**BME646/ECE695DL: Homework 7**

**Spring 2022**
**Due Date: Monday, April 11,2022 (11:59pm ET)**
**Extension with 5 points/day penalty: Saturday, April 16, 2022 (11:59pm ET)**

Turn in your solutions via BrightSpace.

# 1 Introduction

The overall goal of this homework is to provide you with a basic understanding of adversarial training with the use of a Generator and a Discriminator. While the Discriminator will look very similar to the classification network you created for your **HW4**, the generator will involve new ideas. In particular, your Generator code will use Transpose Convolutions to expand noise vectors into images that will look very similar to the images in your training data.

## 1.1 Goals

This homework has the following specific goals:

1. To develop and train a Generative Adversarial Network in which the Discriminator is based on your solution to **HW4**.
2. To use the Large-scale CelebFaces Attributes (CelebA) Dataset for this homework:

   `https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html`.

   What that means is that your network will be creating "deep fakes" of the face images in the CelebA dataset.

### 1.2 Background

Generative Adversarial Networks (GANs) are remarkable due to their ability to take random noise and map it into images. Depending on the data these models are trained on, your generator-produced images will look very similar to the training images without being exactly the same as any of them.

The CelebFaces Attributes Dataset (CelebA) posted at:

`https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html`

is a large-scale face attributes dataset with more than 200,000 celebrity images, each with 40 attribute annotations. The images contain a large variety of poses and backgrounds. The processed and cropped images provided for this homework have been sourced from a pre-processed subset of the data with dimensions $64 \times 64$. You will find a little more than 89,000 images in the dataset specified for this homework

## 2 Getting Ready for This Homework

Before embarking on this homework, do the following:

1. Go through Slides 26 through 43 of the Week 9 slide deck on Semantic Segmentation and develop a good understanding of the concept of what is meant by Transpose Convolution.
2. Also go through the Slides 43 through 52 of the same set of Week 9 slides to fully understand the relationship between the Kernel Size, Padding, and the Output Size for Transpose Convolution. Make sure you understand the example shown on Slide 45 in which a 4-channel $1 \times 1$ noise vector is expanded into a 2-channel $4 \times 4$ noise image. This example is foundational to designing the Generator side of a GAN.
3. Understand the GAN material on Slides 61 through 77 of the Week 11 Slide deck on "Generative Adversarial networks":

   `https://engineering.purdue.edu/DeepLearn/pdf-kak/GAN.pdf`
4. For additional depth, you may wish to read the original GAN paper by Goodfellow et el.:

`https://arxiv.org/pdf/1406.2661.pdf`

5. When you are learning about a new type of a neural network, playing with an implementation by varying its various parameters and seeing how that affects the results can often help you gain deep insights in a short time. If you believe in that philosophy, execute the following the script in the `ExamplesAdversarialLearning` directory of DLStudio:

   dcgan_DG1.py

   It uses the `PurdueShapes5GAN` dataset that is described on Slides 54 through 59 of the Week 11 slides. Instructions for downloading this dataset are on the main DLStudio webpage.

# 3 Instructions

1. Download the data from the BrightSpace link for this homework.
2. In the data provided, we have 57k images in the **Train** folder and 32k images in the **Test** folder. You should combine the images in both these folders and use them for training.
3. Design your Generator and Discriminator. Use you HW4 solution for the Discriminator. Write code for the Generator using the notion of Transpose Convolution. Be careful with how you measure the loss for updating the Discriminator and the Generator parameters.
4. Document your work, plot your generated images and the loss functions as training progresses.
5. Submit and bring an end to Homework 7.

# 4 Submission Instructions

You can assume that the files exist locally.

- Make sure to submit your code in Python 3.x and not Python 2.x.

- Create a .zip archive with the following required files:

  `hw07_training.py`

  pdf report(see the submission template)

  and optionally any additional helper python modules such as `model.py`, `dataloader.py`, etc. and upload it onto the assignment link on BrightSpace.

- **Please do NOT include your trained model in your submission:**
  `net.pth`

  We will be executing your submitted code to generate these files for verification during validation.

- **Your code must be your own work.** We will use your source code for plagiarism detection and verification of performance. Submission of both your source code and the report (in pdf) is mandatory to receive a grade.

- You can resubmit a homework assignment as many times as you want up to the deadline. Each submission will overwrite any previous submission.