Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Δ.Π.Μ.Σ. Επιστημη Δεδομενων και Μηχανικη Μαθηση

Unsupervised Translation of Grand Theft Auto V Images to Real Urban Scenes

Μεταπτυχιακη Διπλωματικη Εργασια

του

ΕΥΑΓΓΕΛΟΥ ΤΣΟΓΚΑ

Επιβλέποντες: Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Γεώργιος Σιόλας
Εργαστηριακό και Διδακτικό Προσωπικό Ε.Μ.Π.

Εργαστηριο Τεχνητης Νοημοσυνης και Συστηματων Μαθησης
Αθήνα, Ιανουάριος 2022

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Δ.Π.Μ.Σ. Επιστήμη Δεδομένων και Μηχανική Μάθηση
Εργαστήριο Τεχνητής Νοημοσύνης και Συστημάτων Μάθησης

Unsupervised Translation of Grand Theft Auto V Images to Real Urban Scenes

Μεταπτυχιακη Διπλωματικη Εργασια

του

ΕΥΑΓΓΕΛΟΥ ΤΣΟΓΚΑ

Επιβλέποντες: Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Γεώργιος Σιόλας
Εργαστηριακό και Διδακτικό Προσωπικό Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 19η Ιανουαρίου 2022.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
....	....	....
Ανδρέας-Γεώργιος Σταφυλοπάτης	Στέφανος Κόλλιας	Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Ιανουάριος 2022

(Υπογραφή)

....

Ευάγγελος Τσόγκας

Διπλωματούχος του Μεταπτυχιακού Προγράμματος Επιστήμης Δεδομένων και Μηχανικής Μάθησης Ε.Μ.Π.

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Δ.Π.Μ.Σ. Επιστήμη Δεδομένων και Μηχανική Μάθηση
Εργαστήριο Τεχνητής Νοημοσύνης και Συστημάτων Μάθησης

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Τα τελευταία χρόνια, η Βαθιά Μάθηση έχει αναπτυχθεί ραγδαία και έχει συμβάλει σημαντικά στην εξέλιξη της Όρασης Υπολογιστών. Ομοίως, από την εμφάνιση των Παραγωγικών Αντιπαραθετικών Δικτύων (Generative Adversarial Networks, GANs), ο τομέας της Παραγωγικής Τεχνητής Νοημοσύνης έχει υποστεί ριζικές αλλαγές. Τα GANs είναι μια οικογένεια μοντέλων που μπορούν να μαθαίνουν μοτίβα από υπάρχοντα δεδομένα, όπως εικόνες ή κείμενο, και στη συνέχεια να παράγουν νέο περιεχόμενο με εντυπωσιακά αποτελέσματα. Η αποτελεσματικότητα των GANs έχει προκαλέσει μεγάλο ενδιαφέρον δίνοντας αφορμή για πολλές νέες προσεγγίσεις και εφαρμογές, καθώς όλο και περισσότερη έρευνα αφιερώνεται γύρω από αυτά. Ένα τέτοιο ερευνητικό θέμα είναι η μετάφραση από εικόνα σε εικόνα, το αντικείμενο του μετασχηματισμού εικόνων από ένα πεδίο έτσι ώστε να έχουν το ύφος ή τα χαρακτηριστικά εικόνων ενός άλλου πεδίου. Η μετάφραση από εικόνα σε εικόνα μπορεί να εφαρμοστεί για τη μεταφορά καλλιτεχνικού ύφους (π.χ. για τη μετατροπή μιας φωτογραφίας ώστε να μοιάζει με τον πίνακα ενός διάσημου ζωγράφου) ή ακόμη και για τη γεφύρωση του χάσματος μεταξύ συνθετικών και πραγματικών εικόνων. Αυτή η διατριβή επικεντρώνεται στο τελευταίο και αποσκοπεί στη μετατροπή εικόνων του παιχνιδιού Grand Theft Auto V (GTA V) ώστε να μοιάζουν με ρεαλιστικές εικόνες αστικών περιοχών, εφαρμόζοντας σύγχρονες μεθόδους μετάφρασης εικόνας σε εικόνα. Πιο συγκεκριμένα, εκπαιδεύτηκαν τέσσερα μοντέλα μη επιβλεπόμενης μάθησης βασισμένα σε GANs για την ενίσχυση του ρεαλισμού των εικόνων του GTA V. Οι μεταφρασμένες εικόνες αξιολογήθηκαν με τη χρήση κοινών μέτρων αξιολόγησης των GANs, καθώς και μέσω της επίδοσης στη σημασιολογική κατάτμηση. Τα αποτελέσματα υποδεικνύουν ότι τα μοντέλα που βασίζονται στη μάθηση της κυκλικής συνέπειας (cycle-consistency learning) μπορούν να διατηρήσουν καλύτερα τις λεπτομερείς γεωμετρίες, ενώ τα μοντέλα που βασίζονται στην αντιφατική μάθηση (contrastive learning) εκτελούν πιο επιθετικές αλλαγές με αποτέλεσμα να κάνουν περισσότερα λάθη, όπως να γεμίζουν τον ουρανό με δέντρα που δεν υπάρχουν. Η αξιολόγηση της ποιότητας των εικόνων μέσω της σημασιολογικής κατάτμησης αποδείχθηκε πιο αξιόπιστη σε τέτοιες περιπτώσεις, καθώς μετρικές όπως η Fréchet Inception Distance (FID) δεν μπορούν να ανιχνεύσουν τέτοιες αναντιστοιχίες στη δομή τους.

Λέξεις Κλειδιά

Μετάφραση από εικόνα σε εικόνα, βαθιά μάθηση, υπολογιστική όραση, παραγωγικά αντιπαραθετικά δίκτυα, σημασιολογική κατάτμηση.

Abstract

Over the past years, Deep Learning has grown rapidly and has contributed significantly to the development of Computer Vision. Similarly, since the emergence of Generative Adversarial Networks (GANs), the field of Generative Artificial Intelligence has been revolutionized. GANs are a family of models, a framework, that can learn patterns from existing data, such as images or text, and then generate new content with impressive results. The effectiveness of GANs has sparked a lot of interest giving rise to many new approaches and applications as more and more research is devoted around them. One such research topic is image-to-image translation, the task of transforming images from one domain so that they have the style or characteristics of images from another domain. Image-to-image translation can be applied to transfer artistic style (e.g. to transform a photo to look like a painting of a famous painter) or even to bridge the gap between synthetic and real images. This thesis focuses on the latter and aims to transform images of the open-world game Grand Theft Auto V (GTA V) to look like realistic urban scenes by applying state-of-the-art image-to-image translation methods. Specifically, four unsupervised models based on GANs were trained to enhance the realism of GTA V images. The translated images were evaluated with the use of common GAN evaluation measures as well as through the performance in semantic segmentation. The results suggest that models based on cycle-consistency learning can better preserve detailed geometries, while models based on contrastive learning perform more aggressive changes resulting in more mistakes, like populating the sky with trees that do not exist. The evaluation of the quality of the images through semantic segmentation proved to be more reliable in such cases, as metrics like Fréchet Inception Distance (FID) cannot detect such mismatched scene structures.

Keywords

Image-to-image translation, deep learning, computer vision, generative adversarial networks, semantic segmentation.

Ευχαριστίες

Ευχαριστώ τον επιβλέποντα καθηγητή της διπλωματικής μου εργασίας, κ. Ανδρέα-Γεώργιο Σταφυλοπάτη, για την ευκαιρία που μου έδωσε να επιλέξω ελεύθερα και να ασχοληθώ με ένα θέμα που με ενδιέφερε ιδιαιτέρως, καθώς και τον κ. Γεώργιο Σιόλα, για την καθοδήγησή του και τη συμβουλευτική του παρουσία.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου, και πρωτίστως τους γονείς μου για την ανιδιοτελή υποστήριξή τους, όπως επίσης και τους φίλους μου, για τη στήριξη και την υπομονή τους καθ' όλη τη διάρκεια του τελευταίου δύσκολου έτους.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

With the advent of deep learning, a substantial amount of progress has been made in various computer vision tasks such as image classification, object detection, semantic segmentation, face recognition and human pose estimation. The recent advances of generative methods and especially Generative Adversarial Networks (GANs) have sparked a huge interest in Generative Artificial Intelligence (AI). Generative AI is a technology that allows the creation of new content such as images, text and audio by learning underlying patterns of existing content. This field of AI offers many different use cases from simple content generation [27, 16] to more complicated tasks like sketch colorization [48], face aging [1], super-resolution [29], text-to-image synthesis [47], text-to-speech [28] and symbolic music generation [14].

Image-to-image translation, which falls under the category of Generative AI, is a class of vision and graphics problems where the goal is to transform images from one domain so that they have the style or characteristics of images from another domain. The techniques used to perform image-to-image translation are mostly based on GANs and can be used for tasks like artistic style transfer (e.g. photos to paintings) or to bridge the gap between synthetic and real images.

The rendering of realistic visual content is one of the most defining goals of computer graphics. Deep learning methods such as image-to-image translation could be used in the future to assist realistic graphics rendering as a post-processing step [37]. Photorealism is something many video games strive for and Grand Theft Auto V (GTA V) is one of them. GTA V is a 2013 action-adventure game developed by Rockstar North and published by Rockstar Games. It features a realistic open-world design that lets players freely roam and drive in the open countryside and the fictional city of Los Santos, which is based on Los Angeles. The photorealistic enhancement of GTA V graphics is something that the modding community also strives for by creating mods (short for “modifications”) to alter various aspects of the game so that it looks and feels more realistic.

Another application of image-to-image translation is in the context of domain adaptation [22]. Domain adaptation is a field of computer vision and machine learning which aims to solve tasks in a target domain that has a different, but related distribution to the source domain data on which a model was trained. Therefore, the goal is to reduce the shift between the two distributions by aligning the two representations (e.g. in feature space). For example, a model that allows self-driving cars to recognize driving areas may perform badly on snowy or foggy roads if the dataset which it was trained on did not capture such conditions. Therefore, the model should be adapted to new weather conditions or even to new driving scenarios in cities with different layouts.

Domain adaptation can also be used as a solution to small annotated datasets, with not enough data to train effectively a deep learning model. In the case of semantic segmentation, which is the task of classifying each pixel of an image into a category (e.g. road, car, person etc.) the annotation of training data is extremely time-consuming as it is usually performed manually. As a result, the creation of semantic segmentation datasets is very expensive leading to smaller amounts of available data. SYNTHIA [39], is an example of a commonly used dataset in the context of domain adaptation. Even though it consists of synthetic annotated images simulating driving scenarios in a virtual city, it can be used in order to adapt a model for semantic segmentation and scene understanding problems in real-world data. Similarly, annotated images from the game GTA V can be used for domain adaptation as that too simulates realistic driving scenarios.

1.2 Goals

The goal of this thesis is to apply and evaluate state-of-the-art methods in order to transform GTA V images into realistic urban scenes. These methods could later be used as the core of frameworks to improve the realism of game graphics or the performance of semantic segmentation through domain adaptation. For that purpose, this work aims to offer the following:

- Translations of GTA V images to two real-world datasets with different styles.
- Qualitative comparison of the visual results of four image-to-image translation models that are based on two different learning methods: cycle consistency and contrastive learning.
- Quantitative evaluation of the translated images using common GAN metrics as well as semantic segmentation.

1.3 Thesis Outline

This thesis is structured as follows:

Chapter 2 introduces the reader to the task of image-to-image translation. First, it provides the theoretical background of generative adversarial networks (GANs), as well as their deep convolutional and conditional versions. The models tested in this thesis are strongly based on those methods. Next, the different types of image-to-image translation settings are presented with a few representative models and applications for each. Finally, two related works are introduced which use image-to-image translation on GTA V images for two different tasks: photo-realistic enhancement of GTA V graphics and domain adaptation for semantic segmentation.

Chapter 3 introduces the datasets, pre-processing, and models used to translate GTA V images to real urban scenes. Specifically, four models for unsupervised image-to-image translation are analyzed and applied to translate GTA V images to two different real-world datasets. Additionally, the methods followed to evaluate the results are presented. These include the use of common GAN metrics as well as semantic segmentation.

Chapter 4 presents the final qualitative and quantitative results. It compares the performance of the image-to-image translation models both visually and with the use of metrics that evaluate the quality of the generated images.

Chapter 5 concludes this thesis by listing some final remarks related to the performance of the models and evaluation methods and it proposes potential future directions.

Appendices are also provided with additional visual results to complement chapter 4 without overcrowding the main body of the thesis.

Chapter 2

Image-to-Image Translation Using GANs

This chapter first provides a basic theoretical background on Generative Adversarial Networks (GANs), which is the main machine learning method used by the models tested in this thesis. Additionally, it introduces the types of different image-to-image translation tasks and most commonly adopted methods based on GANs. Finally, it presents related applications that specifically involve translating GTA V images to real-world images.

2.1 Theoretical Background

2.1.1 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) are an approach to generative modeling that was introduced by Goodfellow et al. [15] in 2014. The goal is to capture the distribution of the training data and discover its patterns so that the model can be used to generate new data by sampling from the learned distribution. In order to do this, the GAN framework uses two neural networks: the generator and the discriminator. In the original form of the framework (Vanilla GAN) these networks, also known as adversarial nets, are multilayer perceptrons (MLP). During training, the generator is constantly trying to outsmart the discriminator by generating better fake samples, while the discriminator is trying to decide whether a particular sample is real, meaning that it comes from the original dataset, or is fake and was produced by the generator. This process is called adversarial training and is based on game theory, where the generator competes against an adversary (i.e. the discriminator). Specifically, the GAN framework corresponds to a minimax two-player game where the discriminator D tries to maximize the probability it correctly classifies real and fake data, and the generator G tries to minimize the probability that D will predict its outputs are fake. The GAN loss function is formulated as:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}\,(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))]. \tag{2.1}$$

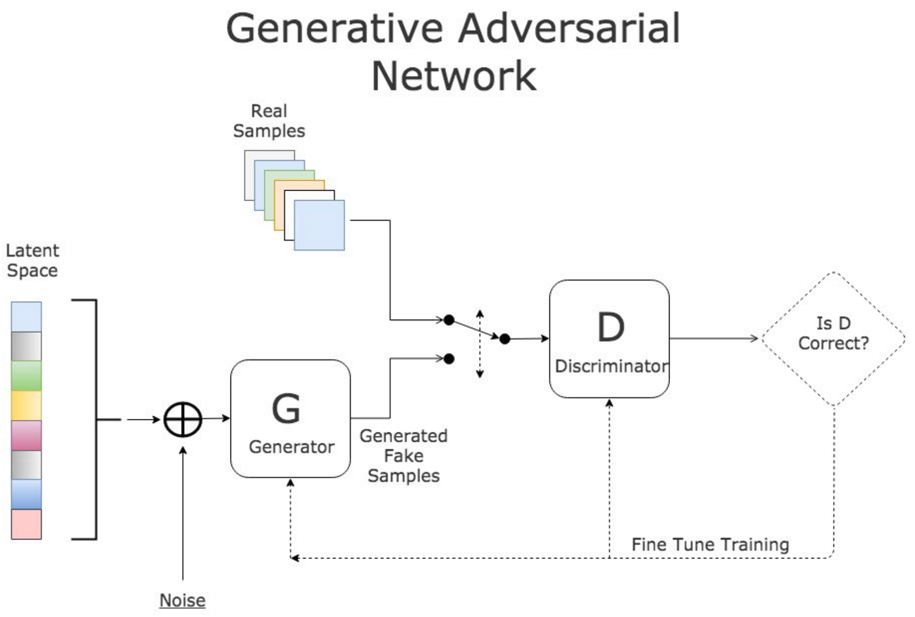


Figure 2.1: Illustration of the framework and learning process of Generative Adversarial Networks. Source: tinyurl.com/9v9t6m2c

$G(z)$ represents the generator function which maps a random noise vector z to data-space and G tries to estimate the distribution p_{data} that the training data come from so it can generate fake samples from the estimated distribution p_g. $p_z(z)$ is a prior on input noise variables. $D(x)$ is the scalar output of the discriminator and represents the probability that x came from the data rather than p_g. $D(G(z))$ is the scalar probability that the output of the generator G is real. D is trained to maximize $logD(x)$ and simultaneously G is trained to minimize $log(1 - D(G(z)))$.

Figure 2.1 illustrates the process of training GANs. During each training step a minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ is sampled from noise prior $p_g(z)$ and a minibatch of m examples $\{x^{(1)}, \ldots, x^{(m)}\}$ is sampled from data generating distribution $p_{data}(x)$. The discriminator tries to identify real from fake data and based on the outcome both networks update their parameters alternately through backpropagation and the flow of gradients. Specifically, the discriminator is updated by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right] \tag{2.2}$$

and the generator is updated by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right). \tag{2.3}$$

In theory, the solution to the minimax game is where $p_g = p_{data}$, which means that neither the generator nor the discriminator can improve further as the discriminator is unable to differentiate between real and fake data, i.e. $D(x) = \frac{1}{2}$. In practice, however, models do not always train to this point. Moreover, the training of GANs is usually quite difficult and unstable and may suffer from problems like the vanishing gradients and the mode collapse [2]. The vanishing gradient problem occurs in GANs when the discriminator learns to perfectly differentiate between real and fake data and as a result, it does not provide reliable gradient information to train the generator. Indeed, it is often an easier problem to distinguish real from fake data than to generate realistic samples. The mode collapse problem occurs when the generator tries to fool the discriminator with a few patterns or examples and only generates this small subset of outputs resulting in a limited variety of produced data.

2.1.2 Deep Convolutional GANs

The use of multilayer perceptrons as generators and discriminators restricts the performance and applications of GANs, especially in deep learning tasks such as image, text, or sound generation. Radford et al. [36] extended the previous work by introducing deep convolutional generative adversarial networks (DCGANs), which use convolutional neural networks (CNNs) for the generator and discriminator. This framework serves as a strong baseline for many other works that use GANs for deep generative modeling. The proposed architectural guidelines for the training of stable DCGANs are the following:

- Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm in both the generator and the discriminator.
- Use ReLU activation in the generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

Additionally, the authors initialize the weights of the networks from a zero-centered Normal distribution with a standard deviation of 0.02. They also use the Adam optimizer with a learning rate of 0.0002 instead of the usual 0.001 and change the momentum term β_1 from the default 0.9 to 0.5 to improve the training process.

The authors applied the DCGAN framework to image generation in their work. The discriminator of DCGAN is basically a CNN for image classification which tries to distinguish between real and fake images. What may be more interesting is the generator, which is comprised of fractional-strided convolutions, also known as transposed convolutions, instead of traditional convolutions. The standard convolutional layers typically reduce (downsample) the spatial dimensions (height and width) of the input, or keep them unchanged. On the other hand, transposed convolutions increase (upsample) the spatial dimensions of intermediate feature maps. As shown in figure 2.2, the input noise vector

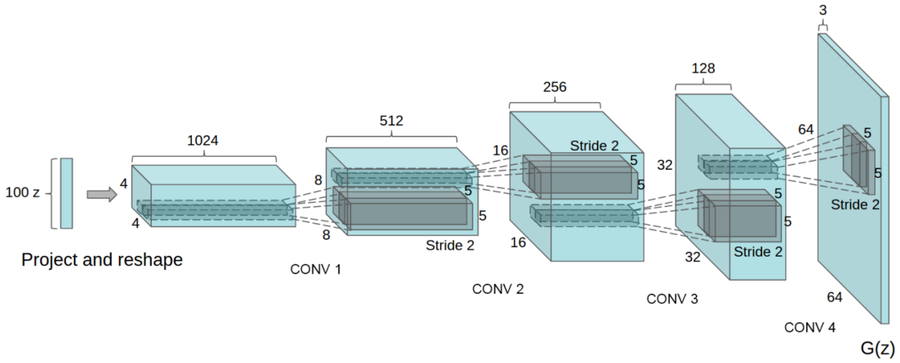


Figure 2.2: The convolutional generator of the DCGAN proposed in [36].

z of the generator is first projected and reshaped to 1024×4×4, and then with a series of transposed convolutions, it goes to 3×64×64 which is the generated output image.

2.1.3 Conditional GANs

Traditional GANs can only generate samples randomly from the distribution they have learned and there is no control over the output. Therefore, Mirza et al. [31] introduced the conditional version of GANs. The conditioning can be performed by simply feeding some extra information y into both the generator and discriminator as an additional input layer. The conditional variable y can be any type of information. For example, in the case of image-to-image translation, y is an entire image and more than one conditional variable can also be used.

The authors used conditional GANs (cGANs) to control the generated images according to their class labels. As shown in figure 2.3, the input of the generator is not only the noise vector z but also the class labels y. As a result, the generator produces only images of the desired classes. Additionally, the discriminator's evaluation is done not only on the similarity between fake and real images but also on the correspondence of the fake images to its input labels. In this case, the objective function of the two-player minimax game is modified to:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x} \mid \boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z} \mid \boldsymbol{y})))]. \tag{2.4}$$

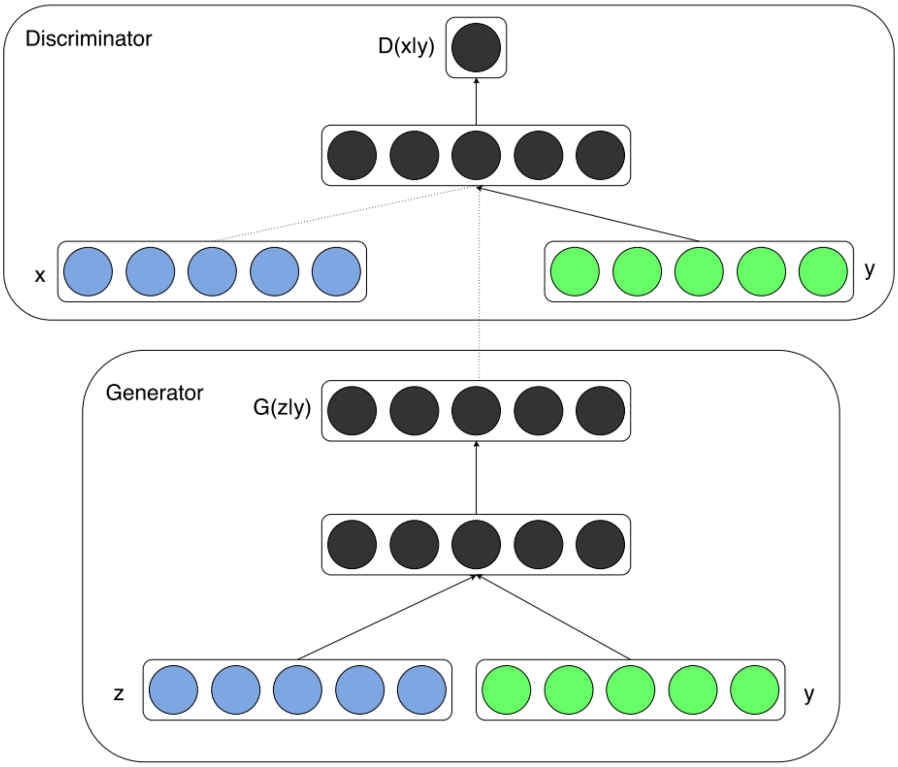


Figure 2.3: The conditional GAN proposed in [31].

2.2 Image-to-Image Translation Types

2.2.1 Supervised Translation

Image-to-image translation is the task of transforming images from one domain so that they have the style or characteristics of images from another domain while preserving the content representations. A good way to think of image-to-image translation is as a special case of the Conditional GAN. However, in this case, the generator is conditioned on a complete image (rather than just a class) and the networks are CNN variations based on DCGAN. When the ground truth of the transformation is available in the form of training image pairs from the source and target domains then the task is called supervised or paired image-to-image translation. For example, if we have many aligned image pairs of the same locations captured both day and night, then it is possible to train a deep learning model to translate photos from day to night.

Pix2pix [23] is one of the strongest baselines that allow this kind of transformation using aligned data. It includes the training of a generative adversarial network that aims to generate synthetic images indistinguishable from real-world images deploying an additional L_1 pixel-wise regression loss between the translated image and its ground truth pair. Figure 2.4 shows an example of pix2pix applications.

Pix2pixHD [46] and later SPADE [34] introduced many improvements to the pix2pix

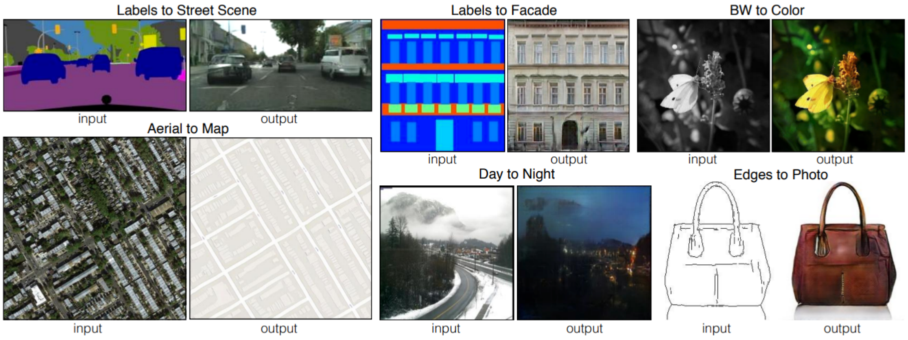


Figure 2.4: Example applications of supervised image-to-image translation using pix2pix. Source: [23]

model in order to synthesize more realistic and high-resolution images from semantic label maps. Figure 2.5 shows an example usage of NVIDIA Canvas (also known as GauGAN) which is a desktop application based on SPADE. It allows the generation of realistic-looking landscapes out of simple brushstrokes which represent semantic label maps.

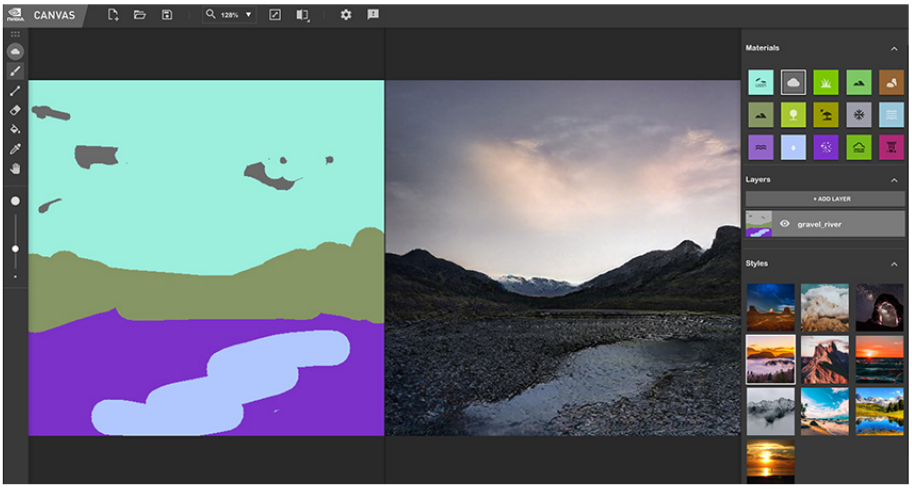


Figure 2.5: Example application of NVIDIA Canvas which is based on SPADE. Source: https://www.nvidia.com/en-us/studio/canvas/

2.2.2 Unsupervised Translation

The applications where the supervised setting of image-to-image translation can be applied are quite limited due to the difficulty of obtaining a large amount of paired data. For example, if the task were to translate photos from summer to winter, it would be much easier to create a dataset consisting of photos from random locations instead of the same locations during summer and winter. Another application would be translating real photos to have the style of a famous artist's paintings. It would be nearly impossible to create a paired dataset in this case, especially if the artist is not in life anymore.

Zhou et al. [50] introduced CycleGAN, which allows performing image-to-image translation even when training on unpaired data. Similar to pix2pix it is based on conditional GANs, but it uses two pairs of generators and discriminators. Each pair translates images from one domain to another and its key feature is the cycle-consistency loss, which ensures that the translations are reversible. Figure 2.6 shows some applications of CycleGAN in an unsupervised setting. This model, along with other approaches that introduce various improvements, are tested in this thesis and described in more detail in Chapter 3.

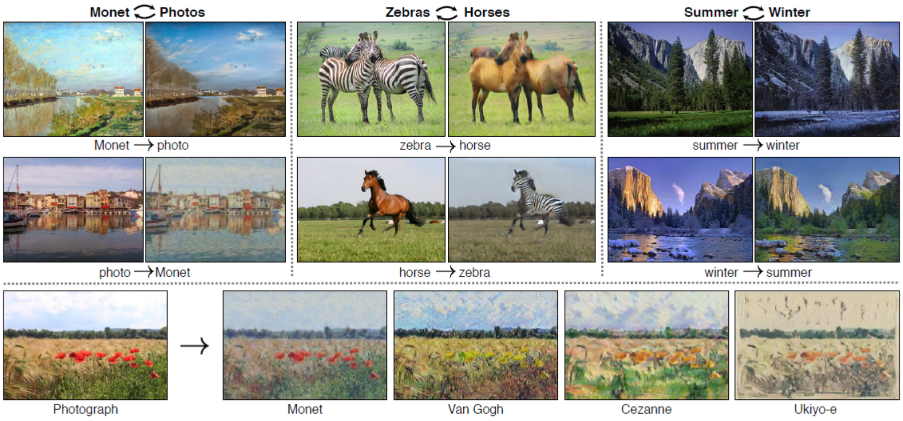


Figure 2.6: Example applications of unsupervised image-to-image translation using CycleGAN. Source: [50]

2.2.3 Multi-Domain Unsupervised Translation

CycleGAN and other similar models are restricted in translating only one domain to another. For example, if the task were to translate a person's face to show three different emotions (e.g. angry, happy and fearful) CycleGAN should be trained three times, one for each domain of images. This would not be practical, especially when translating to even more domains.

StarGAN [11] extends CycleGAN, in order to train a single network for multi-domain unsupervised translation. Specifically, StarGAN can apply different translations on an input image by providing the target domain label. Figure 2.7, shows an example of applying StarGAN on the facial attribute transfer and facial expression synthesis tasks. Each output is synthesized given the input image and the corresponding domain label.

StarGAN v2 [12] improves the diversity of StarGAN's synthesized images across domains, by providing style codes of a specific domain instead of predefined labels. Style codes are extracted from a style encoder and represent different styles for a specific domain. For example, if the domains are based on the gender of a person the style codes could represent different hairstyles or facial hair. Figure 2.8 shows example applications of StarGAN v2.

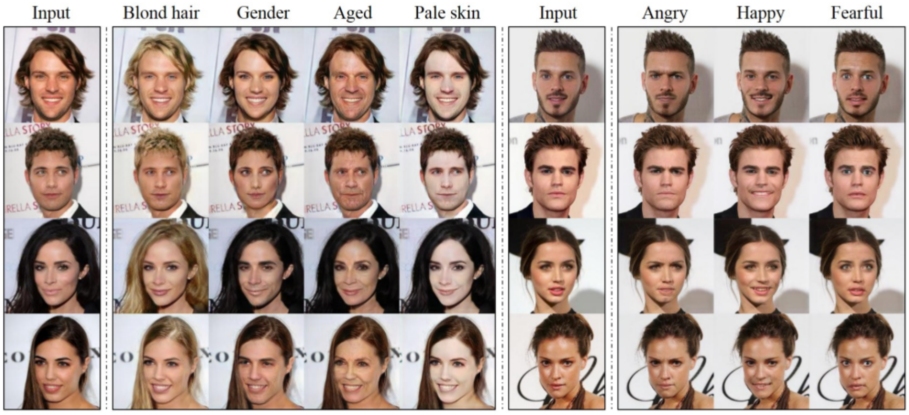


Figure 2.7: Example of multi-domain translations on the facial attribute transfer and facial expression synthesis tasks using StarGAN. Source: [11]

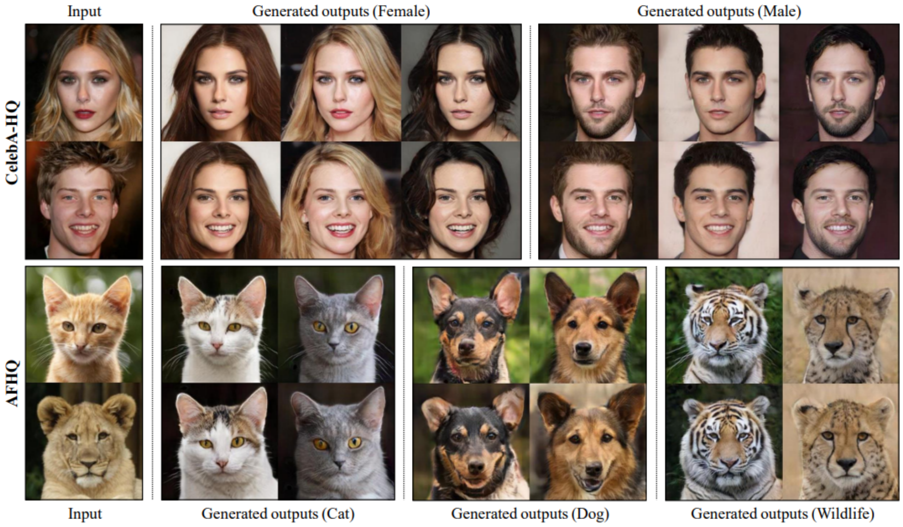


Figure 2.8: Example applications of diverse translations across domains using StarGAN v2. Source: [12]

2.3 Related Work Using GTA V Images

Richter et al. [37] propose in their work, “Enhancing photorealism enhancement”, a framework to connect the deep learning approaches for image-to-image translation to the rendering pipelines of computer games. The key ingredient of their model is the use of G-buffers, which are intermediate rendering buffers produced by game engines during the rendering process. Specifically, they designed a network to modulate features from a rendered image according to representations extracted from G-buffers that provide information about geometric structure (surface normals, depth), materials (shader IDs, albedo, specular intensity, glossiness, transparency), and lighting (approximate irradiance and emission, sky, bloom). They train their model using an LPIPS loss [49] which penalizes large structural differences between input and output images and a perceptual discriminator that consists of a robust semantic segmentation network, a perceptual feature extraction network, and multiple discriminator networks. After an ablation study of such training and architectural choices, they created a model that manages to improve the photorealistic enhancement of GTA V images and reduce artifacts compared to general image-to-image translation models that are trained only using the source and target domain images.

Hoffman et al. proposed CyCADA [22], an adversarial domain adaptation model that is based on the CycleGAN image-to-image translation model. In the case of GTA V, they use CyCADA for the semantic segmentation adaptation of the synthetic images to real-world imagery. This way they manage to recover approximately 40% of the performance on semantic segmentation lost to domain shift. CyCADA adapts representations at both pixel-level and feature-level and uses the cycle consistency together with a semantic consistency loss to guide the mapping from one domain to another. The cycle consistency loss, as introduced by CycleGAN, ensures that the translations from one domain to the other are reversible. The semantic consistency is enforced through a pre-trained source task model which is used as a noisy labeler to encourage an image to be classified in the same way after translation as it was before translation.

Chapter 3

Methodology

This chapter describes the datasets used in this work, along with their preprocessing, as well as the architectures and theoretical background of the different models that were tested. Finally, it analyzes the methods that were followed to evaluate and compare the performance of the models.

3.1 Datasets

This section presents the datasets used for the image-to-image translation task that is explored. Specifically, a single dataset consisting of GTA V images represents the source domain and two real-world datasets are used as target domains. Therefore, two different kinds of translations are produced in this thesis, each with its own style.

3.1.1 GTA V

The motivation behind collecting the GTA V images in [38] is to create a very large dataset with pixel-accurate semantic labels. GTA V is an open-world game that offers a highly realistic environment and driving scenarios, therefore the performance of segmentation models could benefit from such data. As mentioned in Chapter 1, such a task is very challenging due to the amount of human effort required to trace accurate object boundaries and as a result, the annotation of a single image requires 60 and 90 minutes for the CamVid [5] and Cityscapes [13] datasets respectively. The authors, however, extracted 25 thousand images from GTA V and successfully annotated them automatically in only 49 hours. Annotating this dataset manually following the approach used for CamVid or Cityscapes would take 12 person-years.

GTA V, however, is not an open-source game, which means that the source code is not available. As a result, the creation of semantic labels is not very straightforward and the authors apply a technique called detouring [6] using an off-the-shelf graphics debugging tool called RenderDoc [26]. Specifically, they create a wrapper around the game's graphics library (i.e. Direct3D 11) and they use it to intercept communication with the graphics hardware to gain access to the game's resources and save all information needed

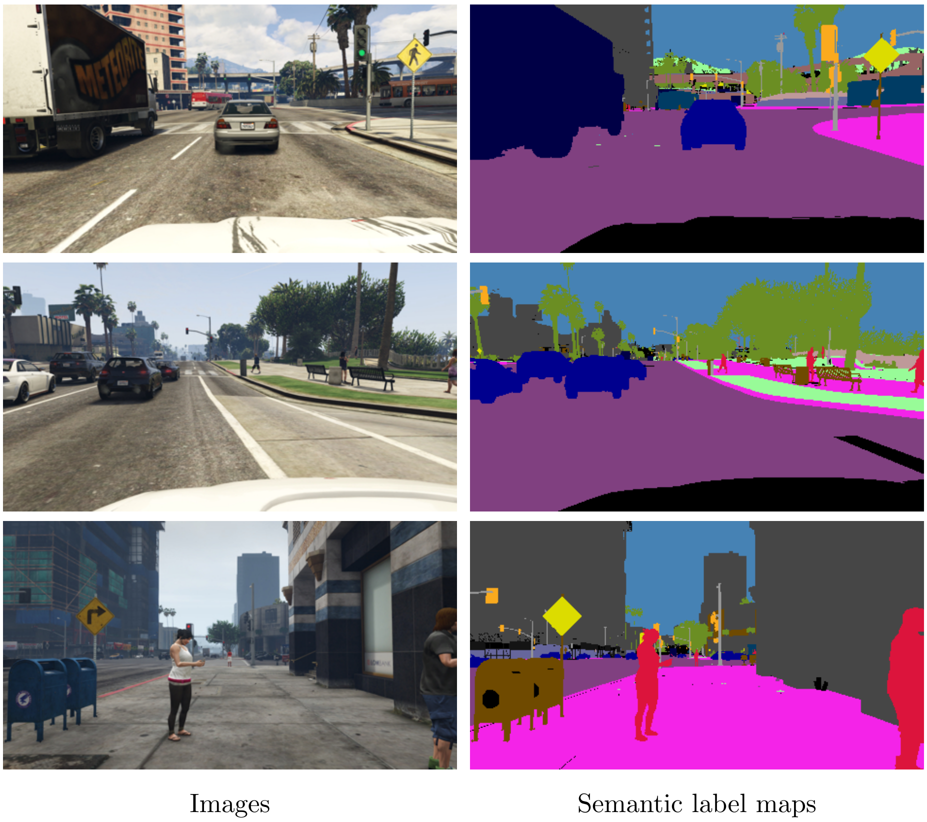


Figure 3.1: Example images from the GTA V dataset with their corresponding semantic label maps.

to reproduce a frame. They later use this information to annotate the images efficiently using label propagation. Specifically, after annotating the first image their annotation tool automatically propagates the labels to all image patches that share the same mesh, texture and shader combination in all images. This way, as the annotation progresses more and more image patches are pre-labeled significantly decreasing the annotation time per image.

The dataset is available at https://download.visinf.tu-darmstadt.de/data/from_games/ and it consists of 24966 frames from GTA V along with their semantic labels. Figure 3.1 shows some samples from this dataset. Each frame has a resolution of 1914×1052 pixels and there are labels for 19 classes, which are compatible with the Cityscapes dataset described below. In this thesis, 18785 frames were used to train the image-to-image translation models, while the rest 6181 frames were used to evaluate their performance following the official test split.

3.1.2 Cityscapes

Cityscapes [13] is a large-scale dataset that is commonly used to develop and test models for semantic segmentation and visual understanding of complex urban scenes. In order to create this dataset, the authors recorded videos on a moving vehicle in 50 cities

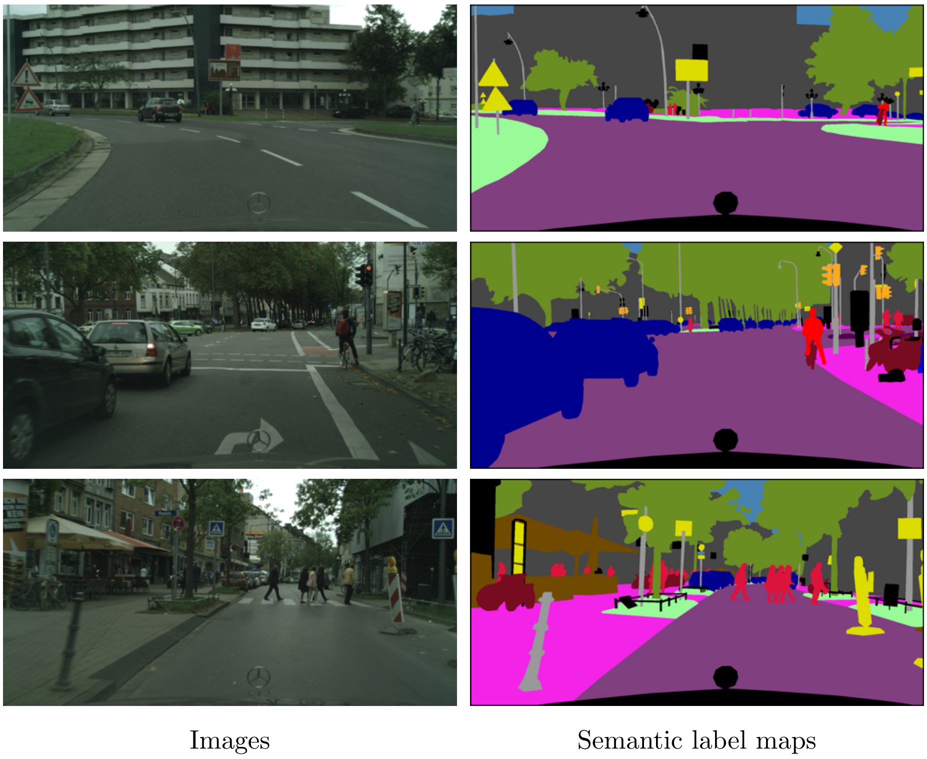


Figure 3.2: Example images from the Cityscapes dataset with their corresponding semantic label maps.

of Germany and other neighboring countries. The videos were recorded during several months across spring, summer and fall to ensure the diversity of the data. However, the authors chose not to record under unfavorable weather conditions such as heavy rain or snow believing that would require specialized techniques and datasets.

The images were recorded with an automotive-grade stereo camera at a frame-rate of 17Hz. The sensors that were mounted behind the windshield yielded high dynamic-range (HDR) images with 16 bits linear color depth. These images are also provided at 8-bit low dynamic range (LDR) for comparability and compatibility with other existing datasets. The authors provide 5000 images selected from 27 cities with dense pixel-level annotations that were done on the 20$^{\text{th}}$ frame of a 30-frame video snippet. Additionally, they provide 20000 images with coarse annotations for the remaining 23 cities done every 20s or 20m driving distance. All images have a resolution of 2048×1024 pixels.

The data can be found at https://www.cityscapes-dataset.com/ and Figure 3.2 shows a few samples. In this thesis, these images are used in two different ways. The 20000 coarsely annotated images are used for the image-to-image translation task in order to transform GTA V to look like Cityscapes. The semantic labels are not used for this task. The densely annotated images are then used to evaluate the translated results as explained in section 3.4. The official train-validation split is followed for the densely annotated images, comprising of 2975 and 500 images respectively. The test set of 1525 images is not used in the experiments, because no annotations are provided and serves as

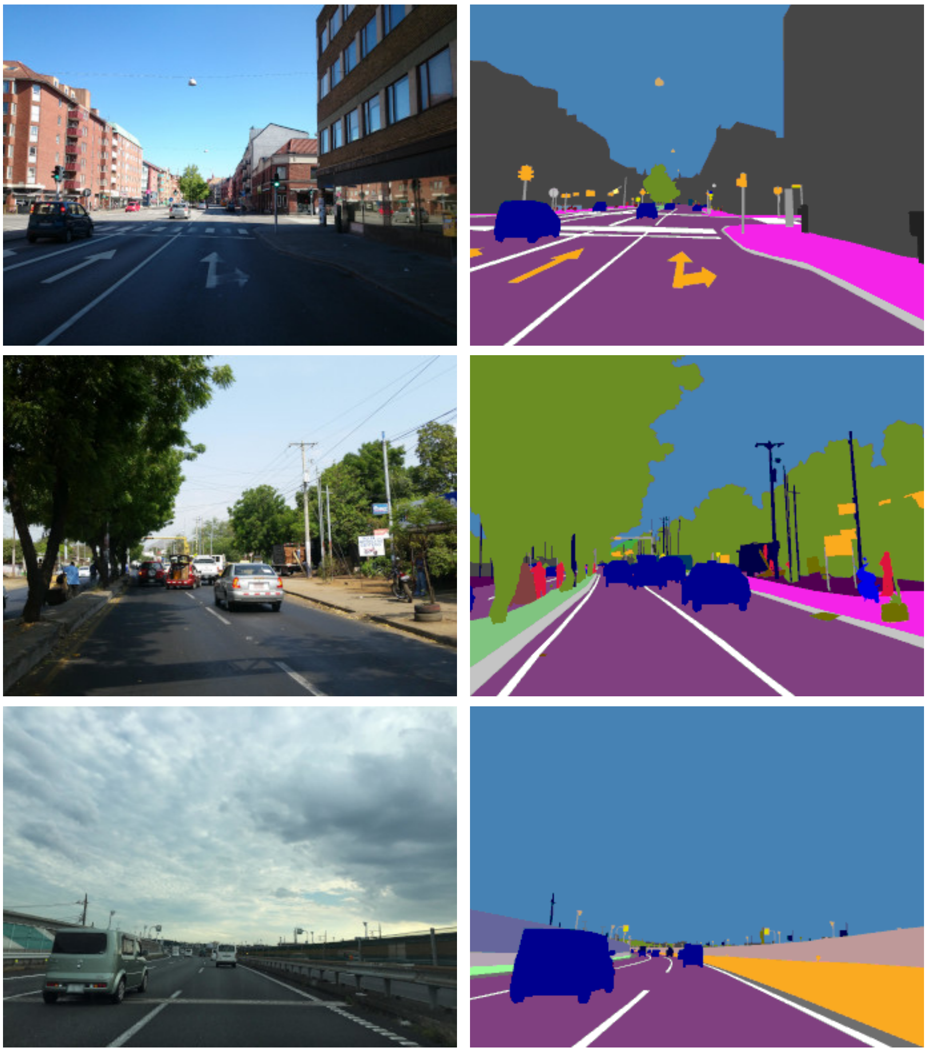


Figure 3.3: Example images from the Mapillary Vistas dataset with their corresponding semantic label maps.

a benchmark.

3.1.3 Mapillary Vistas

Mapillary Vistas [32], similarly to Cityscapes, is a large-scale street-level image dataset introduced to contribute to the visual scene understanding task. Its main motivation is the diversity, richness of detail and geographic extent of street-level data. The dataset comprises of 25000 densely annotated images into 66 object categories. Figure 3.3 shows a few samples from the dataset.

Unlike GTA V and Cityscapes, the image data of Mapillary Vistas are not collected as frames from video footage. Instead, the images are extracted from Mapillary: https://www.mapillary.com/ and visually cover parts of Europe, North and South America, Asia, Africa and Oceania. Mapillary is a community-led service where people and organizations can contribute to visualizing the world and building better maps. Anyone can collect and share street-level images of any place. As a result, the dataset consists of images taken from different devices, such as smartphones or action cameras, as well as differently experienced photographers. Additionally, in contrast to Cityscapes, the images of Mapillary Vistas were selected in a way that they cover a variety of weather and lighting scenarios. Sunny, rainy, cloudy, foggy and snowy conditions are represented in the dataset. Furthermore, images are taken at day, dawn, dusk and night. The images do not have a fixed resolution, but a minimum width/height of 1920×1080 is imposed.

This dataset can be found at https://www.mapillary.com/dataset/vistas. In this thesis, the 18000 train images of the official train-validation-test split are used in order to transform GTA V to look like Mapillary Vistas, while the 2000 images of the validation set are used to evaluate the results. In this case, the semantic label maps are not used for the evaluation of the translated results, because the semantic object categories are not directly compatible with the GTA V annotations.

3.2 Preprocessing

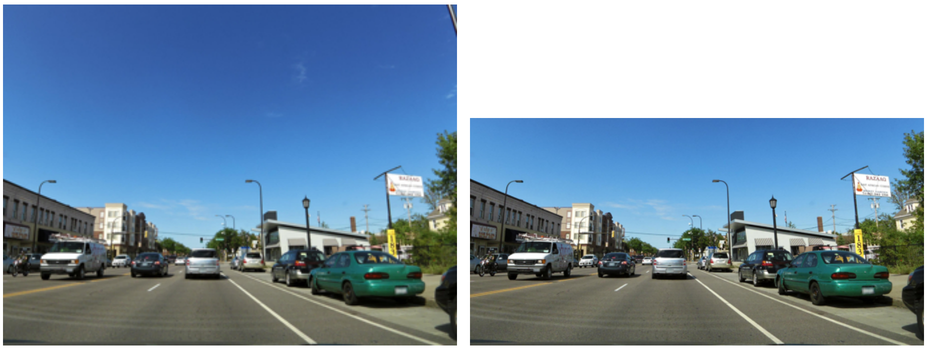
The training of image-to-image translation models can be quite computationally expensive, rendering it extremely time-consuming or even impossible (due to memory constraints) to train them using high-resolution images in a single conventional GPU. For that reason, all the images were resized to reduce the resolution but keeping the original aspect ratio. Specifically, GTA V images were resized from the original 1914×1052 resolution to 468×256 and Cityscapes images from 2048×1024 to 512x256.

Before resizing the images of Mapillary Vistas they were cropped so that their height equals half their width. Due to the nature of the dataset, many images contain too much sky volume. This impacted the models negatively, as they learned to erase objects such as buildings and replace them with the sky. Appendix B shows an example of this type of failure. By cropping the images in order to reduce the sky volume in a straightforward way this problem was solved. Figure 3.4 shows a sample from the dataset before and after cropping. After that, the images were resized reducing the height to 256 and keeping the original aspect ratio.

Finally, it is important to note that the semantic label maps of GTA V and Cityscapes were resized using the nearest-neighbor interpolation method, because the pixel values represent class labels and they should not be changed, as would happen using other methods such as bi-linear interpolation.

3.3 Image-to-Image Translation Models

The available datasets that were presented in section 3.1 are not paired. That means that there is no ground truth of how a GTA V image should be translated to Cityscapes



Original Cropped

Figure 3.4: Example image from the Mapillary Vistas dataset before and after cropping as a preprocessing step.

or Mapillary Vistas style. Therefore, the task at hand is that of unsupervised or unpaired image-to-image translation. This section analyzes the four chosen models suitable for such a task and which were tested in this thesis.

3.3.1 CycleGAN

CycleGAN was presented by Zhou et al. [50] introducing the use of two generative adversarial networks in order to tackle the difficulty of unsupervised image-to-image translation. By using two generators and two discriminators it is possible to learn both mappings between two domains X and Y. For example, considering the task of this thesis where X denotes the GTA V images and Y denotes either the Cityscapes or Mapillary Vistas images, CycleGAN learns to translate both GTA V images to real-world images ($X \to Y$) and the inverse ($Y \to X$). The generator G performs the first mapping $G : X \to Y$ and the second generator F performs the inverse mapping $F : Y \to X$, while the corresponding discriminators aim to distinguish between real and translated images. The official source code of CycleGAN is available at https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix.

Adversarial Loss

Both GANs are trained traditionally using the adversarial loss. Continuing the example from above, the generator G tries to generate images $G(x)$ that look similar to real-world images from domain Y, while D_Y tries to distinguish between translated GTA V samples $G(x)$ and the real-world images y. The objective is expressed as:

$$\begin{aligned}\mathcal{L}_{\text{GAN}}\left(G, D_Y, X, Y\right) = \;& \mathbb{E}_{y\sim p_{\text{data}}(y)}\left[\log D_Y(y)\right] \\ & + \mathbb{E}_{x\sim p_{\text{dan}}(x)}\left[\log\left(1 - D_Y(G(x))\right]\right. .\end{aligned} \tag{3.1}$$

Similarly, the generator F tries to generate images $F(y)$ that look similar to GTA V images x, while D_X aims to distinguish between translated real-world images $F(y)$ and the actual GTA V images x. This objective is expressed as:

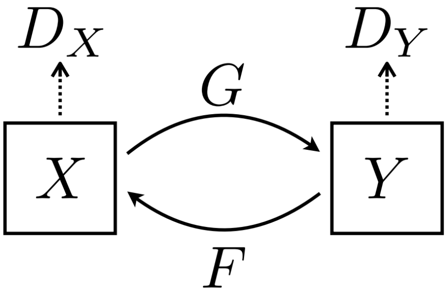


Figure 3.5: Illustration of the adversarial training of CycleGAN's generators and discriminators. Source: [50]

$$\begin{aligned}\mathcal{L}_{\mathrm{GAN}}\left(F, D_X, X, Y\right) = {} & \mathbb{E}_{x \sim p_{\mathrm{data}}(x)}\left[\log D_X(x)\right] \\ & + \mathbb{E}_{y \sim p_{\mathrm{dan}}(y)}\left[\log \left(1 - D_X(G(y))\right] .\end{aligned} \tag{3.2}$$

The above procedure is illustrated in Figure 3.5.

Cycle Consistency Loss

Cycle consistency is a key feature of the model (hence the name CycleGAN). The concept of transitivity can be found in the task of translating between languages [4, 18]. For example, an English phrase translated to French should be identical to the original if it is then translated back to English. Similarly, translating a French phrase to English and back to French should give the same phrase. This is the idea that the authors use but in the context of translating images between domains. The cycle consistency loss encourages that $F(G(x)) \approx x$ and $G(F(y)) \approx y$. This means that if we translate from one domain to the other and back again we should arrive at where we started. This is enforced by using two cycle consistency losses. The forward cycle-consistency loss: $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$ and the backward cycle consistency loss: $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$. Figure 3.6 illustrates this procedure. The objective, which is expressed as:

$$\begin{aligned}\mathcal{L}_{\mathrm{cyc}}\left(G, F\right) = {} & \mathbb{E}_{x \sim p_{\mathrm{data}}\ (x)}\left[\|F(G(x)) - x\|_1\right] \\ & + \mathbb{E}_{y \sim p_{\mathrm{data}}\ (y)}\left[\|G(F(y)) - y\|_1\right] ,\end{aligned} \tag{3.3}$$

tries to minimize the $L1$ distance between the original image and its reconstruction through a series of translations.

Identity Loss

The last loss function added is the identity loss [42], and the authors use it to encourage the generator to learn the identity mapping. This way, when an image of the target domain is used as an input, the generator is expected to produce nearly the same image. The objective can be expressed as:

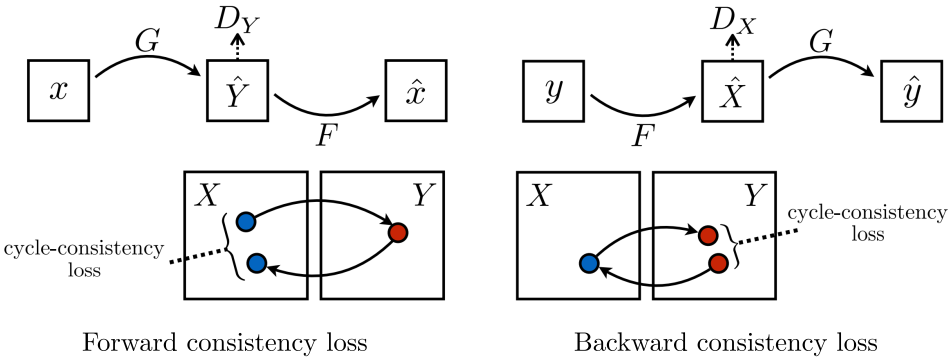


Figure 3.6: Illustration of the forward and backward consistency losses used to train CycleGAN. Source: [50]

$$\mathcal{L}_{\text{identity}}(G, F) = \mathbb{E}_{y \sim p_{\text{data}}(y)}\left[\|G(y) - y\|_1\right] + \mathbb{E}_{x \sim p_{\text{data}}(x)}\left[\|F(x) - x\|_1\right]. \tag{3.4}$$

In practice, the authors noticed that it helps retain the colors of the original image, while not using it might lead to unnecessary changes of the input image's tint.

Architecture

Both of CycleGAN's generators follow Johnson et al. [25] and have the same architectures. These consist of two convolutional layers followed by nine residual blocks and lastly two transposed convolutions and the output layer. An instance normalization is used after each convolutional layer followed by a ReLU activation function. In the case of the discriminators, the authors use 70 × 70 PatchGAN [24] which consists of five convolutional layers. The discriminators also use instance normalization except for the first convolutional layer and as activation function, they use the leaky ReLU with a slope of 0.2.

3.3.2 AttentionGAN

AttentionGAN gets its name from the Attention-Guided Generative Adversarial Networks that it uses and it was introduced by Tang et al. [43, 44]. This model builds upon CycleGAN with the goal to improve the performance of image-to-image translation by identifying the most discriminative foreground objects of the images and applying transformations only on those, minimizing the change of the background. For example, when transforming an image of a horse to look like a zebra, only this object should change and the background should remain intact. The authors introduced two architectural schemes for AttentionGAN, with the second being an improvement of the first. Therefore, this thesis analyzes and tests only the second scheme. The official code of AttentionGAN can be found at https://github.com/Ha0Tang/AttentionGAN.

As previously mentioned AttentionGAN builds upon CycleGAN. Indeed, these two models share most of their architectural and training details including the use of cycle consistency and identity loss. The novelty of AttentionGAN, however, is the attention-guided generators. By equipping these generators with a built-in attention module, it is possible to disentangle the most discriminative foreground objects from the unwanted background of the images. Specifically, each of the two generators G and F have two separate sub-networks for generating content masks and attention masks. The generator G for example consists of three parts. The parameter-sharing encoder G_E, the content mask generator G_C which produces $n-1$ content masks $\{C_y^f\}_{f=1}^{n-1}$ and the attention-mask generator G_A, which produces both $n-1$ foreground attention masks $\{A_y^f\}_{f=1}^{n-1}$ and one background attention mask A_y^b ($n = 10$ in the experiments). In order to produce the translated image, the attention masks are multiplied by the corresponding content masks, which is expressed as:

$$G(x) = \sum_{f=1}^{n-1} \left(C_y^f * A_y^f \right) + x * A_y^b \tag{3.5}$$

for the generator G, and for the generator F as:

$$F(y) = \sum_{f=1}^{n-1} \left(C_x^f * A_x^f \right) + y * A_x^b. \tag{3.6}$$

In the case of cycle consistency, the reconstruction process of an image from domain X is formulated as:

$$F(G(x)) = C_x * A_x + G(x) * (1 - A_x)\,. \tag{3.7}$$

Similarly, if the image comes from domain Y it is expressed as:

$$G(F(y)) = C_y * A_y + G(y) * (1 - A_y)\,. \tag{3.8}$$

The above procedures are illustrated in Figure 3.7.

3.3.3 CUT

CUT [33] is a novel approach for unpaired image-to-image translation, which is based on contrastive learning. In contrast to CycleGAN and AttentionGAN, CUT abandons the use of the cycle consistency loss, because it is too restrictive. The cycle consistency loss helps preserve the content of the translated image. For example, when translating a horse to a zebra the pose of the horse should not change. Similarly, the cycle consistency loss ensures that when translating back to the horse we get the original. However, a zebra

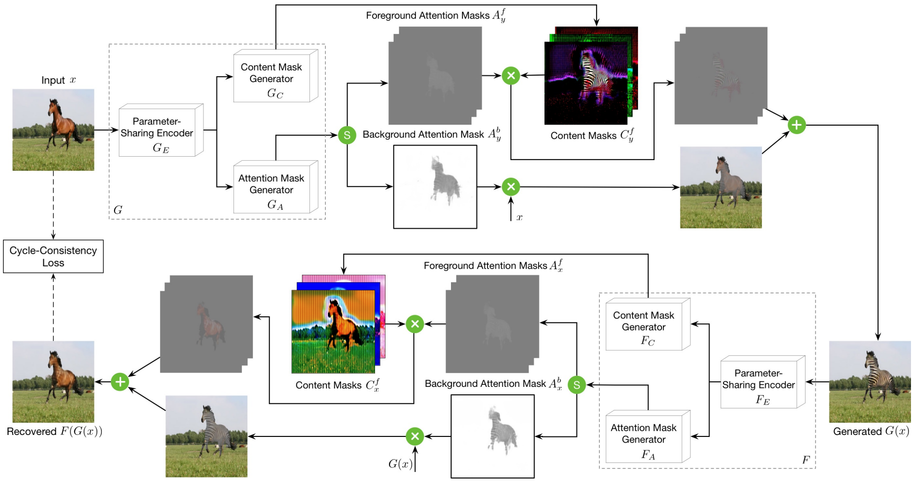

Foreground Attention Masks A_y^f
Input x
Content Mask Generator G_C
Parameter-Sharing Encoder G_E
Attention Mask Generator G_A
G
S
Background Attention Mask A_y^b
Content Masks C_y^f
x
Cycle-Consistency Loss
Foreground Attention Masks A_x^f
Content Mask Generator F_C
Parameter-Sharing Encoder F_E
Attention Mask Generator F_A
F
Content Masks C_x^f
Background Attention Mask A_x^b
Recovered $F(G(x))$
Generated $G(x)$
$G(x)$

Figure 3.7: The framework of AttentionGAN. The illustration shows the way the attention and content masks are produced and embedded into the image-to-image translation task. The symbols $\oplus$, $\otimes$ and ⓢ denote element-wise addition, multiplication, and channel-wise Softmax, respectively. Source: [43]

could translate to multiple horses (brown, black, white, etc.) and that would be correct. It should not really matter what the color of the horse is as long as it keeps the same pose as the zebra and of course looks like a horse. The cycle consistency though does not allow this to happen. This is why the authors introduce the patchwise contrastive loss. This loss, named PatchNCE loss by the authors, allows the model to keep the important aspects of the image intact, like the orientation and the structure of the foreground object to be translated, but does not apply restrictions on the appearance. Additionally, CUT uses only one generator-discriminator pair making it faster and lighter to train, but can only translate images in one direction. The official code of the model can be found at https://github.com/taesungp/contrastive-unpaired-translation.

As shown in Figure 3.8, the generator considers a pair of input and output patches at the same location and enforces that the embeddings of the pair are similar. Meanwhile, the patches from other locations should be mapped far from them. This is formulated as a contrastive loss inspired by InfoNCE loss [45], where the corresponding input and output patches form a positive pair, while patches from other locations form negative pairs. The objective is to maximize the mutual information between the positive pair while minimizing that of the negative pairs. For example, a zebra's head should be more closely associated with an input horse's head than the same horse's leg or other parts of the image. The authors do this by formulating a classification problem with the target class as the positive pair. In order to project the patches to the embedding space, they use the first half of the generator as an encoder. The output (called a "query") and input embedding pair of positive patches are denoted as v, v^+ respectively and the negative patches as v^-. The cross-entropy loss used for this task is calculated as:

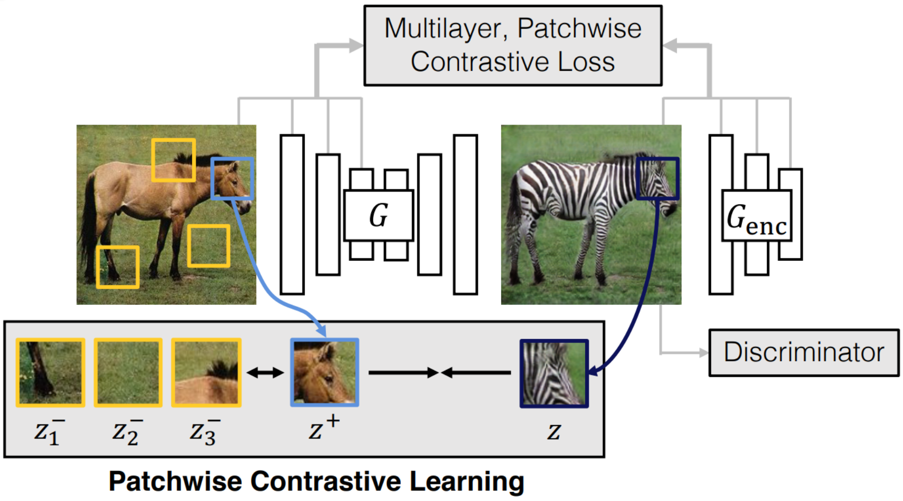

Multilayer, Patchwise Contrastive Loss
G
G_{enc}
Discriminator
z_1^-
z_2^-
z_3^-
z^+
z
Patchwise Contrastive Learning

Figure 3.8: Illustration of the patchwise contrastive learning used by CUT, which maximizes mutual information between the corresponding input and output patches. Source: [33]

$$\ell\left(\boldsymbol{v}, \boldsymbol{v}^{+}, \boldsymbol{v}^{-}\right)=-\log \left[\frac{\exp \left(\boldsymbol{v} \cdot \boldsymbol{v}^{+} / \tau\right)}{\exp \left(\boldsymbol{v} \cdot \boldsymbol{v}^{+} / \tau\right)+\sum_{n=1}^{N} \exp \left(\boldsymbol{v} \cdot \boldsymbol{v}_{n}^{-} / \tau\right)}\right] \tag{3.9}$$

and represents the probability of selecting the positive patch over the negatives. The vector distances are scaled by a temperature $\tau = 0.07$.

Finally, following CycleGAN, the authors apply an identity loss regularization, but with a twist. In this case, they provide samples of the target domain to the generator and enforce the contrastive loss to prevent making unnecessary changes to the input images of the source domain. This loss can be considered as a learnable, domain-specific version of the identity loss used by CycleGAN.

3.3.4 DCLGAN

DCLGAN was introduced by Han et al. [17]. This model follows CUT and employs the technique of contrastive learning for image-to-image translation, but this time in a dual learning setting. The authors believe that one embedding (used by CUT) may not be enough to capture the domain gap when the images of the two domains are not similar enough. For example, translating a horse into a zebra is relatively easy as it performs changes mostly on color. However, when translating a cat to a dog significant geometric changes should be applied. In order to improve the performance of contrastive learning in such cases, the authors suggest the use of two different encoders. These encoders learn separate embeddings for the two domains to maximize the mutual information between input and output image patches. The official code of DCLGAN can be found at https://github.com/JunlinHan/DCLGAN.

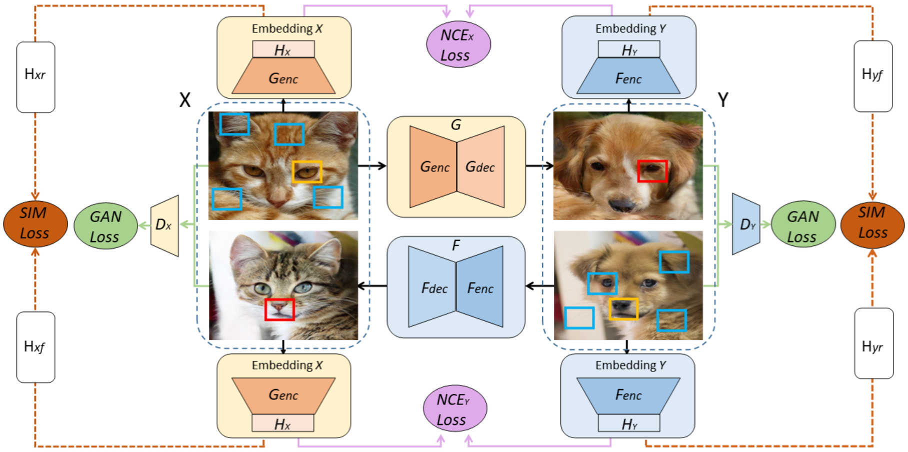


Figure 3.9: Illustration of the dual-contrastive learning setting of DCLGAN. Separate domain embeddings are extracted from each of the two generators. Source: [17]

DCLGAN uses two pairs of generators and discriminators like CycleGAN. Therefore, this model can also learn two mappings, $G : X \rightarrow Y$ and $F : Y \rightarrow X$, and translate images in both directions. The learning process of DCLGAN is very similar to CUT, with the main difference that it uses the first part of both its generators in order to map the image patches to the embedding space. Figure 3.9 illustrates this process. For example, the generator G is split into the encoder part G_{enc} and the decoder part G_{dec}. Then image features are extracted from four layers of the encoder which are then passed to a two-layer MLP head H_X producing the embedding for domain X. Similarly for the generator F, the encoder F_{enc} and the projection head H_Y produce the embedding for domain Y. Then the maximization of mutual information is done by formulating a classification problem with the target class as the positive pair of image patches and employing the same PatchNCE loss that CUT uses for contrastive learning. Additionally, the authors choose to apply the same identity loss regularization with CycleGAN and not the contrastive learning based one that CUT uses, because the first one is faster to compute. Finally, the SIMLoss (similarity loss) is used by an alternative version of DCLGAN called SimDCL. This version of the model aims to solve the problem of mode collapse that appears in some tasks, where the model keeps producing the same output independently of input. SIMLoss was not used in the experiments of this thesis, as no such problem was encountered.

3.4 Evaluation Protocol

This section describes the evaluation methods used in the experiments. Specifically, the translated results of each model were evaluated in two different ways. The first way includes common metrics used to evaluate the generated images of GANs, while the second way uses the translated images in the context of semantic segmentation and compares their performance on the Cityscapes validation set. As mentioned in section 3.1, the annotations of the Mapillary Vistas dataset are not directly compatible with the GTA V annotations. This is why only the translated images from GTA V to Cityscapes were considered in the segmentation task.

3.4.1 GAN Metrics

The evaluation of generated images in the context of unsupervised image-to-image translation is not an easy task as there is no ground truth of how an image should be translated. For example, it is not possible to know which is the correct translation of a cat into a dog. As a result, common metrics like the Structural Similarity Index (SSIM), Peak Signal to Noise Ratio (PSNR), or even a simple Mean Squared Error (MSE) can not be used, because they require paired images. The most straightforward way to evaluate unpaired translations is just by looking at them and scoring them manually. One common way of doing this is the Amazon Mechanical Turk (AMT) [7] perceptual studies, where participants are asked to assess the realism of the translated images. However, studies based on human subjectivity cannot provide a fair benchmark for comparing different approaches in research. For that reason, some works have tried to design metrics suitable for evaluating the quality of images without requiring a ground truth or human intervention. The three most commonly used such metrics are chosen for the experiments. Those are the Inception Score (IS), Fréchet Inception Distance (FID) and Kernel Inception Distance (KID). All three metrics use a pre-trained Inception v3 [41] on ImageNet for their calculation. Their PyTorch [35] implementation used in the experiments is available at `https://github.com/toshas/torch-fidelity` and their details are described below.

Inception Score (IS)

The inception score [40] is a metric used for evaluating the quality and variety of generated images by GANs. The first step of calculating the inception score is using the Inception v3 model to predict the class probabilities of each generated image and get the conditional label distribution $p(y|x)$. Y is a random variable with 1000 possible values since the classifier is pre-trained on ImageNet in this case. If an image has a distinct object included in the categories of ImageNet, then the probability of this label will be close to 1, while the others will be close to 0. This is a characteristic of a good quality image. Therefore, the conditional probability of all images in the collection should have low entropy. Additionally, the marginal probability distribution indicates how much variety there is in the generator's output. The higher the entropy of the integral of the marginal probability the higher the variety of the images is. The inception score is calculated using the Kullback-Leibler (KL) divergence between the conditional and marginal distributions and is expressed as:

$$\mathrm{IS} = \exp\left(\mathbb{E}_{\mathbf{x}} D_{KL}(p(y \mid \mathbf{x}) \| p(y))\right). \tag{3.10}$$

A higher inception score is better because then the two distributions are dissimilar. This means that each image has a distinct object which is classified with a high probability and that the collection of the generated images has a variety of labels (i.e. the images are diverse).

Although the authors find that the inception score correlates well with the human evaluation of image quality, this metric has the disadvantage that it is calculated only using the generated fake images and does not compare them in any way to the real images with which the generator was trained. Therefore, other metrics such as FID are usually preferred over IS for the evaluation of generated images.

Fréchet Inception Distance (FID)

The Fréchet Inception Distance [20] measures the Fréchet distance, also called Wasserstein-2 distance, between feature vectors of images extracted from a pre-trained Inception v3. This metric is designed to solve the limitation of IS and compares the statistics of generated images to the statistics of a collection of real-world images, while being more consistent and reliable. FID does not extract probabilities from the output layer like IS. Instead, for each image, it extracts a 2048 feature vector from the last pooling layer. This way, the mean m and the covariance matrix C are calculated for the collection of real images. Similarly, the mean m_w and the covariance matrix C_w are calculated for the generated images. Finally, FID is expressed as the distance between these two Gaussian distributions of real and generated images:

$$FID = \|\boldsymbol{m} - \boldsymbol{m}_w\|_2^2 + \operatorname{Tr}\left(\boldsymbol{C} + \boldsymbol{C}_w - 2\left(\boldsymbol{C}\boldsymbol{C}_w\right)^{1/2}\right). \tag{3.11}$$

A lower FID indicates better-quality images as that means that the statistics of real and generated images are similar and thus the synthetic images look more realistic.

Kernel Inception Distance (KID)

Kernel Inception Distance (KID) [3] is a metric similar to FID but is shown to be superior. Same with FID, it measures the difference between the distributions of generated and real-world images in the representation space of a pre-trained Inception v3 as a measure of image quality. Specifically, it calculates the Maximum Mean Discrepancy (MMD), between Inception feature vectors using a polynomial kernel:

$$k(x, y) = \left(\frac{1}{d} x^\top y + 1\right)^3, \tag{3.12}$$

where d is the representation dimension. According to the authors, KID has an unbiased estimator, unlike FID, which is more reliable for small datasets as it does not depend on the number of samples.

3.4.2 Semantic Segmentation

Apart from GAN metrics, semantic segmentation was also used to evaluate the transformation of GTA V images to look like Cityscapes images. The two datasets have compatible annotations, so it is easy to measure the segmentation performance when using translated data. The idea is that if the translated images are similar to images of the target domain, then the performance of semantic segmentation should be better than using the original images of the source domain. In this case, the chosen semantic segmentation model is DeepLabv3+ [10]. DeepLabv3+ was trained each time using translated GTA V images produced by one of the image-to-image translation models. Then, the segmentation was evaluated on the Cityscapes validation set. Finally, whichever model produced the data that gave the best results on the Cityscapes validation set is considered to be better at translating GTA V images. The metrics used for the evaluation of semantic segmentation are the per-pixel accuracy, per-class accuracy, mIoU and class IoU. These are described below, along with the details of DeepLabv3+.

DeepLabv3+

DeepLabv3+ [10] is a semantic segmentation model proposed by Google and builds upon previous DeepLab models [8, 9]. It follows an encoder-decoder architecture and one of its key features is the atrous separable convolution. The encoder module reduces the feature maps and provides rich semantic information with the help of atrous convolution and the decoder module refines the segmentation results along object boundaries in order to obtain sharper segmentations. This architecture is illustrated in Figure 3.10.

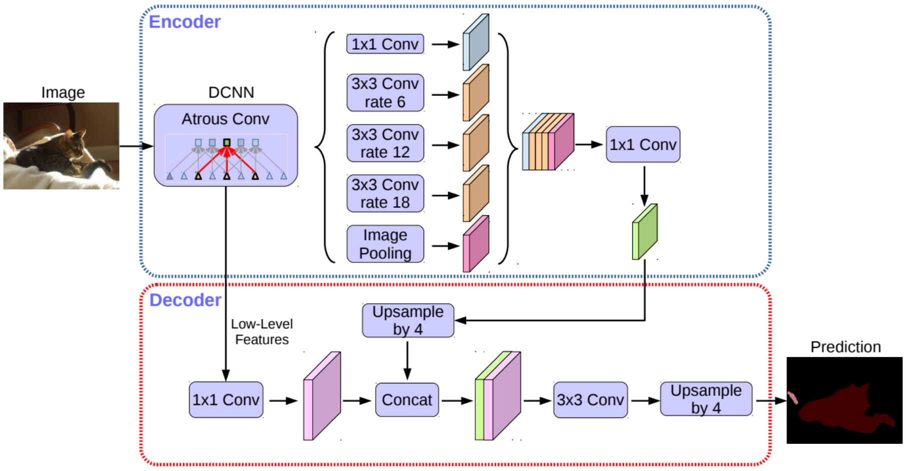


Figure 3.10: The encoder-decoder architecture of DeepLabv3+. Source: [10]

The atrous convolution is employed to extract features of deep convolutional neural networks at an arbitrary resolution. It can also control the filter's field-of-view to capture multiscale information. The atrous convolution is also known as dilated convolution or hole algorithm because it inflates the kernel by putting holes between its values. An additional parameter r called the atrous or dilation rate controls the number of zeros between two consecutive filter values along each spatial dimension. A rate of 1 corresponds to the standard convolution. Figure 3.11 shows an example of applying the atrous convolution with kernel size 3 × 3.

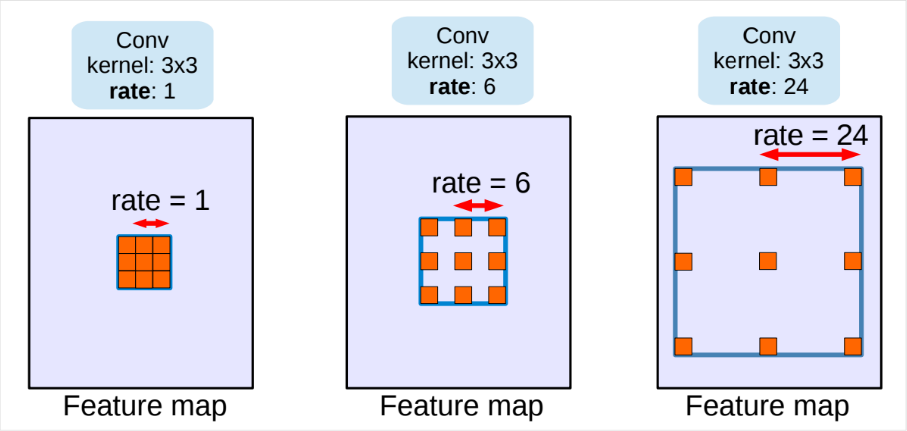
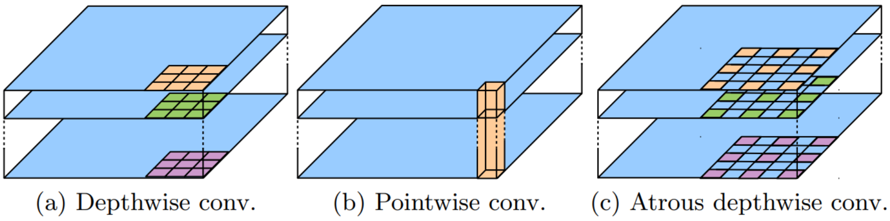


Figure 3.11: Example of atrous convolution with kernel size 3 × 3 and different rates. Source: [8]

DeepLabv3+ employs depthwise wise separable convolution in combination with atrous convolution to give the atrous separable convolution. The depthwise separable convolution decomposes a standard convolution into a depthwise convolution (applying a single filter for each input channel) and a pointwise convolution (combining the outputs from depthwise convolution across channels). The authors find that the atrous separable convolution significantly reduces the computational complexity of the model. This operation is shown in figure 3.12.



(a) Depthwise conv. (b) Pointwise conv. (c) Atrous depthwise conv.

Figure 3.12: The depthwise separable convolution decomposes a standard convolution into (a) a depthwise convolution and (b) a pointwise convolution. DeepLabv3+ uses atrous separable convolution where atrous convolution is adopted in the depthwise convolution, as shown in (c). Source: [10]

Segmentation Metrics

The most straightforward way to evaluate the performance of a model in image semantic segmentation is the per-pixel accuracy, which simply measures the percentage of pixels in the image that were correctly classified. However, this metric can provide misleading results in the case of an imbalanced dataset, as the per-pixel accuracy can be high even if the model classifies well only the dominant class and performs much worse on the less frequent classes. Alternatively, the per-class accuracy measures the ratio of correctly labeled pixels for each class and then averages over the classes.

Most commonly used is the Intersection-Over-Union (IoU), also known as the Jaccard Index. This metric measures the area of overlap between the predicted and target segmentation masks divided by the area of their union:

$$IoU = \frac{\text{target mask} \cap \text{predicted mask}}{\text{target mask} \cup \text{predicted mask}}. \tag{3.13}$$

For each class, the area of the intersection between the two masks is the area of the correctly classified pixels for that particular class, while the area of the union includes all the pixels of that class be it ground truth or predicted (either correctly or not). In terms of the confusion matrix, the intersection includes the true positives and the union includes the true positives plus the false positives plus the false negatives. This metric is more reliable than the accuracy even in the case of imbalanced classes, still, it also has a limitation. Although it measures the number of pixels that are common between the predictions and ground truth segmentation masks it does not necessarily evaluate how accurate the segmentation boundaries are. In the results, the IoU is reported for each class separately as well as globally (i.e. the mean of IoUs over the classes). These are called the class IoU and mIoU respectively.

Chapter 4

Results

This chapter presents both qualitative and quantitative results of the image-to-image translation task, evaluating the performance of the four models tested. Additionally, it offers comparisons on the semantic segmentation task by using the translated images of each model to train DeepLabv3+, while real-world images from the Cityscapes dataset were used to test the performance.

4.1 Image-to-Image Translation Results

In the experiments, all four models were trained using 256×256 random crops of the images and tested on full-sized 512×256 images. Additionally, a random horizontal flip was applied for data augmentation. The Adam optimizer was used with a learning rate of 0.0002 and the models were trained for up to 25 epochs. More training does not show any significant improvements, however, a common practice in literature is to apply a linear learning rate decay during additional training epochs as the training of these models is quite unstable. Instead, in this work, checkpoints were saved at the end of each epoch and the saved checkpoint with the best results was chosen for each model as they require a lot of time to be trained. Table 4.1 shows the number of parameters of each model and its training time for each dataset. CUT is the fastest and lightest to train, while DCLGAN is the slowest. The GPU used for training the models and reporting the training times is an NVIDIA RTX 3060 OC.

Figure 4.1 shows a few examples of GTA V images transformed to look like Cityscapes. All models are able to capture the color style of the 8-bit LDR images of the Cityscapes dataset. Additionally, they completely change the texture of the roads to match the smooth asphalt which most frequently appears in Cityscapes, something that traditional color transfer methods are not able to do. CycleGAN and AttentionGAN can also make vegetation look greener and more voluminous (2^{nd} image) or even try to remove haze from distant objects (4^{th} image). It is interesting to note that the models based on cycle-consistency learning (i.e. CycleGAN and AttentionGAN) can better preserve the geometry of the objects in the images. They can even preserve very fine details like letters in signs

Method	Parameters	Time (hours/epoch)	
		GTA V → Cityscapes	GTA V → Mapillary Vistas
CycleGAN	28.286 M	1.8	1.6
AttentionGAN	29.176 M	2.1	1.9
CUT	14.703 M	1.6	1.4
DCLGAN	29.274 M	2.6	2.3

Table 4.1: Comparison of training time and number of parameters (total of all networks) for each model.

(5th image). On the other hand, CUT and DCLGAN that use contrastive learning, apply more aggressive or unnecessary changes which lead to lower quality images. For example, both CUT and DCLGAN erroneously tend to populate the sky (or even buildings) with trees. A large amount of trees along the roadsides is indeed a characteristic of central European cities. Another common feature of the Cityscapes dataset is the Mercedes star on the hood of the car used to capture the images. Unfortunately, for that reason, all models tend to hallucinate star logos at the bottom of the images.

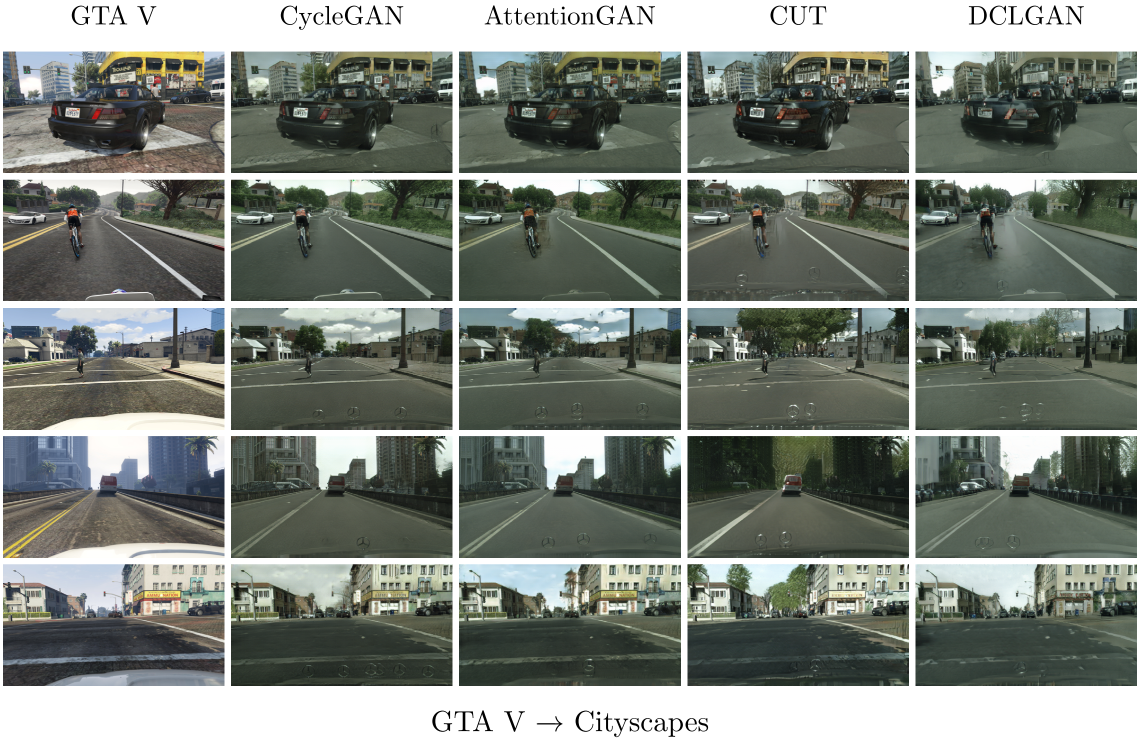
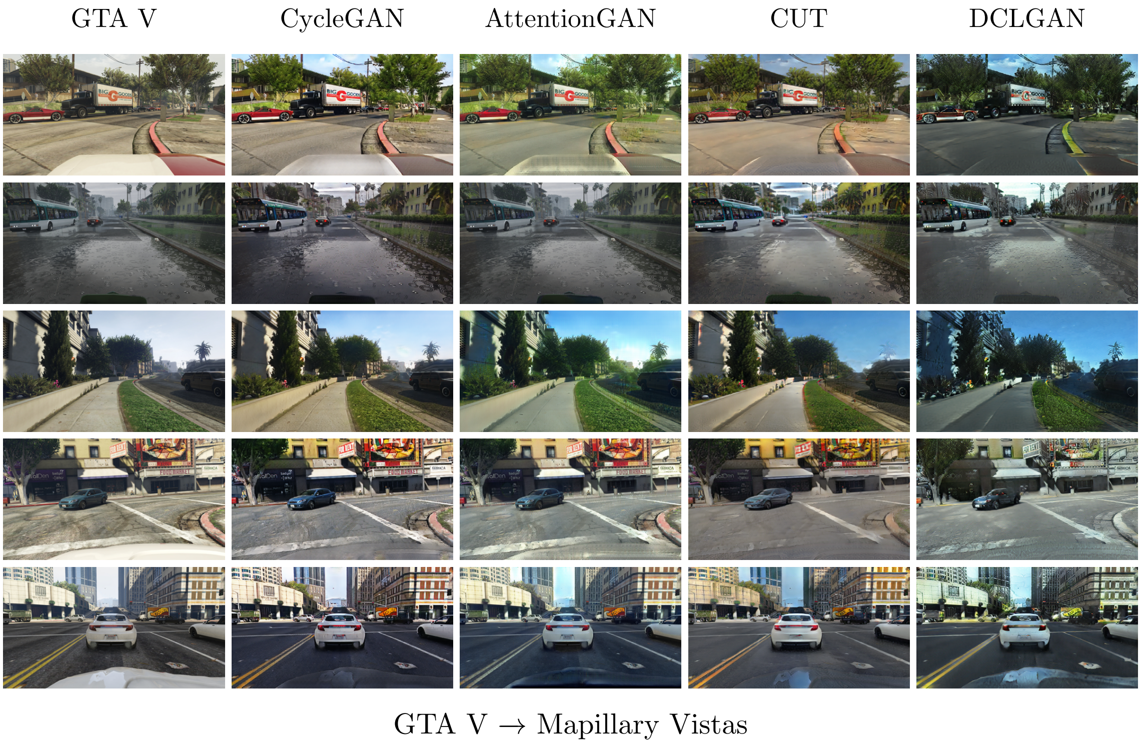


Figure 4.1: Example translations from GTA V to Cityscapes using four different models.

Similar changes can be seen in Figure 4.2 when translating GTA V images to Mapillary Vistas images. In this case, the models capture the bright and vibrant colors of the latter dataset. Similar to Cityscapes, CycleGAN and AttentionGAN produce higher quality images than CUT and DCLGAN, however, AttentionGAN applies minimal changes in many images as the Mapillary Vistas dataset is very diverse and the domain gap is relatively small. This frequently results in almost empty attention masks preserving almost the entire images intact (2^{nd} image). One common mistake that all models tend to make in this case is that they try to blend the hood of the car with the road. This is because the images of Mapillary Vistas are usually not taken from a camera mounted on the windshield of a car and as a result, GANs try to erase the hood which frequently appears in the GTA V dataset.


GTA V CycleGAN AttentionGAN CUT DCLGAN

GTA V → Mapillary Vistas

Figure 4.2: Example translations from GTA V to Mapillary Vistas using four different models.

Appendix A shows a few examples of side-by-side translations of the same GTA V images to Cityscapes and Mapillary Vistas using CycleGAN. Additionally, appendix C shows translation examples of two models tested at the beginning of the experiments, but not included in the analysis. Those are GANILLA [21], a variant of CycleGAN that introduces changes to the generator such as a feature pyramid network and LPTN [30], an approach to image translation using a laplacian pyramid network. When tested on full-size GTA V images GANILLA could not perform the translations and worked only with square images. LPTN was excluded because it introduced many halo-like artifacts in almost all images while not performing such striking texture changes.

Table 4.2 evaluates the performance of the models using quantitative metrics. Starting with IS, it seems that the original GTA V images have a quite high score and the best one compared to Cityscapes translations. That is problematic and makes this metric unreliable. As already mentioned, the fact that this metric does not compare generated images with real images is a serious disadvantage. FID and KID generally correlate well to each other. The performance on Mapillary Vistas translations is intuitive as CycleGAN has the best FID and KID and indeed generated the highest quality images, while CUT and DCLGAN have quite worse results. The same cannot be said in the case of Cityscapes translations, where strangely CUT has the best FID and KID values followed by DCLGAN. That may happen because DCLGAN and especially CUT tend to populate the sky, and sometimes buildings, with trees. This way, they manage to better approximate the statistics of the Cityscapes dataset, but at the cost of altering the semantic content of the images. FID and KID, however, cannot evaluate such mismatched scene structures giving a better score when they should not.

Method	GTA V → Cityscapes			GTA V → Mapillary Vistas		
	IS ↑	FID ↓	KID ↓	IS ↑	FID ↓	KID ↓
GTA V	4.263	103.988	0.079	4.263	48.390	0.035
CycleGAN	3.491	55.864	0.022	4.248	**30.226**	**0.015**
AttentionGAN	**3.659**	56.348	0.021	**4.353**	32.858	0.018
CUT	3.246	**49.998**	**0.015**	4.131	33.580	0.021
DCLGAN	3.069	54.541	0.020	3.867	36.423	0.022

Table 4.2: Quantitative evaluation of the translated images generated by different models. The performance of the original GTA V images is also provided for comparison.

4.2 Semantic Segmentation Results

In order to evaluate the quality of the GTA V images that were transformed to look like Cityscapes, DeepLabv3+ was trained separately using the translated images of each image-to-image translation model and then evaluated on the Cityscapes validation set. The PyTorch implementation of DeepLabv3+ used in the experiments is available at `https://github.com/VainF/DeepLabV3Plus-Pytorch`. The backbone (i.e. the encoder) used was a ResNet-50 [19] model pre-trained on ImageNet. DeepLabv3+ was trained each time using the SGD optimizer with a polynomial learning rate decay scheduler starting from a learning rate of 0.01. For data augmentation, a random horizontal flip and random color jitter (brightness, contrast, and saturation) were applied. The model was trained using a batch size of 16 random 256×256 crops. The low resolution of the images constrained by the image-to-image translation models hurts the performance of the semantic segmentation, especially on small objects, and is a limitation of this work.

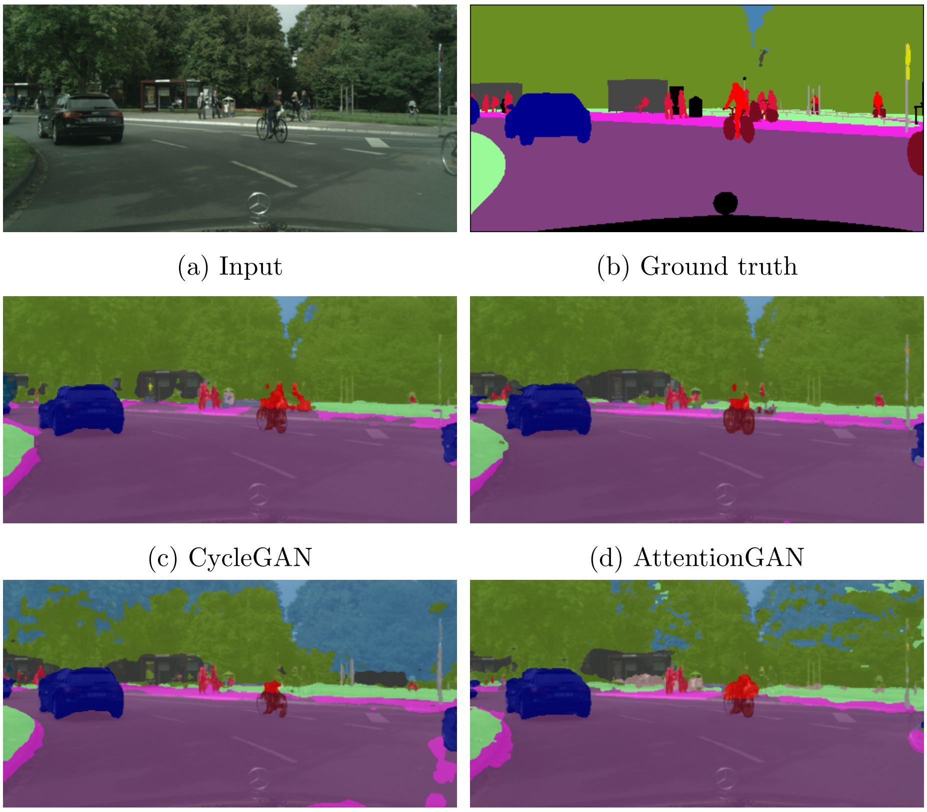
Table 4.3 evaluates the overall performance of DeepLabv3+ using different training data. The translated GTA V images of CycleGAN and AttentionGAN help DeepLabv3+ perform better than using the original GTA V images, while the translated images of CUT and DCLGAN worsen the performance. That happens because, as already mentioned, CUT and DCLGAN make more mistakes than CycleGAN and AttentionGAN, which alter the semantic content of the images. This becomes clear when looking at the IoUs of each class separately at table 4.4. The performance of DeepLabv3+ using images of CUT and DCLGAN is lower on the 'sky' and 'vegetation' classes. The model seems to get confused because CUT and DCLGAN tend to generate trees in the sky where they do not exist. Figure 4.3 shows such an example, where DeepLabv3+ predicts some areas of sky although those pixels belong to vegetation. According to the performance in semantic segmentation, CycleGAN generates the highest quality images followed by AttentionGAN. However, all models are able to translate effectively roads and sidewalks as the corresponding class IoUs show a significant increase compared to the original GTA V images.

Data	Per-pixel accuracy	Per-class accuracy	mIoU
Cityscapes	92.79%	70.02%	60.65%
GTA V	83.26%	46.68%	36.44%
CycleGAN	86.33%	**49.01%**	**38.40%**
AttentionGAN	**86.49%**	48.80%	37.96%
CUT	82.69%	44.95%	33.64%
DCLGAN	83.59%	45.82%	34.30%

Table 4.3: Semantic segmentation performance of DeepLabv3+ on the Cityscapes validation set when trained with the translated GTA V images of different models. The performance when trained with the original Cityscapes and GTA V images (1st and 2nd row) is also provided for comparison.

Class	Cityscapes	GTA V	CycleGAN	AttentionGAN	CUT	DCLGAN
road	96.55	82.13	90.98	91.21	89.29	89.70
sidewalk	73.37	31.06	43.41	43.17	40.10	41.19
building	87.04	79.75	79.74	80.25	74.74	76.48
wall	44.39	24.52	23.64	29.23	22.47	17.81
fence	36.06	12.24	16.89	12.93	11.19	14.32
pole	37.55	19.64	22.32	23.73	20.38	22.95
traffic light	37.93	17.92	21.36	23.94	19.52	17.37
traffic sign	53.13	3.01	9.58	6.73	2.83	8.04
vegetation	87.62	78.63	77.74	78.12	68.09	70.91
terrain	54.69	34.35	34.29	37.00	32.38	26.39
sky	91.92	84.80	77.35	74.66	59.52	65.36
person	62.62	45.77	46.73	44.91	43.84	43.21
rider	37.24	15.49	19.80	12.84	11.14	16.17
car	88.56	76.22	79.27	79.77	75.37	74.82
truck	54.83	33.47	31.45	27.20	26.42	28.88
bus	65.85	29.60	30.34	28.57	10.70	17.88
train	50.97	0	0	0	0	0
motorcycle	33.49	16.13	17.49	16.43	21.43	12.88
bicycle	58.59	7.63	7.23	10.49	9.77	7.32

Table 4.4: The class IoU scores of DeepLabv3+ when trained with the translated GTA V images of different models. The scores for the original Cityscapes and GTA V images are also provided for comparison.



(a) Input

(b) Ground truth

(c) CycleGAN

(d) AttentionGAN

(e) CUT

(f) DCLGAN

Figure 4.3: Example semantic segmentation predictions of DeepLabv3+. (a) Input image from the Cityscapes validation set. (b) Ground truth label map. (c), (d), (e), (f) Overlay predictions of DeepLabv3+ when trained with the translated GTA V images of the corresponding models.

Chapter 5

Conclusions

This thesis explores the problem of transforming images from the game Grand Theft Auto V (GTA V) to look like realistic urban scenes. This task falls under the category of unsupervised image-to-image translation as there is no ground truth that exactly maps GTA V images to real-world images. The related work on this problem, presented in chapter 2, served as the main motivation of this thesis. In practice, a framework can be built around image-to-image translation by leveraging additional information such as semantic label maps or intermediate graphic buffers (G-buffers) produced by the game's engine. This can improve the quality of the results and can also serve as a method for domain adaptation.

In this work, four state-of-the-art unsupervised image-to-image translation models were trained to translate GTA V images to match the style of two different real-world datasets. CycleGAN and AttentionGAN are based on cycle-consistency learning, while CUT and DCLGAN leverage contrastive learning. Cycle consistency serves as a constraint and that may be good or bad depending on the situation. Contrastive learning allows more variety to the translated images and performs much better when images of the two domains are not very similar. However, the domain gap between GTA V images and real-world data is relatively small as GTA V already looks realistic enough. The results show that in this case, the bijection constraint enforced by cycle consistency is more than welcome as the generated images can retain detailed geometries and even letters (e.g. on store signs). On the other hand, models based on contrastive learning make more aggressive changes that distort images and even make more mistakes like adding non-existent trees to the sky. When evaluating the quality of the images in such cases, the performance in semantic segmentation was a more reliable measure than GAN metrics like FID and KID. That is because those metrics cannot identify mismatches between the structure or semantic content of two images.

The translated images, although far from perfect, are impressive and show that CylceGAN and AttentionGAN perform better on this particular image-to-image translation task. Therefore, a future direction of this thesis could be to use one of these models as the core mechanism of a framework to perform more effective domain adaptation for semantic segmentation, by using not only semantic label maps as in [22], but also G-buffers to improve the final quality, following [37]. However, even CycleGAN and AttentionGAN make mistakes due to the mismatch between the structure of the GTA V sceneries of Los Santos (the fictional city appearing in GTA V) and images from real-world datasets. Since Los Santos is based on the real city of Los Angeles, it would be interesting to create a dataset with images of the latter and train again image-to-image translation models in an unsupervised manner. Feeding data to the networks that are closer to what the results should ideally be, could maybe improve the realism of the translated GTA V images.

Bibliography

[1] G. Antipov, M. Baccouche, and J.-L. Dugelay. Face aging with conditional generative adversarial networks. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2089–2093, 2017.

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017.

[3] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton. Demystifying mmd gans, 2021.

[4] R. W. Brislin. Back-translation for cross-cultural research. *Journal of Cross-Cultural Psychology*, 1(3):185–216, Sept. 1970.

[5] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognit. Lett.*, 30:88–97, 2009.

[6] D. Brubacher. Detours: Binary interception of win32 functions. In *Windows NT 3rd Symposium (Windows NT 3rd Symposium)*, Seattle, WA, July 1999. USENIX Association.

[7] M. Buhrmester, T. Kwang, and S. D. Gosling. Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, 6(1):3–5, 2011. PMID: 26162106.

[8] L. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017.

[9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018.

[10] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In V. Ferrari,

M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Computer Vision – ECCV 2018*, pages 833–851, Cham, 2018. Springer International Publishing.

[11] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[12] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[13] 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.

[14] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *AAAI*, 2018.

[15] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

[16] J. Guo, S. Lu, H. Cai, W. Zhang, Y. Yu, and J. Wang. Long text generation via adversarial training with leaked information. In *AAAI*, 2018.

[17] J. Han, M. Shoeiby, L. Petersson, and M. A. Armin. Dual contrastive learning for unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2021.

[18] D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T.-Y. Liu, and W.-Y. Ma. Dual learning for machine translation. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 820–828, Red Hook, NY, USA, 2016. Curran Associates Inc.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[20] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[21] S. Hicsonmez, N. Samet, E. Akbas, and P. Duygulu. Ganilla: Generative adversarial networks for image to illustration translation. *Image and Vision Computing*, page 103886, 2020.

[22] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1989–1998. PMLR, 10–15 Jul 2018.

[23] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017.

[24] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017.

[25] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 694–711, Cham, 2016. Springer International Publishing.

[26] B. Karlsson. Renderdoc. https://renderdoc.org.

[27] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2019.

[28] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[29] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, 2017.

[30] J. Liang, H. Zeng, and L. Zhang. High-resolution photorealistic image translation in real-time: A laplacian pyramid translation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[31] M. Mirza and S. Osindero. Conditional generative adversarial nets. *ArXiv*, abs/1411.1784, 2014.

[32] G. Neuhold, T. Ollmann, S. R. Bulò, and P. Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5000–5009, 2017.

[33] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*, 2020.

[34] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[36] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2016.

[37] S. R. Richter, H. A. AlHaija, and V. Koltun. Enhancing photorealism enhancement. *arXiv:2105.04619*, 2021.

[38] S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *European Conference on Computer Vision (ECCV)*, volume 9906 of *LNCS*, pages 102–118. Springer International Publishing, 2016.

[39] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016.

[40] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 2234–2242, Red Hook, NY, USA, 2016. Curran Associates Inc.

[41] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[42] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. *ArXiv*, abs/1611.02200, 2017.

[43] H. Tang, H. Liu, D. Xu, P. H. Torr, and N. Sebe. Attentiongan: Unpaired image-to-image translation using attention-guided generative adversarial networks. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2021.

[44] H. Tang, D. Xu, N. Sebe, and Y. Yan. Attention-guided generative adversarial networks for unsupervised image-to-image translation. In *International Joint Conference on Neural Networks (IJCNN)*, 2019.

[45] A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding, 2019.

[46] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[47] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017.

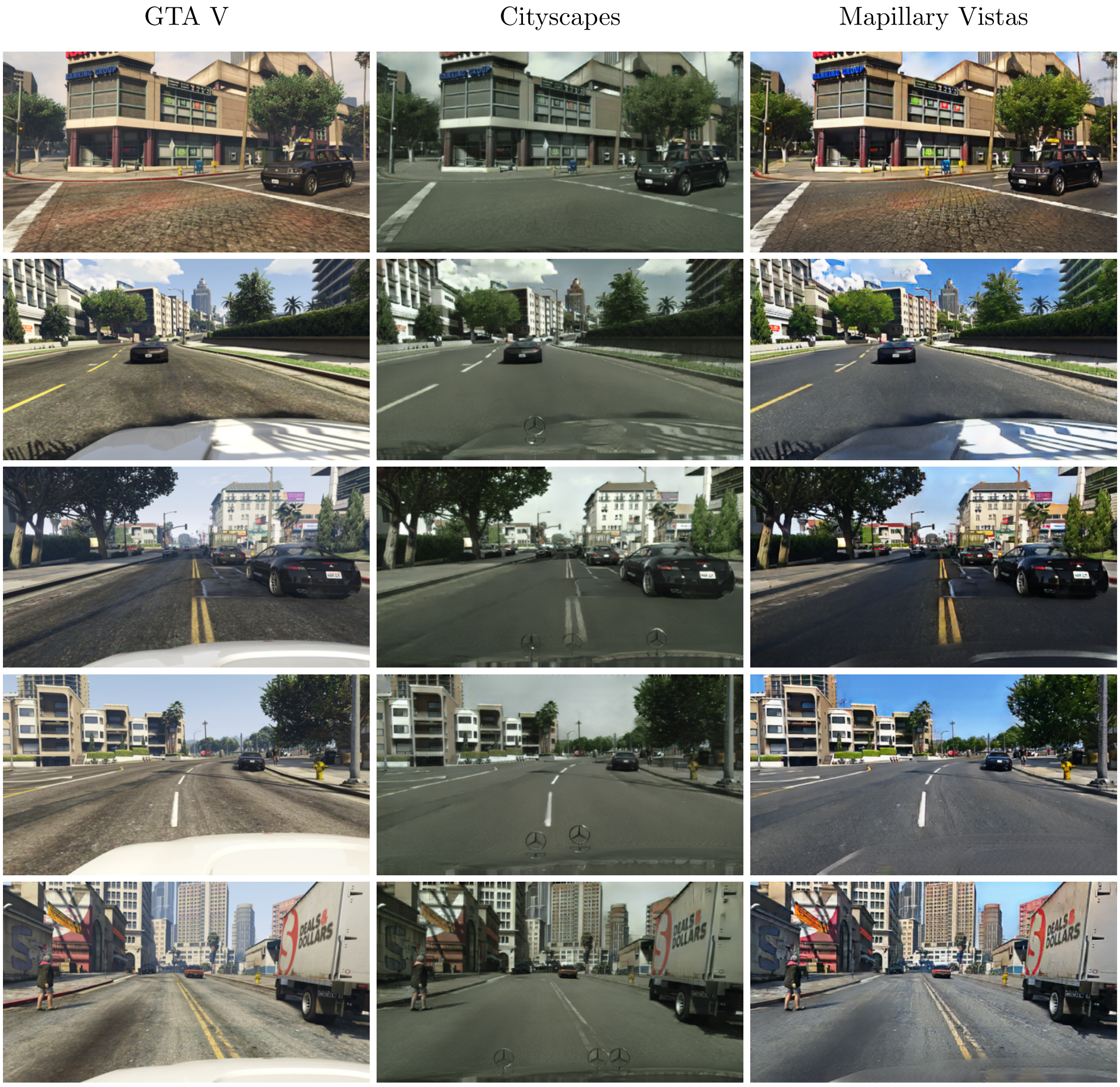
[48] L. Zhang, C. Li, T.-T. Wong, Y. Ji, and C. Liu. Two-stage sketch colorization. *ACM Transactions on Graphics*, 37(6), Nov. 2018.

[49] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[50] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Appendix A

Additional Results



GTA V

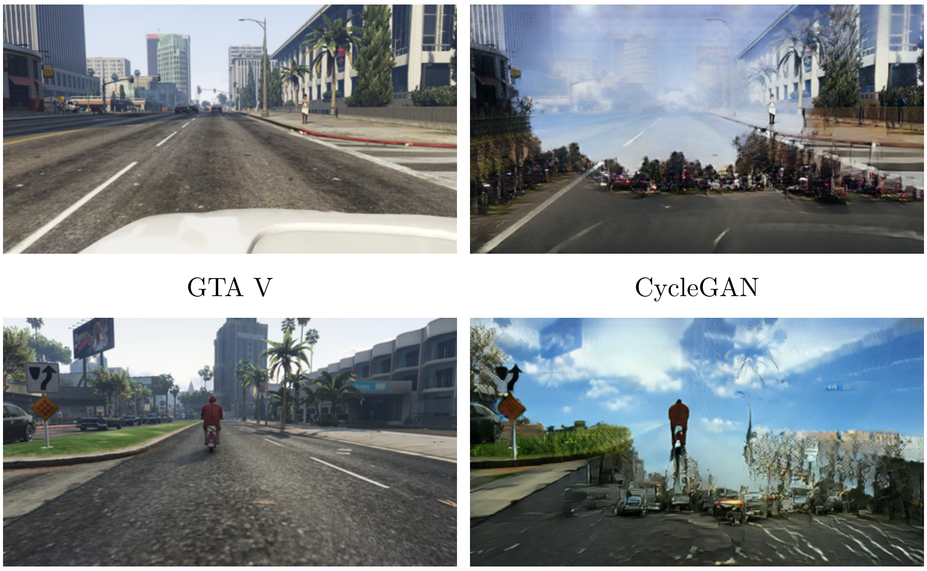
Cityscapes

Mapillary Vistas

Figure A.1: Side-by-side comparison of GTA V images transformed to match the Cityscapes and Mapillary Vistas styles using CycleGAN.

Appendix B

Failures on Uncropped Mapillary Vistas



GTA V

CycleGAN

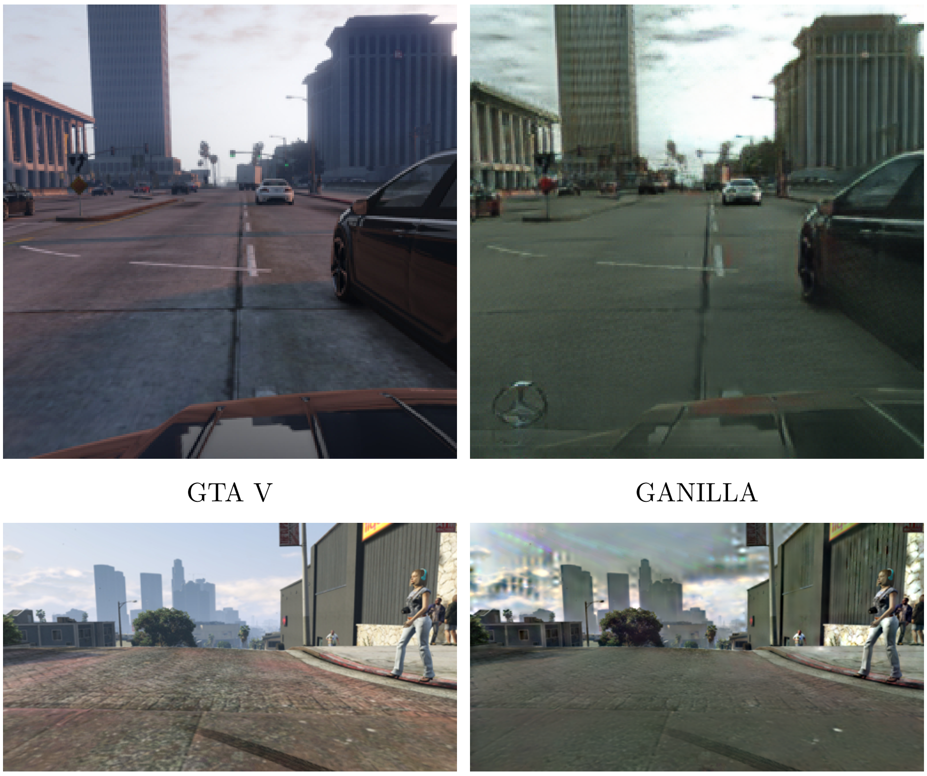
GTA V

CUT

Figure B.1: When not cropping Mapillary Vistas images as a pre-processing step to reduce sky volume, the image-to-image translation models often try to erase objects and replace them with the sky, resulting in images that look more like stairways to heaven rather than urban scenes.

Appendix C

Example Results of Other Models



GTA V

GANILLA

GTA V

LPTN

Figure C.1: In early experiments, two additional image-to-image translation models were tested, but are not included in the analysis. GANILLA [21] only worked on square images and not on the full-size GTA V images. LPTN [30] introduced many holo-like artifacts in most images, while not performing so striking texture changes. The samples shown in this figure are translations from GTA V to Cityscapes.