Guidelines:
The solution to this HW should be submitted as a Jupyter Notebook (please run all blocks before submitting the code). The text should be added between blocks in response to the different sections.

1. Plot a simple example in 2D for which calculating the SVD of X is not equivalent to calculating the Principal components of X. Explain.

   Download MNIST dataset using the following lines:
   import tensorflow as tf
   $tf.keras.datasets.mnist.load\_data()$

2. Use the two leading principle components to project a 1000 sample of the digit '1' into a two-dimensional space. Use a scatter plot to show the result.

3. Use the two leading principle components to project a 1000 samples of the digit '0'. Use a scatter plot to show the result.

4. Use the two leading principle components to project a 2000 combined samples from the last two sections. Use a scatter plot to show the result, with color to indicate the sample label.

5. How many components are required to capture 90% of the variance of the datasets from the last three sections?

   Select one object from the processed COIL20 dataset
   https://www.kaggle.com/datasets/codebreaker619/columbia-university-image-library

6. Embed the selected object into a two-dimensional space using Diffusion Maps- show the results for several values of σ. What is considered a good value for σ in this example?

7. Use TSN-E to embed all objects from COIL20. Run the algorithm twice and compare the results. Are they identical? How can you stabilize the results?

8. Compare the results from the previous section to UMAP. Both in terms of run time and representation quality.