assignment 1

Task 2:
(1) Create a random dataset in 3D space that has three clusters.
(2) By using Expectation Maximization algorithms, try to implement Gaussian Mixture Modelling (GMM) clustering algorithm and cluster these data correctly.
(3) You should visualize at least two E and M steps in between.

1. Creating the Dataset:

I generated a random dataset of points in 3D space, which was divided into three distinct clusters. Each cluster had 100 points, giving a total of 300 data points. The clusters were sufficiently spaced out, making it easier for the algorithm to identify them.

2. Visualizing the Initial Data:

I plotted a 3D scatter plot of the points to confirm the presence of three clusters. This helped to verify the accuracy of the clustering algorithm.

3. Applying the Gaussian Mixture Model (GMM):

To cluster the data, I used the GMM algorithm, which assumes that data points come from a combination of several Gaussian (bell-shaped) distributions. GMM alternates between two steps:

a) **Expectation Step (E-Step)**: The algorithm calculates the probability that each point belongs to each cluster.

b) **Maximization Step (M-Step)**: It updates the cluster parameters (mean, covariance, etc.) to maximize the likelihood of points being assigned to the right clusters.

4. Iterative Process:

I ran the GMM algorithm for 100 iterations, visualizing the progress after every 10 iterations. Initially, the clusters were far from their true positions, but with each iteration, the algorithm improved the cluster assignments.

5. Final Results:

After completing 100 iterations, GMM successfully identified the three clusters. The final 3D plot showed that the data points were grouped correctly, aligning well with the original clusters.

6. Conclusion:

The GMM algorithm was effective in clustering the data. By applying the E and M steps iteratively, the algorithm progressively improved its understanding of the underlying data structure, making it an excellent tool for uncovering hidden patterns in complex datasets.