CS-449, Fall 2025
Homework #2: Autoencoders
Due Date: Monday, November 19th @ 11:59PM
Total Points: 10.0

In this assignment, you will work with your group to implement and train an autoencoder on a Emoji dataset **using PyTorch.** You may discuss the homework with other groups, but do not take any written record from the discussions. Also, do not copy any source code from the Web.

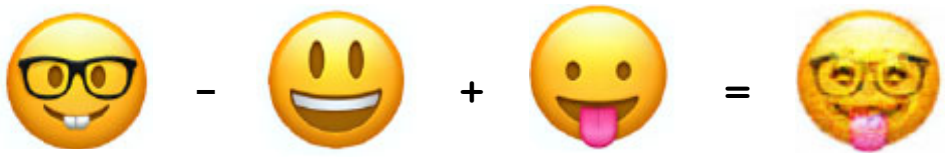
Guidelines

- You can choose an architecture of your choice to build your autoencoder.
  Note: A convolution-based architecture (https://arxiv.org/abs/1511.06434) is encouraged.
- Your autoencoder should use MSE as its primary loss function.
- You should experiment to select the size of your latent embedding.
- You may use regularization techniques to enhance the performance of your autoencoder.
- You must ensure that the output dimensionality of the autoencoder matches the input.
- You must use the Huggingface Emoji dataset for this project
  (see https://huggingface.co/datasets/valhalla/emoji-dataset).
- You should use the image "text" to select a related subset of images to work with (*e.g.,* "face", "superhero", etc.).
- You should divide this subset into training, validation and test sets using a 60/20/20 ratio.
- You should apply data augmentation or adversarial examples to expand these splits into approximately 600, 200 and 200 images, respectively.
- You are encouraged to reduce the resolution of the dataset to something appropriate for computational resources (*e.g.,* 64 x 64 x 3).
- You may find the python PIL package useful for manipulating images.
- You may find it helpful to use a random number seed for reproducibility when debugging.
- You should save your model weights and latent representations.
- You can complete the assignment entirely as a Jupyter Notebook, or you can turn in a single Python file with an accompanying PDF of the results.
- Brief starter code is provided that illustrates how to read the dataset, display an Emoji and reduce the resolution. You can completely disregard this code and write your own to perform these operations.

Steps to complete the homework

1. (5.0 points) Implement and train your autoencoder on the subset of the Emoji dataset that you selected and augmented:
   a. describe your dataset and the steps that you used to create it,
   b. provide a summary of your architecture
   c. discuss and explain your design choices,
   d. list hyper-parameters used in the model,
   e. plot learning curves for training and validation loss as a function of training epochs,
   f. provide the final average error of your autoencoder on your test set,
   g. provide a side-by-side example of 5 input and output images, and
   h. discuss any decisions or observations that you find relevant.

2. (5.0 points) Select an attribute from the Emoji dataset (internal or external to your selected subset) to compose with any image from your selected subset. Use vector arithmetic on latent representations to generate a composite image that expresses the attribute. For example, I chose to add the glasses from "nerd face" to the "face with stuck out tongue":



2306: Nerd Face	0001: Smiling Face With Open Mouth	1335: Face with Stuck Out Tongue	Generated Image

   a. specify which attribute you selected, the vector arithmetic applied and the resulting image(s) as displayed above,
   b. provide a qualitative evaluation of your composite image, and
   c. discuss ways to improve the quality of your generated image.

Submission Instructions

Turn in your homework as a single zip file, in Canvas. Specifically:

1. Create a single Jupyter Notebook or a single PDF file with the answers to the questions above, and your graphs.
2. Create a single ZIP file containing:
   - Your `homework2.pdf`, and `.py` code file, or
   - Your Juptyer Notebook
3. Turn in the zip file under Homework #2 in Canvas.

Good luck, and have fun!