Deep Learning Final Project Proposal
Group 3

What problem did you select and why did you select it?
The problem we are looking to solve is generating better-quality animation faces by optimizing GANs, DCGANs

What database/dataset will you use? Is it large enough to train a deep network?
We aim to use anime faces images obtained from Kaggle. This is a dataset consisting of 21551 anime faces scraped from www.getchu.com, which are then cropped using the anime face detection algorithm in https://github.com/nagadomi/lbpcascade_animeface. All images are resized to 64 * 64 for the sake of convenience. Please also cite the two sources when using this dataset.

What deep network will you use?
We will be using the DCGANs model with the additional optimization method from paper and tested by us.

Will it be a standard form of the network, or will you have to customize it?
We will use a standard version of the network as the baseline, and apply optimizations to it.

What framework will you use to implement the network? Why?
We will implement the network with TensorFlow since the baseline model is implemented with TensorFlow, so we will also be utilizing the same framework.

What reference materials will you use to obtain sufficient background on applying the chosen network to the specific problem that you selected?
We will be referring to the following papers and websites for ideas and background information:
https://machinelearningmastery.com/how-to-develop-a-conditional-generative-adversarial-network-from-scratch/
https://github.com/hwalsuklee/tensorflow-generative-model-collections
https://machinelearningmastery.com/how-to-evaluate-generative-adversarial-networks/
https://www.hindawi.com/journals/misy/2022/9005552/
https://arxiv.org/pdf/1801.09195.pdf
https://proceedings.neurips.cc/paper_files/paper/2018/file/90365351ccc7437a1309dc64e4db32a3-Paper.pdf
https://paperswithcode.com/method/label-smoothing

How will you judge the performance of the network? What metrics will you use?
We will use loss and gradient as metrics, and quality of images to evaluate the performance of the model.

Provide a rough schedule for completing the project.
4/8/23: Finish proposal, create GitHub
4/9/23-4/15/23: Download the code and data, implement and optimize the model
4/16/23-4/22/23: Finalized work, Starting report
4/24/23 Submit report