# Final Exam Report

Exam Title: Generative Adversarial Neural.

Course Name: CS 551 – Deep Learning.

Professor's Name: Dr. Alaoui Mhamdi.

University Name: Bishop’s University.

Students:

Johan Stevenson Mogollon Arcos / Student ID: 002359844, Contribution: All code, experiments, and analysis.

Fabio Cozzuto / Student ID: 002214965. Contribution: All code, experiments, and analysis.

## Abstract

## In this report we analyze the results of the final exam, in which we conducted an investigation of generative adversarial networks (GANs) and their different configurations. Two architectures Vanilla\_GAN and CycleGAN were used, both were trained with a dataset of “grumpy cats”, the idea is to compare the performance under two augmentations: basic and deluxe. We analyzed the loss curves of the networks, both for the Generator and the Discriminator, by visual inspection of the generated image at different iterations, in the same way, we evaluated how the various configurations influenced the learning process. The results lead us to interesting conclusions, providing knowledge, ideas and learning about strategies for training GANs for image transformation tasks.

## Introduction

Generative Adversarial Networks (GANs) are an important part in deep learning, in this case, for create images. The main idea is to develop two neural networks that compete like in a game: one player (Generator) tries to create fake images that looks real, and another player (Discriminator) tries to identify what are real or fake. This game allows both players to improve its performance, learns how to create more and more images that it looks like so real.

One thing that we can do with GANs is image-to-image translation, and this is when we want to change an image to make it look like another type or style. CycleGAN is a useful tool to do this, it uses two Generators to go from one style to the other and vice versa and uses something called cycle consistency loss to make sure that if we change an image to another style and then back, it looks like the original.

The quality of the images that GANs produce depends on the training set of images and how we modify them to get more variety. This modification is called data augmentation, and it is nothing more than the different ways in which we can modify the images so that the GANs learn better and help the images they generate to be as good as the initial images.

In this paper, we are going to see what happens when we use two different ways of augmenting the data: a “basic” way and a “deluxe” way that is more elaborate. We will use both GANs, Vanilla\_GAN and CycleGAN to change images of “grumpy cats” from one type to another and compare the results. We will see how these the augmentations affect how the networks learn and how realistic the images they create. The idea is to better understand how GANs works and acquire knowledge in this field.

## Part 1: Deep Convolutional GAN

* **Implement Data Augmentation [10 points]**

.

* **Implement the Discriminator of the DCGAN [10 points]**

.

* **Generator [10 points]**

.

* **Training Loop [10 points]**
* **Experiment [10 points]**

.

## Part 2: CycleGAN

* **Data Augmentation**
* **Generator [20 points]**
  + Encoding Stage
  + Transformation Stage
  + Decoding Stage
* **CycleGAN Training Loop [20 points]**

We use L1

* **CycleGAN Experiments [15 points]**

.

**Can you account for these differences?**

We see that when we use cycle consistency, the numbers that tell us how the networks are learning (the loss curves) appear more stable, they do not change so much. This could be because consistency loss acts as a rule that tells the networks not to just create fake images that "fool" the other network, but also to make sure that if you change an image and then change it again, it looks like the original. This rule helps the training to be more stable and helps the networks learn in a more organized way, rather than in a disorganized way, trying to "fool" each other all the time. The generator also seems to learn better when we use this rule, perhaps because it must follow this extra condition.

**Provide explanations as to why there might or might not be a noticeable difference between the two sets of results.**

Analyzing the images, the reason we don't see a big difference in the images in the end is because the two types of “grumpy cats” we are using are not that different in style. If the change we want to make isn't too difficult, maybe the network can learn to do it right even without the cycle consistency rule. Also, the rule has a weight (lambda), and if we don't adjust it correctly, it may not help, additionally in this case we use 𝐿1 loss, as the original paper suggested. Perhaps if we trained longer or used larger networks, or if the difference between cat styles were more noticeable, we would see a significant improvement in the images when we use the cycle coherence rule. Which indicates, that the rule helps the learning process to be more stable, however, for the images to look much better in this case, we could apply further augmentations or we need to train for a longer time.

## Improvement or Further Experimentation

.

## Conclusions

.