Rochester Institute of Technology

RIT Scholar Works

Theses

11-30-2018

Embedded CycleGAN for Shape-Agnostic Image-to-Image Translation

Ram Longman
rl9416@rit.edu

Follow this and additional works at: <https://scholarworks.rit.edu/theses>

Recommended Citation

Longman, Ram, "Embedded CycleGAN for Shape-Agnostic Image-to-Image Translation" (2018). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.

Embedded CycleGAN for Shape-Agnostic Image-to-Image Translation

Ram Longman
November 30, 2018

A Thesis Submitted
in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in
Computer Engineering

Committee Approval:

Dr. Raymond Ptucha <i>Primary Advisor- R.I.T Dept. of Computer Engineering</i>	Date
---	------

Dr. Amlan Ganguly <i>Secondary Advisor- R.I.T Dept. of Computer Engineering</i>	Date
--	------

Dr. Andres Kwasinski <i>Secondary Advisor- R.I.T Dept. of Computer Engineering</i>	Date
---	------

Department of Computer Engineering

Acknowledgements

I would like to thank my advisor Dr. Raymond Ptucha for his dedicated support and guidance. His patient mentorship carried me through my final years at RIT and helped me grow academically, professionally, and personally.

I would like to thank my committee members, Dr. Amlan Ganguly and Dr. Andres Kwasinski, for being a part of this thesis.

I would like to thank all my professors, teachers, and TAs throughout my academic pursuit for teaching me, challenging me, and helping me grow.

I would like to thank all my classmates for being a part of this journey with me. It wouldn't have been the same without them.

I would like to thank my family for the immense and unconditional support they provided me. They gave me the peace-of-mind and strength I needed to get to where I am today.

Dedication

I would like to dedicate this thesis to my parents. To my father that supported me, taught me the value of education, and guided me to where I am today. To my mom that raised me, taught me the value of love, and was always there for me. Without them none of this would have been possible.

Abstract

Image-to-Image translation is the task of translating images between domains while maintaining the identity of the images. The task can be used for entertainment purposes and applications, data augmentation, semantic image segmentation, and more. Generative Adversarial Networks (GANs), and in particular Conditional GANs have recently shown incredible success in image-to-image translation and semantic manipulation. However, such methods require paired data, meaning that an image must have ground-truth translations across domains. Cycle-consistent GANs solve this problem by using unpaired data. Such methods work well for translations that involve color and texture changes but fail when shape changes are required. This research analyzes the trade-offs between the cycle-consistency importance and the necessary shape changes required for natural looking imagery. The research proposes simple architectural and loss changes to maintain cycle-consistency strength while allowing the model to perform shape changes as required. The results demonstrate improved translations between domains that require shape changes while preserving performance between domains that don't require shape changes.

Contents

Signature Sheet	i
Acknowledgements	ii
Dedication	iii
Abstract	iv
List of Figures	vii
Chapter 1: Introduction	1
1.1 Motivation	1
1.2 Outline	2
1.3 Related Work	3
Chapter 2: Background	5
2.1 Convolutional Neural Networks	5
2.2 Generative Adversarial Networks	6
2.3 Cycle-Consistent Generative Adversarial Networks	9
Chapter 3: Datasets	15
3.1 Flowers	15
3.2 ImageNet	16
3.3 Cityscapes	17
Chapter 4: Proposed Model and Methodology	18
4.1 Intuition	18
4.2 Loss Function	23
4.3 Implementation	24
Chapter 5: Experiments, Results, and Analysis	26
5.1 Experiments	26
5.1.1 Cycle-Consistency Strength	26
5.1.2 An Embedding Channel	28
5.1.3 Embedding Channel Regularization	28
5.1.4 Embedding Channel Variations	29

5.1.5 Dogs: Pomeranians and Pugs.....	30
5.1.6 Semantic Image Segmentation	30
5.2 Results and Analysis	31
5.2.1 Cycle-Consistency Strength	31
5.2.2 An Embedding Channel	40
5.2.3 Embedding Channel Regularization	47
5.2.4 Embedding Channel Variations.....	52
5.2.5 Dogs: Pomeranians and Pugs.....	63
5.2.6 Semantic Image Segmentation	68
Chapter 6: Conclusion and Future Work	74
6.1 Conclusion.....	74
6.2 Challenges	74
6.3 Future Work	75
Bibliography	77

List of Figures

Figure 1- A typical CNN architecture, from [4].	6
Figure 2- GAN loss function terms, from [30].	7
Figure 3- GAN distribution objective illustration, from [5].	8
Figure 4- Generative Adversarial Network’s workflow in Computer Vision, from [24].	9
Figure 5- CycleGAN training procedure, from [7].	10
Figure 6- DiscoGAN training procedure, from [23].	12
Figure 7- typical discriminator architecture in cycle-consistent GANs, from [29].	13
Figure 8- typical generator architecture in cycle-consistent GANs, from [29].	14
Figure 9- Flowers dataset, from left to right: Daisy, Dandelion, Rose, Sunflower, and Tulip.	15
Figure 10- Sample from the Pug class (left) and from the Pomeranian class (right) of ImageNet.	16
Figure 11- Cityscapes dataset. Real image (left) and semantic label map image (right).	17
Figure 12- General flow of the proposed model.	22
Figure 13- Rose to Daisy sample 1, no cycle-consistency.	32
Figure 14- Rose to Daisy sample 1, low cycle-consistency.	32
Figure 15- Rose to Daisy sample 1, medium cycle-consistency.	33
Figure 16- Rose to Daisy sample 1, high cycle-consistency (CycleGAN).	33
Figure 17- Rose to Daisy sample 2, no cycle-consistency.	34
Figure 18- Rose to Daisy sample 2, low cycle-consistency.	34
Figure 19- Rose to Daisy sample 2, medium cycle-consistency.	35
Figure 20- Rose to Daisy sample 2, high cycle-consistency (CycleGAN).	35
Figure 21- Daisy to Rose sample 1, no cycle-consistency.	36
Figure 22- Daisy to Rose sample 1, low cycle-consistency.	36
Figure 23- Daisy to Rose sample 1, medium cycle-consistency.	37
Figure 24- Daisy to Rose sample 1, high cycle-consistency (CycleGAN).	37
Figure 25- Daisy to Rose sample 2, no cycle-consistency.	38
Figure 26- Daisy to Rose sample 2, low cycle-consistency.	38
Figure 27- Daisy to Rose sample 2, medium cycle-consistency.	39
Figure 28- Daisy to Rose sample 2, high cycle-consistency (CycleGAN).	39
Figure 29- Rose to Daisy sample 1, no cycle-consistency with an embedding channel.	40

Figure 30- Rose to Daisy sample 1, low cycle-consistency with an embedding channel.	41
Figure 31- Rose to Daisy sample 1, medium cycle-consistency with an embedding channel.	41
Figure 32- Rose to Daisy sample 1, high cycle-consistency with an embedding channel.	41
Figure 33- Rose to Daisy sample 2, no cycle-consistency with an embedding channel.	42
Figure 34- Rose to Daisy sample 2, low cycle-consistency with an embedding channel.	43
Figure 35- Rose to Daisy sample 2, medium cycle-consistency with an embedding channel.	43
Figure 36- Rose to Daisy sample 2, high cycle-consistency with an embedding channel.	43
Figure 37- Daisy to Rose sample 1, no cycle-consistency with an embedding channel.	44
Figure 38- Daisy to Rose sample 1, low cycle-consistency with an embedding channel.	44
Figure 39- Daisy to Rose sample 1, medium cycle-consistency with an embedding channel.	44
Figure 40- Daisy to Rose sample 1, high cycle-consistency with an embedding channel.	45
Figure 41- Daisy to Rose sample 2, no cycle-consistency with an embedding channel.	45
Figure 42- Daisy to Rose sample 2, low cycle-consistency with an embedding channel.	46
Figure 43- Daisy to Rose sample 2, medium cycle-consistency with an embedding channel.	46
Figure 44- Daisy to Rose sample 2, high cycle-consistency with an embedding channel.	46
Figure 45- Rose to Daisy sample 1, low regularization.	47
Figure 46- Rose to Daisy sample 1, medium regularization.	48
Figure 47- Rose to Daisy sample 1, high regularization.	48
Figure 48- Rose to Daisy sample 2, low regularization.	49
Figure 49- Rose to Daisy sample 2, medium regularization.	49
Figure 50- Rose to Daisy sample 2, high regularization.	49
Figure 51- Daisy to Rose sample 1, low regularization.	50

Figure 52- Daisy to Rose sample 1, medium regularization.....	50
Figure 53- Daisy to Rose sample 1, high regularization.	50
Figure 54- Daisy to Rose sample 2, low regularization.	51
Figure 55- Daisy to Rose sample 2, medium regularization.....	51
Figure 56- Daisy to Rose sample 2, high regularization.	52
Figure 57- Rose to Daisy sample 1, two embedding channels, low cycle- consistency, and low regularization.	53
Figure 58- Rose to Daisy sample 2, two embedding channels, low cycle- consistency, and low regularization.	53
Figure 59- Daisy to Rose sample 1, two embedding channels, low cycle- consistency, and low regularization.	53
Figure 60- Daisy to Rose sample 2, two embedding channels, low cycle- consistency, and low regularization.	54
Figure 61- Rose to Daisy sample 1, two embedding channels, higher cycle- consistency, and medium regularization.....	55
Figure 62- Rose to Daisy sample 2, two embedding channels, higher cycle- consistency, and medium regularization.....	55
Figure 63- Daisy to Rose sample 1, two embedding channels, higher cycle- consistency, and medium regularization.....	55
Figure 64- Daisy to Rose sample 2, two embedding channels, higher cycle- consistency, and medium regularization.....	56
Figure 65- Rose to Daisy sample 1, downsampled embedding and low cycle- consistency.	57
Figure 66- Rose to Daisy sample 2, downsampled embedding and low cycle- consistency.	57
Figure 67- Daisy to Rose sample 1, downsampled embedding and low cycle- consistency.	58
Figure 68- Daisy to Rose sample 2, downsampled embedding and low cycle- consistency.	58
Figure 69- Rose to Daisy sample 1, downsampled embedding and medium cycle- consistency.	59
Figure 70- Rose to Daisy sample 2, downsampled embedding and medium cycle- consistency.	59
Figure 71- Daisy to Rose sample 1, downsampled embedding and medium cycle- consistency.	60
Figure 72- Daisy to Rose sample 2, downsampled embedding and medium cycle- consistency.	60

Figure 73- Rose to Daisy sample 1, high and low resolution embedding channels.	61
Figure 74- Rose to Daisy sample 2, high and low resolution embedding channels.	61
Figure 75- Daisy to Rose sample 1, high and low resolution embedding channels.	62
Figure 76- Daisy to Rose sample 2, high and low resolution embedding channels.	62
Figure 77- Pomeranian to Pug sample 1 using CycleGAN.	63
Figure 78- Pomeranian to Pug sample 1 using the proposed Embedded CycleGAN.	64
Figure 79- Pomeranian to Pug sample 2 using CycleGAN.	65
Figure 80- Pomeranian to Pug sample 2 using the proposed Embedded CycleGAN.	65
Figure 81- Pomeranian to Pug sample 3 using CycleGAN.	66
Figure 82- Pomeranian to Pug sample 3 using the proposed Embedded CycleGAN.	66
Figure 83- Pug to Pomeranian using CycleGAN.	67
Figure 84- Pug to Pomeranian using the proposed Embedded CycleGAN.	67
Figure 85- Semantic image segmentation, CycleGAN as baseline(high cycle-consistency).	69
Figure 86- Semantic image segmentation, Embedded CycleGAN, low cycle-consistency, low regularization.	69
Figure 87- Semantic image segmentation, Embedded CycleGAN, medium cycle-consistency, high regularization.	70
Figure 88- Semantic label maps to real image sample 1. CycleGAN as baseline (high cycle-consistency).	71
Figure 89- Semantic label maps to real image sample 1. Embedded CycleGAN, low cycle-consistency, low regularization.	71
Figure 90- Semantic label maps to real image sample 1. Embedded CycleGAN, medium cycle-consistency, high regularization.	72
Figure 91- Semantic label maps to real image sample 2. CycleGAN as baseline (high cycle-consistency).	72
Figure 92- Semantic label maps to real image sample 2. Embedded CycleGAN, low cycle-consistency, low regularization.	73
Figure 93- Semantic label maps to real image sample 2. Embedded CycleGAN, medium cycle-consistency, high regularization.	73

Chapter 1: Introduction

1.1 Motivation

Paired Generative Adversarial Network (GAN) models for image-to-image translation, such as conditional GAN models, demonstrate impressive image-to-image translation results. Such methods require that each image in the training set has a ground-truth translation across the domains it was to be translated to. This ground-truth provides supervision in the sense that every translation can be compared to its ground-truth, making the problem a lot easier. However, the need for paired data is extremely limiting and can be very expensive and time consuming to obtain. The ability to translate images between domains without ground-truth, i.e., unpaired data, broadens the scope and applicability of the model, allows utilization of massive amounts of unpaired data, and allows translation between domains that wouldn't be possible otherwise.

Cycle-consistent GANs work to close the gap between paired and unpaired image-to-image translation. Introducing the cycle-consistency loss encourages the model to keep the identity of the image the same while only changing the domain of the image. Such models work extremely well in specific cases where the color/texture per class is relatively consistent and where the shape and environment between the translated classes is very similar. On the other hand, such models fail when significant shape changes are required.

Significant shape changes in unpaired image-to-image translation can be very useful in image editing (you can translate a picture of you standing next to

your car to a natural-looking image of you standing next to “your” sports car) and many other applications.

This thesis proposes a model that is based on and builds on top of CycleGAN, a cycle-consistent generative adversarial network. The introduced model is designed to be able to handle more significant shape changes in unpaired image-to-image translation while maintaining the model’s ability to keep the identity of the images in the process.

1.2 Outline

This thesis document is organized as follows. The first chapter is an introduction to the topic. The chapter discusses the motivation for the research, outlines the structure of the thesis work, and discusses some related work. Chapter 2 provides the background needed to understand the problem and work done in this thesis. The chapter briefly discusses Convolution Neural Networks (CNNs), GANs, and Cycle-Consistent GANs. Chapter 3 introduces the datasets used to train and evaluate the different models discussed in this thesis. Chapter 4 introduces the proposed model and its details. First, the chapter details the intuition behind the design of the model. Second, the chapter discusses the loss function of the model. Last, the chapter provides the implementation details of the proposed model. Chapter 5 introduces the experiments performed to test the discussed theory and proposed model and presents the results and their analysis. Last, chapter 6 serves as a conclusion and summary of the work and discusses potential future work.

1.3 Related Work

Semantic manipulation in images using GANs has been the subject of considerable research. Conditional GANs were introduced in 2014 [8] and demonstrated how feeding class information to both the generator and discriminator can allow the model to generate samples of a desired class. Chen et al. [9] introduced InfoGAN, which uses mutual information between the noise input to the generator as well as a latent representation input to the generator to learn and control semantic information in generated images.

Following the success of CycleGAN [7], Choi et al. [10] introduced StarGAN, a cycle-consistent adversarial network that performs multi-domain image-to-image translation using a single generator and a single discriminator using multiple datasets.

Isola et al. [11], Luc et al. [12], and Zhang et al. [13] experimented with image-to-image translation and semantic segmentation using conditional GANs, adversarial networks, and cycle and shape consistent GANs respectively. Bashkirova et al. [14] explored unsupervised video-to-video translation using cycle consistent adversarial networks.

Multiple groups proposed different models to address the challenges with cycle-consistent GANs. Yang et al. [31] proposed to add an attention network in addition to the generator to help focus image translation to the subject object. Gokaslan et al. [32] introduced a discriminator with dilated convolutions to perform image segmentation instead of just binary classification and thus provide the generator with more robust feedback during training to improve shape deformations and artifacts. Lu et al. [33] proposed an identity-guided conditional CycleGAN to translate low-resolution face images to high-resolution face images.

Ehsan Hosseini-Asl et al. [34] proposed a multi-discriminator CycleGAN model for speech domain adaptation. Their model results in stronger discriminators that focus on small details of the frequency features and a generator capable of generating better domain adapted spectrograms. Hiasa et al. [35] extended CycleGAN by adding a gradient-consistency loss to improve accuracy at the boundaries of MRI and CT image synthesis.

Chapter 2: Background

2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) were first introduced by Yann LeCun in 1998 [6], but were largely ignored until Alex Krizhevsky et al.'s [1] 2012 ImageNet performance revolutionized the world of machine learning and computer vision. A CNN is an artificial neural network that combines repeated convolutions and pooling on input stimuli along with traditional fully connected layers to classify the sample. The filters and fully connected layers are learned simultaneously, yielding an efficient and powerful classifier. The connection between the feature extraction and the classification allows the model to learn the best features it can for the specific classification task. CNNs have improved state-of-the-art results in many fields including images, videos, and audio. Some of the tasks include image classification [1][15][16], face recognition [2][17], and natural language processing [3][18]. A typical CNN architecture is presented in Figure 1. Since 2012, CNN research expanded drastically by proposing and analyzing different network architectures, regularization techniques, and modifications to utilize CNNs for various tasks and data.

Figure 1- A typical CNN architecture, from [4].

2.2 Generative Adversarial Networks

Generative Adversarial Networks (GANs) were first introduced by Goodfellow et al. [5] in 2014. A GAN is a model consisting of two neural networks: a generator, G , whose purpose is to generate “fake” samples that mimic samples from a real dataset, and a discriminator, D , whose purpose is to discriminate between real samples from the dataset and fake samples from the generator. The two networks are trained together in an adversarial manner such that the generator learns to generate better samples and the discriminator learns to discriminate between real and fake samples better. GANs are analogous to the interaction between a counterfeiter and a banker. The counterfeiter, analogous to the generator, attempts to generate fake money that will be accepted by the banker. The banker, analogous to the discriminator, attempts to learn and distinguish between real money (real samples) and fake money generated by the counterfeiter. When repeated comparisons between real and counterfeit samples are presented to the banker, the banker will eventually learn from its mistakes and become quite good at distinguishing between fake money from real money.

Similarly, the generator can learn to generate better fake money by knowing which fake money was able to fool the banker better. The optimization problem is defined using the minimax loss presented in (1).

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim P_{data}(x)} (\log D(x)) + \mathbb{E}_{z \sim P_z(z)} (\log(1 - D(G(z)))) \quad (1)$$

The GAN loss function can be explained by examining the two log functions used in (1). Plots of the log terms used are presented in Figure 2 to help illustrate the objective function.

Figure 2- GAN loss function terms, from [30].

The objective of the discriminator is defined to maximize the two functions presented in Figure 2. When the input to the discriminator is a real image, the objective of the discriminator is defined with the left plot in Figure 2. An output of 1 corresponds to the prediction that an image is real and therefore the discriminator's objective is to maximize $\log D(x)$, where x is a real image. An output of 0 indicates a prediction that an image is fake. Therefore, the objective of the discriminator is defined to maximize the function $\log(1-D(G(z)))$, where z is the noise

input to the generator (right plot). Similarly, the objective of the generator is that the discriminator outputs a 1 due to a fake image input. Thus, the objective of the generator is to minimize $\log(1-D(G(z)))$, the right plot in Figure 2.

In an ideal scenario, in which the model converges, the model should reach Nash equilibrium. The training objective is visualized in Figure 3.

Figure 3- GAN distribution objective illustration, from [5].

As presented in Figure 3, the generator distribution (green, solid line) learns over time to match the real data distribution (black, dotted line) as the discriminator (blue, dashed line) learns to differentiate between the real and generated distribution. The model converges at the point where neither the generator nor the discriminator can improve given the state of the other. With respect to Figure 2, this happens when the generator distribution matches the real distribution.

In computer vision, the real distribution can be thought of as the distribution of the dataset used. The general workflow of GANs in computer vision is demonstrated in Figure 4.

Figure 4- Generative Adversarial Network's workflow in Computer Vision, from [24].

As seen in Figure 4, the generator generates fake images from random noise that are fed into the discriminator while real images are fed into the discriminator as well. The generator's objective is to generate fake images that the discriminator classifies as real. The discriminator's objective is to classify both fake and real images correctly.

2.3 Cycle-Consistent Generative Adversarial Networks

Building on top of the success of Generative Adversarial Networks, Zhu et al. [7] introduced the cycle-consistent adversarial network and demonstrated outstanding image-to-image translation performance. Their model consists of two generators and two discriminators. The generators take an image as an input instead of a noise vector and output an image that is meant to keep the identity of the image the same while only changing the domain of the image (i.e., changing a horse to a zebra, leaving all else the same). The discriminators classify the image as real or fake images to provide the generator with feedback during the desired

change of domain. The input is reconstructed from the generated image, using the other generator, and is compared to the original input to ensure that the identity of the image does not change but only the domain. The cycle-consistency, or reconstruction loss, is important as it encourages the model to not change information that does not need to be changed, and thus maintain the identity of the image better. The cycle-consistency loss tells the model to change only the domain of the image and leave everything else the same. For example, when translating an image of a horse to an image of a zebra the model's objective is to only translate the horse into a zebra. The cycle-consistency loss encourages the model to keep the trees or other background content of the image unchanged. The procedure is illustrated in Figure 5, where G is a generator that translates images from domain X to domain Y , F is a generator that translates images from domain Y to domain X , D_Y is a discriminator for both fake and real images in domain Y , and D_X is a discriminator for both real and fake images in domain X .

Figure 5- CycleGAN training procedure, from [7].

The objective of the CycleGAN model builds on the general GAN objective by adding the cycle consistency loss. L1 loss between the original input and the reconstructed input is used for cycle consistency. The objective is shown in (2).

$$\min_{G,F} \max_{D_X,D_Y} L(G, F, D_X, D_Y) = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X) + \lambda L_{cyc}(G, F) \quad (2)$$

Where L_{GAN} is the original GAN objective from (1) and L_{cyc} is the cycle consistency loss presented in (3).

$$L_{cyc}(G, F) = E_{x \sim P_{data}(x)} [\|F(G(x)) - x\|_1] + E_{y \sim P_{data}(y)} [\|G(F(y)) - y\|_1] \quad (3)$$

In parallel, Kim et al. [23] proposed a very similar model abbreviated DiscoGAN. In their work they propose using two generators, the first to translate images from domain A to domain B and the second to translate images from domain B to domain A , and two discriminators, one for domain A and the other for domain B . Just like CycleGAN, DiscoGAN translates a real image, x_A , from domain A to a fake image, x_{AB} , in domain B using generator G_{AB} . The fake image is then fed to generator G_{BA} to generate the reconstructed input, x_{ABA} . The same process is used starting with a real image in domain B . The process is illustrated in Figure 6.

Figure 6- DiscoGAN training procedure, from [23].

The architecture of the discriminator in a cycle-consistent GAN is very similar to the architecture of a traditional CNN for binary classification. The input to the network is an image and the output is a single prediction between 0 and 1. A prediction of 0 corresponds to 100% confidence that the input image is fake while a prediction of 1 corresponds to 100% confidence that the input image is real. The network consists of convolutional layers, which are often followed by batch normalization and a ReLU (rectified linear unit) activation function. A typical discriminator architecture is presented in Figure 7.

Figure 7- typical discriminator architecture in cycle-consistent GANs, from [29].

As shown in Figure 7, each convolutional layer also downsamples the spatial dimensions by a factor of two while the number of channels increases by a factor of two after the first layer and excluding the last prediction layer. A sigmoid activation is used following the last convolutional layer to reduce the prediction to a number between 0 and 1.

The generator's architecture is a little different because both its input and output are images. Therefore, a typical generator architecture for a cycle-consistent GAN model will include several components. First, a sequence of convolutional layers with batch normalization and a ReLU activation function similar to the architecture of the discriminator. The second part network often consists of a residual block. Finally, the last part of the network consists of a sequence of deconvolutional layers that upsample the sample back to the size of an image. A typical CycleGAN architecture is presented in Figure 8.

Figure 8- typical generator architecture in cycle-consistent GANs, from [29].

The generator architecture presented in Figure 8 demonstrates the three parts of the CycleGAN generator. Each convolution layer in the first part, the same as in the discriminator, downsamples the spatial dimensions by a factor of two while each convolutional layer starting from the second layer increases the number of channels by a factor of two. The third part does the exact opposite. The number of channels is reduced by a factor of two and the spatial dimensions are increased by a factor of two with each deconvolution layer to obtain the original image dimensions.

Chapter 3: Datasets

Three datasets were used to analyze the performance of cycle-consistent GANs and to test the proposed model against different types of images, domains, and distributions.

3.1 Flowers

A flower recognition dataset from Kaggle.com [25] with 4242 labeled images of flowers from 5 different categories. The class image distribution is as follows:

- Daisy- 769 images
- Dandelion- 1055 images
- Rose- 784 images
- Sunflower- 734 images
- Tulip- 984 images

The images contain varying number of flowers in various angles, shapes, textures, and colors. A sample from each class is shown in Figure 9.

Figure 9- Flowers dataset, from left to right: Daisy, Dandelion, Rose, Sunflower, and Tulip.

3.2 ImageNet

ImageNet [26] is one of the largest, if not the largest, labeled image datasets that are publicly available. The dataset is part of the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [28] which played a significant role in recent advances in machine learning, deep learning, and computer vision. The dataset contains a blistering number of 14,197,122 labeled images from 21,841 different classes. The dataset was built in a structure consisting of high-level categories with subcategories.

Two classes from the dataset were used to train and test different models. The classes that were used are two different breeds of small dogs: Pugs and Pomeranians. Each class contains 1300 images. A sample of an image from each class is presented in Figure 10.

Figure 10- Sample from the Pug class (left) and from the Pomeranian class (right) of ImageNet.

3.3 Cityscapes

The Cityscapes dataset [27] is a dataset for semantic image segmentation in autonomous driving. The dataset contains 5000 images with fine annotations and 20000 images with coarse annotations. Annotations include semantic, instance-wise, and dense pixel. Labels are of 30 classes related to the driving environment from the point of view of the car. Images were collected from 50 different cities, in different seasons, are all in daytime, and have good/medium weather conditions. An image from each class (real and semantic label maps) is presented in Figure 11.

Figure 11- Cityscapes dataset. Real image (left) and semantic label map image (right).

Chapter 4: Proposed Model and Methodology

4.1 Intuition

Cycle-consistent GANs utilize the cycle-consistency loss to ensure that the identity of the image remains the same while only the domain of the image changes. Such models work extremely well in specific cases where the color and texture of the used classes are relatively consistent and where the shape and environment between the translated classes are very similar. For example, the translation between horses and zebras and vice versa is the perfect case for such a model. Horses are generally brown, zebras generally have black and white stripes, horses and zebras have very similar shapes, and the background environments of horses and zebras are rather similar. Given those characteristics, the model is able to learn the colors of the classes and given that the shapes and environments of both classes are very similar the model learns mainly to change the colors of the animal to those of the translated class. Similarly, the translation between classes of different textures, each of a consistent texture, and with very similar shapes and environments will work just as well.

However, when significant shape changes are required, these models typically fail. The main reasons why such models fail when significant shape changes are required are listed and explained below:

- 1) **Cycle-consistency loss overpowers Generator loss:** when significant shape changes are required, the model faces a trade-off between keeping its identity and changing its domain successfully. The ability of the fake image to successfully fool the discriminator corresponds to a successful domain change while a good reconstructed image

represents successfully keeping the identity of the image the same. As such, the model should be able to change domains successfully when there is no cycle-consistency loss, given that there are enough training examples and the model is able to learn the structures of the domains. However, such translations involve changes in the identities of the images (the model may change the image completely). As cycle-consistency loss is introduced and is increased in importance, the model keeps the identities of the translated images better. However, as a side-effect, the translated images are limited in the amount that they can change their shapes, up to the maximal point, in which the model no longer changes the images in translation. This happens because it is so much more important for the model to successfully reconstruct the images, which is easiest to do if the image doesn't change at all, than it is to change the domain of the image successfully.

- 2) **Lack of Information:** The only input to the generator is the image to be translated and the only output is the translated, or fake, image. Thus, when the fake image is used as input to the generator to reconstruct the input, all the required information to reconstruct the input must exist in the fake image. If the shape doesn't require changes and the color/texture of translation are known the information in the fake image is enough for a good reconstruction. However, when shape changes are required, if the generator changes the shape of the real image to generate a good fake image then when the fake image is used to reconstruct the input it may have changed too much to the point where it no longer contains enough information

to reconstruct the input. Theoretically, the model may perform a perfectly valid translation from the fake image to the original domain. However, because that translation isn't exactly the input, the model will be penalized for it. For example, changing a horse to a zebra doesn't require shape changes and then when changing the zebra back to a horse, the result will naturally be very close to the original horse. On the other hand, translating a dog to a cat may require some shape changes but when translating the cat back to a dog there are many valid translations that may differ significantly from the original dog. In that case, the fake cat image doesn't contain enough information to reconstruct the input successfully. As a result, the model learns to change the original images in the least amount possible to ensure enough information for reconstruction. When shape changes are required, these minimal changes are insufficient.

- 3) **“Weak” Discriminator:** generally, for a model to be able to translate images successfully to a certain domain, the model must be able to learn the structures, colors, and textures of the domain. If, in generic training of the GAN without cycle consistency, the model can't learn the features of a given class then it won't be able to translate images to this class in training with cycle consistency. This problem corresponds to the discriminator not learning the features of the class properly, which in its turn leads to a “weak” generator as well.
- 4) **Inconsistent dataset or insufficient data:** The model needs to be able to learn the structures, colors, and textures of the classes for it to be able to translate images between classes. If there isn't enough data

corresponding to the complexity of the images in the class the model won't be able to learn the features of the class well enough and as a result it won't be able to translate to images of the destination class. Furthermore, if there is enough data per class but it is inconsistent in terms of shape, color, and texture the model may not be able to learn the features of the class appropriately and again not be able to translate images to images of this class.

The main objective of this thesis is to allow for a reduced cycle-consistency loss while maintaining the ability to reconstruct the input properly and keep the identity of the translated images the same. By doing so, the model has more freedom to change the shapes of the images being translated. This objective is achieved by providing the generator with additional information in the form of an additional image channel. As such, this work attempts to solve failure reasons 1 and 2 from above.

The proposed model adds an additional channel to both the input and the output of the generator. The translated image is fed into the discriminator without the additional channel while the additional channel is used when reconstructing the input. Thus, the model is able to learn a separate channel, which can be thought of as an embedding, that helps in the reconstruction of the input. The idea is to encourage the generator to generate a properly translated image and an embedding of the changes that were made or are required to reconstruct the input. In that case, one could argue that the model will then just memorize the entire structure of the input image and use it to reconstruct the input and as a result miss the entire objective of the cycle-consistency loss and the model may change the

should be done) which outputs a translated image in domain B (orange) along with an encoding channel. The translated image is fed into the discriminator without the encoding channel. Both the translated image and the encoding channel are fed into generator $GBtoA$ to reconstruct the input. A cycle consistency loss is used between the input and reconstructed images (apples) and a regularization term is used on the embedding channel used in reconstruction relative to the zeros mask.

4.2 Loss Function

The objective of the proposed model builds on the objective of CycleGAN as its baseline, which builds on top of the original GAN's objective. The original GAN and CycleGAN objectives are defined in (1) and (2) respectively in sections 2.2 and 2.3. An L1 regularization term, as presented in (4) was added to the CycleGAN objective to obtain the complete loss function of the proposed model shown in (5).

$$L_{reg}(G, F) = ||H(G(x))||_1 + ||H(F(y))||_1 \quad (4)$$

Where H is the embedding channel obtained from an output of a generator, G is generator x to y , F is generator y to x , x is a real image from domain X , and y is a real image from domain Y .

$$\min_{G,F} \max_{D_X,D_Y} L(G, F, D_X, D_Y) = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X) + \lambda_{cyc} L_{cyc}(G, F) + \lambda_{reg} L_{reg}(G, F) \quad (5)$$

The hyperparameters λ_{cyc} and λ_{reg} correspond to the weight, or importance, of the cycle-consistency and embedding channel regularization in the loss. If Multiple embedding channels are used λ_{reg} represents a vector of hyper-parameters, where each hyper-parameter corresponds to a different embedding channel. The importance of both components is critical for the performance of the

model. Various levels of cycle-consistency and regularization were tested and are described in detail in Chapter 5.

4.3 Implementation

The PyTorch CycleGAN and pix2pix model was used as baseline for the model. Additional loss terms 'enc_A' and 'enc_B' were added to the model. The additional loss term 'enc_A' corresponds to the calculated L1 regularization term of the embedding channel of the fake image in domain B when translating a real image from domain A to a fake image in domain B . Similarly, the loss term 'enc_B' corresponds to the calculated L1 regularization term of the embedding channel of the fake image in domain A when translating a real image from domain B to a fake image in domain A .

Additional visual representations 'enc_A1' and 'enc_B1' were added to the cycle_gan_model that correspond to the embedding channel of the translated image when translating an image from domain A to domain B and vice versa respectively. The two generators used in the model were defined with four input and output channels (red, green, blue, and embedding) instead of three (red, green, and blue).

The built-in L1 loss function in the neural network package of PyTorch was used to define the L1 regularization term on the embedding channel. The loss was evaluated against an embedding channel of all zeros and thus resulted as an L1 regularization term.

The forward propagation of images through the generators and calculation of the generator loss function were performed as follows:

- An embedding channel of zeros was concatenated with a real image, 'real_A', from domain *A* and was fed to the generator translating images from domain *A* to domain *B*.
- The output of the generator was split to the embedding channel which was saved as 'enc_A1' and the fake image which was saved as 'fake_B'.
- The embedding channel 'enc_A1' was used to calculate the L1 regularization term 'loss_enc_A'.
- The translated image 'fake_B' was fed into the discriminator to evaluate it as an image in domain *B*. The output of the discriminator was used to calculate the generator loss 'loss_G_A'.
- The concatenated output of 'fake_B' and 'enc_A1' was fed to the generator translating images from domain *B* to domain *A*.
- The output from the generator was split to an embedding channel, which was not used in the model, and a reconstructed image 'rec_A'.
- The cycle-consistency loss, 'loss_cycle_A' was calculated using an L1 loss function between the real image 'real_A' and the reconstructed image 'rec_A'.
- The process repeated itself for the translation of images from domain *B* to domain *A* where the generators were flipped and all the 'A's were swapped for 'B'.
- The loss function of the generator was calculated as a sum of all the individual loss terms ('loss_G_A' + 'loss_G_B' + 'loss_cycle_A' + 'loss_cycle_B' + 'loss_idt_A' + 'loss_idt_B' + 'loss_enc_A' + 'loss_enc_B').

Chapter 5: Experiments, Results, and Analysis

5.1 Experiments

Six experiments were designed and executed to thoroughly analyze the logical foundation of this research, to observe patterns and understand cycle-consistent GANs better, and to test the proposed model:

- **Cycle-Consistency Strength:** test a baseline CycleGAN model with various levels of cycle consistency.
- **An Embedding Channel:** test the CycleGAN model with an added embedding channel.
- **Embedding Channel Regularization:** test the Embedded CycleGAN with various levels of regularization on the embedding channel.
- **Embedding Channel Variations:** test several variations on the embedding channel.
- **Dogs: Pomeranians and Pugs:** test the proposed model on a different dataset with a different distribution.
- **Semantic Image Segmentation:** test the proposed model for the task of semantic image segmentation.

The experiments are described in further detail in sections 5.1.1 – 5.1.6.

5.1.1 Cycle-Consistency Strength

The first experiment was designed to examine how a cycle-consistent GAN responds to varying levels of cycle consistency. The objective was to observe how

the ability of the model to change the shape of its domain object and to maintain its identity change in response to the level of cycle-consistency applied.

The Daisy and Rose classes from the flowers dataset were chosen for this experiment for several reasons:

- The two classes have significantly different shapes.
- Each class has consistent structure and colors throughout.
- Both classes have similar environments.
- The classes had just the right amount of data- enough to learn good models and not too much so it doesn't take too long to train each model.

The Daisy class was split to 748 training samples and 20 test samples while the Rose class was split to 764 training samples and 20 test samples.

The CycleGAN model was trained and tested with four different levels of cycle-consistency:

- **No cycle-consistency:** Test how the model performs without cycle-consistency. See if the model can learn the shapes of the classes and change shapes when there's no cycle consistency restriction.
- **Low cycle-consistency:** Test the model with a low level of cycle-consistency. See the trade-off between shape changing and identity keeping.
- **Medium cycle-consistency:** Test the model with a medium level of cycle-consistency. See the trade-off between shape changing and identity keeping.

- **High cycle-consistency:** The actual CycleGAN model with the level of cycle-consistency used by default in their code. Verify that the model fails when required to change shape in this experiment

5.1.2 An Embedding Channel

The second experiment was designed to see how the model, with varying levels of cycle-consistency, responds to the addition of the embedding channel. For comparison reasons, this experiment was done with the same four levels of cycle-consistency as in the baseline experiment as well as with the same data. This test was performed without any regularization on the embedding channel.

5.1.3 Embedding Channel Regularization

The model with a low cycle-consistency value and a low regularization value from the experiment described in section 5.1.2 was chosen to be used in this experiment. The objective of this experiment was to test how the model responds to varying levels of regularization on the embedding channel. Three levels of regularization were tested to observe the pattern of the results corresponding to different levels of regularization.

- **Low regularization:** Introduce a low level of regularization and test how it affects the model. See if it helps the model generalize and observe its effects on the embedding channel.
- **Medium regularization:** Test how a medium level of regularization affects the translated image, reconstructed image, and the embedding.

- **High regularization:** Finally, observe how a high level of regularization affects the model

For comparison reasons, this experiment was performed with the Rose and Daisy classes in the same manner that was used in the experiments in section 5.1.1 and 5.1.2.

5.1.4 Embedding Channel Variations

The embedding channel was integrated to allow the model to learn additional information required to allow proper reconstruction. The objective of this experiment was to test additional embedding architectures to encode the additional information differently and/or provide a different type of regularization. This experiment consisted of three parts:

- Test the model with two additional embedding channels instead of one. This model was tested with different levels of cycle-consistency and regularization to examine the model's response to those parameters.
- Downsample or perform average pooling on the embedding channel to provide a different type of regularization on the embedding channel to encourage the model to learn more robust encodings to be used for reconstruction.
- Test the model with two embedding channels. One in high resolution (full size) and the other at low resolution (downsampled).

Similarly, for comparison reasons, these experiments were performed with the Rose and Daisy classes from the flowers dataset.

5.1.5 Dogs: Pomeranians and Pugs

For comparison reasons, the first three experiments were performed using the same dataset. The objective of this experiment was to test how the proposed model performs on a different dataset with a different distribution.

The Pomeranian and Pug classes from the ImageNet dataset were chosen for this experiment for several reasons:

- The two classes have significantly different shapes.
- Each class has consistent structure and colors throughout.
- Both classes have similar environments.
- The classes had just the right amount of data- enough to learn good models and not too much so it doesn't take too long to train each model.

Each class contained 1300 images in total. 1280 images were used for training and 20 images were used for testing.

5.1.6 Semantic Image Segmentation

This experiment was designed to test the model on yet another dataset with a different distribution. In this case the model was used for a slightly different task: semantic image segmentation. While semantic image segmentation doesn't require significant shape changes it exhibits a similar problem to the one observed when shape changes are required: the semantic feature maps don't contain the information required to reconstruct the input. Thus, this experiment was designed

to test the proposed model on the CityScapes dataset [27] for the semantic image segmentation task. The model was trained with 2975 images from each class (real images and their corresponding ground-truth semantic label maps) and was tested with 50 images from each class.

5.2 Results and Analysis

The results of the six experiments that were performed are presented and analyzed in this section. Each subsection corresponds to an experiment in the order that they were presented in section 5.1. The CycleGAN model is used as a baseline for the evaluation of the proposed Embedded CycleGAN model in this section.

5.2.1 Cycle-Consistency Strength

First, a baseline was established with the Daisy and Rose classes by training and testing a CycleGAN model with varying levels of cycle-consistency: No cycle-consistency, low cycle-consistency, medium cycle-consistency, and high cycle-consistency (the CycleGAN model uses the high cycle-consistency setting). For each test sample three images are shown: the real image, the fake translated image, and the reconstructed image. A few samples from the test set were chosen to illustrate the patterns of the results. For comparison reasons, each sample is presented four times- one for every model used in the following order: no cycle-consistency, low cycle-consistency, medium cycle-consistency, and high cycle-consistency. The results are shown in Figures 13 to 28 with some observations about each test sample in between.

Figure 13- Rose to Daisy sample 1, no cycle-consistency.

Figure 14- Rose to Daisy sample 1, low cycle-consistency.

Figure 15- Rose to Daisy sample 1, medium cycle-consistency.

Figure 16- Rose to Daisy sample 1, high cycle-consistency (CycleGAN).

A few patterns can be observed from the first sample translating a rose to a daisy:

- As cycle-consistency decreased, the model generated better daisies: clearer daisies in a more proportional size to the rose. The high cycle-consistency model completely failed to generate a daisy while the no cycle-consistency model was able to generate a decent looking daisy.

- As cycle-consistency increased, the model maintained the environment or background of the image better. The no cycle-consistency model changed and blurred the background behind the flower while the high cycle-consistency model kept the background almost unchanged.
- As cycle-consistency increased, the reconstructed image became closer to the real image.

Figure 17- Rose to Daisy sample 2, no cycle-consistency.

Figure 18- Rose to Daisy sample 2, low cycle-consistency.

Figure 19- Rose to Daisy sample 2, medium cycle-consistency.

Figure 20- Rose to Daisy sample 2, high cycle-consistency (CycleGAN).

Similarly, sample 2 demonstrates the same patterns as sample 1: As cycle-consistency decreases the model generates better daisies, as cycle-consistency increases the model keeps the background or environment of the original image better in the fake image, and as cycle-consistency increases the reconstructed image becomes better and better.

Figure 21- Daisy to Rose sample 1, no cycle-consistency.

Figure 22- Daisy to Rose sample 1, low cycle-consistency.

Figure 23- Daisy to Rose sample 1, medium cycle-consistency.

Figure 24- Daisy to Rose sample 1, high cycle-consistency (CycleGAN).

Neither of the models were able to translate daisies into roses as demonstrated by test sample 1 of translating a daisy to a rose. It appears that the models were not able to learn the structure of the rose properly because its structure was more complex, varied more in the data, and the amount of data was insufficient. However, it can still be observed that as cycle-consistency increased the model maintained the background or identity of the image better and as cycle-consistency decreased the model was able to change the daisy slightly more to look

a little more like a rose. Also, as cycle-consistency increased the reconstruction of the original image was better.

Figure 25- Daisy to Rose sample 2, no cycle-consistency.

Figure 26- Daisy to Rose sample 2, low cycle-consistency.

Figure 27- Daisy to Rose sample 2, medium cycle-consistency.

Figure 28- Daisy to Rose sample 2, high cycle-consistency (CycleGAN).

Sample 2 translating daisy to rose is different because it has many flowers in the image. As cycle-consistency increases the model maintains the same number of flowers more accurately as well as keeping the background of the image the same. Again, as cycle-consistency decreased the model changed the daisies to look a bit more like roses but it also blurred the background of the image.

5.2.2 An Embedding Channel

The model was trained and tested using the same image sets that were used in section 5.2.1 after adding the embedding channel to the generator. These models were trained without any regularization on the embedding channel. For comparison reasons, the model was trained with the same four levels of cycle-consistency: no, low, medium, and high cycle-consistency levels. Four images are shown for each test sample: the real image, the fake translated image, the reconstructed image, and the embedding channel. The same test images that were presented in the baseline experiment are presented here. Each sample is presented four times- one for every model used in the following order: no cycle-consistency, low cycle-consistency, medium cycle-consistency, and high cycle-consistency. The results are shown in Figures 29 to 44 with some observations about each test sample in between.

Figure 29- Rose to Daisy sample 1, no cycle-consistency with an embedding channel.

Figure 30- Rose to Daisy sample 1, low cycle-consistency with an embedding channel.

Figure 31- Rose to Daisy sample 1, medium cycle-consistency with an embedding channel.

Figure 32- Rose to Daisy sample 1, high cycle-consistency with an embedding channel.

Several patterns can be observed from the first sample translating a rose to a daisy with the embedding channel and without regularization:

- As cycle-consistency decreased the generated daisy appeared more proportional to the rose in size and more visually correct.
- Generally, as cycle-consistency increased the background or identity of the image was maintained better. However, it seems like the fake image with medium cycle-consistency kept the identity and background of the image better than the fake image with high cycle-consistency.
- Reconstruction of the original image appears very good and close to perfect, at least visually, starting at a low level of cycle-consistency.
- The embedding channel appears to learn the structure of the image.
- The embedding channel appears to be emphasized more with medium cycle-consistency: with low cycle-consistency emphasized embeddings aren't as important and with high cycle-consistency the translated image itself contains more information for reconstruction and therefore an emphasized embedding channel isn't required.

Figure 33- Rose to Daisy sample 2, no cycle-consistency with an embedding channel.

Figure 34- Rose to Daisy sample 2, low cycle-consistency with an embedding channel.

Figure 35- Rose to Daisy sample 2, medium cycle-consistency with an embedding channel.

Figure 36- Rose to Daisy sample 2, high cycle-consistency with an embedding channel.

The second sample translating a rose to a daisy supports the observations from the first sample. In this case, the generated daisies appear visually correct

throughout when only the daisy generated by the model with the high cycle-consistency is a bit cut off on its left side.

Figure 37- Daisy to Rose sample 1, no cycle-consistency with an embedding channel.

Figure 38- Daisy to Rose sample 1, low cycle-consistency with an embedding channel.

Figure 39- Daisy to Rose sample 1, medium cycle-consistency with an embedding channel.

Figure 40- Daisy to Rose sample 1, high cycle-consistency with an embedding channel.

As was observed in the baseline experiments, it appears the model doesn't learn the structure of the rose properly. As such, in this case the translations from a daisy to a rose aren't very good. Like the rose to daisy translations, the reconstruction appears close to perfect starting at a low cycle-consistency. This may indicate that too much information is learned by the embedding channel. Again, the most emphasized embeddings appear to be with a medium level of cycle-consistency. In this case it appears the background or identity of the image is kept the better as cycle-consistency increased.

Figure 41- Daisy to Rose sample 2, no cycle-consistency with an embedding channel.

Figure 42- Daisy to Rose sample 2, low cycle-consistency with an embedding channel.

Figure 43- Daisy to Rose sample 2, medium cycle-consistency with an embedding channel.

Figure 44- Daisy to Rose sample 2, high cycle-consistency with an embedding channel.

The same observations from the previous samples are emphasized with this sample. In this case, again, the reconstruction appears almost perfect starting at a low level of cycle-consistency. This is despite the image having many flowers.

5.2.3 Embedding Channel Regularization

In this experiment the model was trained and tested using the same image sets that were used in sections 5.2.1 and 5.2.2 after adding various levels of regularization to the embedding channel. The model was trained with a low level of cycle-consistency and three different levels of regularization: low, medium, and high. Four images are shown for each test sample: the real image, the fake translated image, the reconstructed image, and the embedding channel. The same test images that were presented in the previous experiments are presented here. Each sample is presented three times- one for every model used in the following order: low regularization, medium regularization, and high regularization. The results are shown in Figures 45 to 56 with some observations about each test sample in between.

Figure 45- Rose to Daisy sample 1, low regularization.

Figure 46- Rose to Daisy sample 1, medium regularization.

Figure 47- Rose to Daisy sample 1, high regularization.

As the regularization on the embedding layer increases several patterns can be observed. The most obvious observations are that the structure learned by the embedding channel becomes less emphasized as regularization increases and that the reconstruction of the original image becomes worst as regularization increases. It appears in sample 1 translating a rose to a daisy that the background or environment of the image changes more significantly as a response to change in regularization while the fake flower itself doesn't change as much.

Figure 48- Rose to Daisy sample 2, low regularization.

Figure 49- Rose to Daisy sample 2, medium regularization.

Figure 50- Rose to Daisy sample 2, high regularization.

Again, it can be observed from sample 2 translating a rose to a daisy that the embedding channel became less emphasized and that the reconstruction became

worst as regularization increased. It appears that the background changed more significantly than the flower itself in response to regularization.

Figure 51- Daisy to Rose sample 1, low regularization.

Figure 52- Daisy to Rose sample 1, medium regularization.

Figure 53- Daisy to Rose sample 1, high regularization.

The first sample translating a daisy to a rose supports the main observations that the embedding channel becomes less emphasized and that the reconstruction gets worst as the regularization on the embedding channel increases. Interestingly, the most emphasized component in the embedding channel is the flower and as regularization increases the flower is the only visible component left. This indicates that, when the embedding channel is regularized, the model learns only the important information it needs to reconstruct the input given the changes that were made to generate a good fake image.

Figure 54- Daisy to Rose sample 2, low regularization.

Figure 55- Daisy to Rose sample 2, medium regularization.

Figure 56- Daisy to Rose sample 2, high regularization.

The same observations from sample 1 of translating a daisy to a rose can be observed with sample 2. In this case with many flowers instead of one. Again, as regularization increased the model was forced to focus its attention on learning only the structure of the flower (what was changed) and not the entire image in the embedding channel.

5.2.4 Embedding Channel Variations

The first variation of the embedding channel was to use two embedding channels. The idea was to observe what is learned when additional embedding channels are added and if the model can learn different types of information in the different embedding channels. Two models were trained: one with a low level of cycle-consistency and a low level of regularization and the other with a slightly higher level of cycle-consistency and medium level of regularization on both embedding channels. Four samples from the results of the first model are presented in Figures 57 to 60 and are followed by an explanation. Similarly, four samples from the results of the second model are presented in Figures 61 to 64 and are followed by observations and analysis.

Figure 57- Rose to Daisy sample 1, two embedding channels, low cycle-consistency, and low regularization.

Figure 58- Rose to Daisy sample 2, two embedding channels, low cycle-consistency, and low regularization.

Figure 59- Daisy to Rose sample 1, two embedding channels, low cycle-consistency, and low regularization.

Figure 60- Daisy to Rose sample 2, two embedding channels, low cycle-consistency, and low regularization.

Several observations can be made from the results of the model with two embedding channels, low cycle-consistency, and low regularization:

- Both embedding channels learn structural information of the image.
- It appears that the embedding channels learn opposing colors (in one embedding channel white appears as black and in the other it appears as white).
- One embedding channel appears more emphasized (enc_A2) and its edges are more pronounced.
- The reconstruction is even better than the reconstruction of the same model with only one embedding channel. That is due to the additional embedding channel.
- Fake images are generally good in comparison to other models. The fake images are somewhat similar to those of the same model with only one embedding channel.

Figure 61- Rose to Daisy sample 1, two embedding channels, higher cycle-consistency, and medium regularization.

Figure 62- Rose to Daisy sample 2, two embedding channels, higher cycle-consistency, and medium regularization.

Figure 63- Daisy to Rose sample 1, two embedding channels, higher cycle-consistency, and medium regularization.

Figure 64- Daisy to Rose sample 2, two embedding channels, higher cycle-consistency, and medium regularization.

The results of the second model with two embedding channels are a little different. This model was trained with a higher level of cycle-consistency and medium levels of regularization on both embedding channels. One embedding channel learned a more emphasized structure while structure is barely visible in the second embedding channel. The embedding channels are more focused on the flowers due to the regularization. The reconstruction images using this model are very good and appear close to perfect. That is due to the higher cycle-consistency. The fake images are slightly worse than the first model with two embedding channels due to the higher cycle-consistency. Yet, the fake images are decent and are significantly better than the established baseline.

The second variation of the embedding channel was a downsampled embedding channel. The output embedding channel obtained from the generator was downsampled by a factor of two using 2d average pooling. The idea is that the downsampling will serve as regularization on the embedding channel. As such, the models that were trained in this variation were trained without regularization on the embedding channel. Two models were trained, one with a low level of cycle-consistency and the other with a medium level of cycle-consistency. Following the

same format of the previous variation, four test samples from the first model with a low cycle-consistency value are presented in Figures 65 to 68 and are followed by an analysis. Four test samples from the second model with a medium cycle-consistency value are presented in Figures 69 to 72 and are followed by an analysis.

Figure 65- Rose to Daisy sample 1, downsampled embedding and low cycle-consistency.

Figure 66- Rose to Daisy sample 2, downsampled embedding and low cycle-consistency.

Figure 67- Daisy to Rose sample 1, downsampled embedding and low cycle-consistency.

Figure 68- Daisy to Rose sample 2, downsampled embedding and low cycle-consistency.

Several observations can be made from the results of the model with a low level of cycle-consistency and a downsampled embedding channel:

- Comparing to previous results, the generated fake image is not great. However, the model is able to change domains successfully.
- The reconstructed image is not good. The model barely changed the image from fake to reconstructed.
- The results are very similar to the results obtained with high levels of regularization on the embedding channel. It appears the downsampling does serve as regularization. The regularization effect is too strong in this case.

- Interestingly, the embedding channel when translating from a rose to a daisy is black and when translating from a daisy to a rose it is white. This is true across all the results.

Figure 69- Rose to Daisy sample 1, downsampled embedding and medium cycle-consistency.

Figure 70- Rose to Daisy sample 2, downsampled embedding and medium cycle-consistency.

Figure 71- Daisy to Rose sample 1, downsampled embedding and medium cycle-consistency.

Figure 72- Daisy to Rose sample 2, downsampled embedding and medium cycle-consistency.

The results of the second model with a medium level of regularization are slightly different from the results obtained with a low level of regularization. The fake images obtained with this model don't change shape as well and the reconstructed images are a little better. Both are due to the increased level of cycle-consistency. Interestingly, with this medium level of cycle-consistency, the flowers in the embedding channel have a strong black color. The rest of the image is mostly gray and un-emphasized. Unintentionally, the embedding channel can almost be used for object segmentation. This holds true better for translation from daisies to roses than it does for the translation from roses to daisies.

The third variation on the embedding channel was a combination of two embedding channels. One at high-resolution (full size) and the other at low resolution (downsampled). This is a combination of the previous two variations. The idea was to encourage the model to learn two embedding channels at different frequencies to allow learning of different levels of abstraction. A low level of regularization was applied only on the high resolution embedding channel while no regularization was applied on the low resolution embedding channel. The model was trained with a low level of cycle-consistency. Four test samples are presented in Figures 73 to 76 and are followed by an analysis.

Figure 73- Rose to Daisy sample 1, high- and low-resolution embedding channels.

Figure 74- Rose to Daisy sample 2, high- and low-resolution embedding channels.

Figure 75- Daisy to Rose sample 1, high- and low-resolution embedding channels.

Figure 76- Daisy to Rose sample 2, high- and low-resolution embedding channels.

Several observations can be made from the results of the model with a high- and low-resolution embedding channels:

- Comparing to previous results, the generated fake image is pretty good. There is less blurriness in most of the results and the generated flowers appear rather realistic.
- The reconstructed images are good. There is a slight blurriness when compared to the original, but the original image is clearly reconstructed with good detail.
- Significant differences appear in the low-resolution embedding channel when translating from a rose to a daisy or from a daisy to a rose. When translating from a rose to a daisy the low-resolution embedding appears

mostly bright with only the flowers emphasized. When translating a daisy to a rose the low-resolution embedding channel appears mostly dark with some structure in it.

The high-resolution embedding channel appeared to learn similar information learned when only a single embedding channel was used.

5.2.5 Dogs: Pomeranians and Pugs

The Pomeranian and Pug classes from the ImageNet dataset were used to evaluate the model on different data with a different distribution. Two models were trained and evaluated: the CycleGAN model and the proposed embedded CycleGAN model with one embedding channel, low cycle-consistency, and low regularization. Each model was evaluated using four test images. The results are presented in Figures 73 to 80 and are used to compare the CycleGAN model with the proposed embedded CycleGAN model. Each test image is presented first using the CycleGAN model and then using the proposed embedded CycleGAN model.

Figure 77- Pomeranian to Pug sample 1 using CycleGAN.

Figure 78- Pomeranian to Pug sample 1 using the proposed Embedded CycleGAN.

A few observations can be made from the first sample translating an image of a Pomeranian to an image of a Pug:

- The CycleGAN model didn't change the shape of the original image. The proposed Embedded CycleGAN did change the shape of the face of the dog. Many features that were changed demonstrate the features of the Pug better. However, the resulting Pug in the fake image is not completely realistic and appears a bit twisted.
- Both models maintain the identity of the image well. The CycleGAN model maintains very small details better than the proposed Embedded CycleGAN.
- Both reconstructed images appear equally as good.

Figure 79- Pomeranian to Pug sample 2 using CycleGAN.

Figure 80- Pomeranian to Pug sample 2 using the proposed Embedded CycleGAN.

The same observations that were made about the first sample translating an image of a Pomeranian to an image of a pug hold two in the second sample.

Figure 81- Pomeranian to Pug sample 3 using CycleGAN.

Figure 82- Pomeranian to Pug sample 3 using the proposed Embedded CycleGAN.

The third test sample translating an image of a Pomeranian to an image of a Pug is a little different. The Pomeranian in this image has long hair extending from his neck and body. Both models attempt to merge this area with the environment for the image to appear natural to the discriminator. The result is a blur in the texture and colors of the floor. Here again, the CycleGAN model doesn't change the shape of the image while the embedded CycleGAN does. In this case the face of the generated Pug appears rather realistic. In this test case the image generated by

CycleGAN is significantly blurrier than the image generated by the proposed Embedded CycleGAN. The identity of the image is maintained about equally as well using both models. Both reconstructed images are good and appear very similar.

Figure 83- Pug to Pomeranian using CycleGAN.

Figure 84- Pug to Pomeranian using the proposed Embedded CycleGAN.

Both models failed translating images of Pugs to images of Pomeranians. The Pomeranian training set contained more variation and it appears that with the size of the training set the models were unable to successfully learn the appearance of

images from this class. The models were only able to learn the color of the class, which is a light brown/yellow color. One test sample is shown in Figures 79 and 80. The CycleGAN model didn't change the shape of the image at all and only changed the image to contain more brown/yellow color. The Embedded CycleGAN model did change the shape of the dog in the image just a little. However, the result is far from looking like a Pomeranian. The proposed Embedded CycleGAN model also changed the color of the dog to the most common color of a Pomeranian-brown/yellow. The CycleGAN model maintained the identity of the image better than the proposed model.

5.2.6 Semantic Image Segmentation

The Cityscapes dataset was used to test the model for semantic image segmentation in the driving scene from the point-of-view of the driver. For comparison reasons, a baseline CycleGAN model was trained and evaluated. Two versions of the proposed embedded CycleGAN model were also trained and evaluated: one with a low level of cycle-consistency and a low level of regularization and the other with a medium level of cycle-consistency and a high level of regularization. Both proposed models experienced mode collapse translating a real image to semantic label maps. Early on in training, the models converged to a noisy and bad translation from which they weren't able to progress. However, the models did much better generating real images from semantic label maps. A test sample from each model of semantic image segmentation are shown in Figures 81 to 83 and are followed by several observations. Two test samples, for each model,

generating real images from semantic label maps are presented in Figures 84 to 89 and are followed by an analysis.

Figure 85- Semantic image segmentation, CycleGAN as baseline(high cycle-consistency).

Figure 86- Semantic image segmentation, Embedded CycleGAN, low cycle-consistency, low regularization.

Figure 87- Semantic image segmentation, Embedded CycleGAN, medium cycle-consistency, high regularization.

The fake semantic maps generated by the Embedded CycleGAN models failed due to mode collapse as can be seen in Figures 82 and 83. The CycleGAN model was able to avoid mode collapse and generate a decent looking semantic label map image. The reconstruction of the model with medium cycle-consistency and high regularization appears best, while all reconstructions are good. As expected, the embedding channel with high regularization appears empty while the embedding channel of the model with a low regularization level captures the structure of the image in a visible way.

Figure 88- Semantic label maps to real image sample 1. CycleGAN as baseline (high cycle-consistency).

Figure 89- Semantic label maps to real image sample 1. Embedded CycleGAN, low cycle-consistency, low regularization.

Figure 90- Semantic label maps to real image sample 1. Embedded CycleGAN, medium cycle-consistency, high regularization.

All models were able to generate decent looking images from the semantic label maps. All images appear similar in terms of quality. Most of the details are correct and are captured in the synthesized image. However, some small details are blurry and unclear. The CycleGAN model seems to confuse buildings with trees.

Figure 91- Semantic label maps to real image sample 2. CycleGAN as baseline (high cycle-consistency).

Figure 92- Semantic label maps to real image sample 2. Embedded CycleGAN, low cycle-consistency, low regularization.

Figure 93- Semantic label maps to real image sample 2. Embedded CycleGAN, medium cycle-consistency, high regularization.

The same observations made from the first test sample translating semantic label maps into real images can be made about the second test sample. All synthesized images appear similar in terms of quality. Most of the big details are correct while some of the small details may appear blurry or unclear. Again, the CycleGAN model appears to confuse trees with buildings.

Chapter 6: Conclusion and Future Work

6.1 Conclusion

Unpaired image-to-image translation is a difficult task and the gap between unpaired and paired methods remains large. The introduction of the cycle-consistent GAN helped close the gap and served as a gateway towards further research and significant improvements in unpaired image-to-image translation. However, such models present major challenges when shape changes are required in translation between domains. This work provides a deep analysis into the reasons cycle-consistent GANs fail when shape changes are required. Such models face a direct trade-off between the ability of the model to change the shape of the object in the image to the ability of the model to maintain the identity of the image with good detail. This trade-off is analyzed and examined through a series of experiments. This work attempts to extend the performance of cycle-consistent GANs to a wider range of data that requires shape changes by proposing several simple architectural and loss changes. Through a series of experiments, it was shown that the proposed model can handle more significant shape changes while maintaining the identity of the original images in the translated images with fine detail.

6.2 Challenges

GANs, cycle-consistent GANs, and the proposed embedded CycleGAN all face some significant challenges. First, due to the adversarial nature of the models, all

GAN based models face a convergence difficulty. The loss function contains competing terms that need to be balanced. In the original GAN, it's the loss of the generator against the loss of the discriminator. In cycle-consistent GANs the cycle-consistency loss is added and needs to be balanced against the generator loss in addition to the balance that needs to be between the generator and discriminator. In the proposed embedded CycleGAN another loss term is added- a regularization on the embedding channel. Theoretically, the idea is that the model will learn only information that's needed, where needed is defined as if the benefit from maintaining the information is greater than the penalty applied for keeping the information. Thus, a careful balance needs to be maintained between the regularization term, the cycle-consistency term, and the generator loss term. Such trade-offs between loss terms in neural networks presents difficult optimization and convergence problems. Hyperparameter fine-tuning becomes crucial and difficult in this case.

6.3 Future Work

The model proposed by this work presents promising potential in shape-agnostic image-to-image translation. However, this thesis barely scratches the surface of the variations and extensions that can be made based on this model. Different ways to encourage the embedding channel to learn only what's necessary for the shape changes it needs to make should be explored. The ability to do so successfully will yield, theoretically, better results. Another route of exploration is to separate the embedding channel into high and low frequency channels that are extracted and inserted in different locations in the middle of the network. Such a

model may encourage learning of different frequency components in different channels and those can be utilized in different ways for reconstruction or otherwise. Different attention models could potentially be used to allow for a more focused embedding channel.

Bibliography

- [1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks.", Advances in neural information processing systems, 2012.
- [2] F. Schroff, D. Kalenichenko, and J. Philbin. "FaceNet: A Unified Embedding for Face Recognition and Clustering". arXiv preprint arXiv: 1503.03832v3, 2015.
- [3] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schutze. "Comparative Study of CNN and RNN for Natural Language Processing". arXiv: 1702.01923v1 [cs.CL] 7 Feb 2017.
- [4] Saleh Albelwi and Ausif Mahmood. "A Framework for Designing the Architectures of Deep Convolutional Neural Networks". Entropy 2017, 19(6), 242; doi:10.3390/e19060242.
- [5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. "Generative adversarial nets". In NIPS 2014, pp. 2672-2680.
- [6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-Based Learning Applied to Document Recognition". Proc. Of the IEEE, November 1998.
- [7] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks". arXiv: 1703.10593v4 [cs.CV] 19 Feb 2018.
- [8] Mehdi Mirza and Simon Osindero. "Conditional Generative Adversarial Nets". arXiv: 1411.1784v1 [cs.LG] 6 Nov 2014.
- [9] X. Chen, Y. Duan, R. Houthoofd, J. Schulman, I. Sutskever, and P. Abbeel. "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets". arXiv: 1606.03657v1 [cs.LG] 12 Jun 2016.

- [10] Y. Choi, M. Choi, M. Kim, J.W. Ha, S. Kim, and J. Choo. “StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation”. In CVPR 2018.
- [11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. “Image-to-Image Translation with Conditional Adversarial Networks”. arXiv: 1611.07004v2 [cs.CV] 22 Nov 2017.
- [12] P. Luc, C. Couprie, S. Chintala, and J. Verbeek. “Semantic Segmentation using Adversarial Networks”. arXiv: 1611.08408v1 [cs.CV] 25 Nov 2016.
- [13] Z. Zhang, L. Yang, and Y. Zheng. “Translating and Segmenting Multimodal Medical Volumes with Cycle- and Shape-Consistency Generative Adversarial Network”. In CVPR 2018.
- [14] D. Bashkirova, B. Usman, and K. Saenko. “Unsupervised Video-to-Video Translation”. arXiv: 1806.03698v1 [cs.CV] 10 Jun 2018.
- [15] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In ICLR 2015.
- [16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. “Going Deeper with Convolutions”. arXiv: 1409.4842v1 [cs.CV] 17 Sep 2014.
- [17] P. Kamencay, M. Benco, T. Mizdos, and R. Radil. “A New Method for Face Recognition Using Convolutional Neural Network”. Digital Image Processing and Computer Graphics. Volume 15. Number 4. 2017. Special Issue.
- [18] A. Conneau, H. Schwenk, Y. Le Cun, and L. Barrault. “Very Deep Convolutional Networks for Text Classification”. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 1107–1116.

- [19] R. Girshick, J. Donahue, T. Darrell, and J. Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation”. arXiv:1311.2524v5 [cs.CV] 22 Oct 2014.
- [20] S. Ren, K. He, R. Girshick, and J. Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In NIPS 2015.
- [21] K. He, G. Gkioxari, P. Dollar, and R. Girshick. “Mask R-CNN”. arXiv:1703.06870v3 [cs.CV] 24 Jan 2018.
- [22] J. Long, E. Shelhamer, and T. Darrell. “Fully Convolutional Networks for Semantic Segmentation”. In CVPR 2015.
- [23] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. “Learning to Discover Cross-Domain Relations with Generative Adversarial Networks”. arXiv:1703.05192v2 [cs.CV] 15 May 2017.
- [24] SkyMind.ai Wiki Blog post. “A Beginner’s Guide to Generative Adversarial Networks (GANs)”.
- [25] A. Mamaev. “Flower Recognition”. Kaggle Dataset. Kaggle.com
- [26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. “ImageNet: A Large-Scale Hierarchical Image Database”. CVPR 2009.
- [27] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The Cityscapes Dataset for Semantic Urban Scene Understanding,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [28] O. Russakovsky*, J. Deng*, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei (* = equal contribution) “ImageNet Large Scale Visual Recognition Challenge”. IJCV, 2015.

- [29] P. Vicol. "Programming Assignment 4: CycleGAN". CSC321 Winter 2018. University of Toronto.
- [30] R. Ptucha. "Generative Adversarial Networks (GANs). CMPE 789- Special Topics: Deep Learning." April 2018. Rochester Institute of Technology.
- [31] C. Yang, T. Kim, R. Wang, H. Peng, and C.-C. J. Kuo. "Show, Attend, and Translate: Unsupervised Image Translation with Self-Regularization and Attention". arXiv:1806.06195v1 [cs.CV] 16 June 2018.
- [32] A. Gokaslan, V. Ramanujan, D. Ritchie, K. I. Kim, and J. Tompkin. "Improving Shape Deformation in Unsupervised Image-to-Image Translation". arXiv:1808.04325v1 [cs.CV] 13 Aug 2018.
- [33] Y. Lu, Y.W. Tai, and C.K. Tang. "Attribute-Guided Face Generation Using Conditional CycleGAN". arXiv:1705.09966v2 [cs.CV] 14 Nov 2018.
- [34] E. Hosseini-Asl, Y. Zhou, C. Xiong, and R. Socher. "A Multi-Discriminator CycleGAN for Unsupervised Non-Parallel Speech Domain Adaptation". arXiv:1804.00522v2 [cs.CL] 3 Apr 2018.
- [35] Y. Hiasa, Y. Otake, M. Takao, T. Matsouka, K. Takashima, A. Carass, J. L. Prince, N. Sugano, and Y. Sato. "Cross-modality image synthesis from unpaired data using CycleGAN". arXiv:1803.06629v3 [cs.CV] 31 Jul 2018.