

# CS 639: Deep Learning for Computer Vision, Spring 2023

## Problem Set 4

Instructor: Yong Jae Lee (yongjaelee@cs.wisc.edu)

TA: Utkarsh Ojha (uojha@wisc.edu)

TA: Mu Cai (mucai@cs.wisc.edu)

**Due: Tuesday, May 2<sup>nd</sup>, 11:59 PM**

### Instructions

1. Download the zipped assignment file from Canvas.
2. Once you unzip the downloaded content, please upload the folder to your Google Drive. Then, open each `*.ipynb` notebook file with Google Colab by right-clicking the `*.ipynb` file. No installation or setup is required. For more information, please see this tutorial on [using Colab](#).
3. Next, we recommend editing your `*.py` file on Google Colab, set the ipython notebook and the code side by side. Work through the notebook, executing cells and implementing the codes in the `*.py` file as indicated. You can save your work, both `*.ipynb` and `*.py`, in Google Drive (click “File” -> “Save” or press “Ctrl/Cmd + s”) and resume later if you don’t want to complete it all at once.
4. While working on the assignment, keep the following in mind:
  - The notebook and the python file have clearly marked blocks where you are expected to write code. **Do not write or modify any code outside of these blocks.**
  - **Do not add or delete cells from the notebook.** You may add new cells to perform scratch computations, but you should delete them before submitting your work.
  - **Run all cells, and do not clear out the outputs, before submitting.** You will only get credit for code that has been run.
5. Once you have completed a notebook, download the completed `uniqueid_PS4.zip` file, which is generated from your last cell of the `style_transfer.ipynb` file. Submit this to Canvas. Note that only one person from the group will need to do this.
6. **You may complete the assignment individually or with a partner (i.e., maximum group of 2 people). If you worked with a partner, provide the name of your partner in the `*.ipynb` file. We will be using MOSS to check instances of plagiarism/cheating.**

As part of this assignment, you will first implement generative adversarial networks and generate nice looking MNIST images. Then, you will work on style transfer algorithms.

The notebooks `generative_adversarial_networks.ipynb` and `style_transfer.ipynb` will walk you through those pipelines in PyTorch. You are required to write code on `gan.py` and `style_transfer.py`.<sup>1</sup>

---

<sup>1</sup> The coding assignment is adapted from Stanford CS 231n.

**[OPTIONAL] Extra credit short answer problems [up to 0.5% each added to final class score]**

Save your answers into a PDF, and submit it together with your code to Canvas.

1. Explain why VAEs tend to generate blurrier images than GANs.
2. Explain why the gram matrix is a good choice for representing image style.