# toon2real: Translating Cartoon Images to Realistic Images

A thesis
Submitted in partial fulfillment of the requirements for the Degree of
Bachelor of Science in Computer Science and Engineering

Submitted by

| | |
|---|---|
| **KM Arefeen Sultan** | **150104111** |
| **Md. Nahidul Islam** | **150104127** |
| **Sayed Hossain Khan** | **150104133** |
| **Labiba Kanij Rupty** | **150104147** |

Supervised by

**Mohammad Imrul Jubair**
Assistant Professor
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

**Department of Computer Science and Engineering**
**Ahsanullah University of Science and Technology**

Dhaka, Bangladesh

June 2019

# CANDIDATES' DECLARATION

We, hereby, declare that the thesis presented in this report is the outcome of the investigation performed by us under the supervision of Mohammad Imrul Jubair, Assistant Professor, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh. The work was spread over two final year courses, CSE4100: Project and Thesis-I and CSE4250: Project and Thesis-II, in accordance with the course curriculum of the Department for the Bachelor of Science in Computer Science and Engineering program.

It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

KM Arefeen Sultan
150104111

Md. Nahidul Islam
150104127

Sayed Hossain Khan
150104133

Labiba Kanij Rupty
150104147

# CERTIFICATION

This thesis titled, **“toon2real: Translating Cartoon Images to Realistic Images”**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in June 2019.

**Group Members:**

| | |
|---|---|
| **KM Arefeen Sultan** | **150104111** |
| **Md. Nahidul Islam** | **150104127** |
| **Sayed Hossain Khan** | **150104133** |
| **Labiba Kanij Rupty** | **150104147** |

Mohammad Imrul Jubair
Assistant Professor & Supervisor
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Prof. Dr. Kazi A Kalpoma
Professor & Head
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

# ACKNOWLEDGEMENT

First and foremost, we are grateful to Almighty Allah for blessing us with the good health and well being we required to work on this thesis.

Next, we are really thankful to our beloved supervisor, Mohammad Imrul Jubair, without whose assistance we couldn't have achieved what we have achieved so far. It was his motivation and constant support that kept us going.

We also want to take this opportunity to express our gratitude to Prof. Dr. Kazi A. Kalpoma, Head of the department and all other faculty members and staffs of the Department of CSE, AUST, who have believed in us and helped and encouraged us in every possible way they could.

We also want to thank the external examiners of our thesis, Md. Taksir Hasan Majumdar and Mir Tafseer Nayeem, who scrutinized our work and showed us proper guideline.

Last but not least, we want to thank the people who made us come this far, our parents. We want to thank them for always being there for us and guiding us.

Dhaka
June 2019

KM Arefeen Sultan

Md. Nahidul Islam

Sayed Hossain Khan

Labiba Kanij Rupty

# ABSTRACT

In terms of Image-to-image translation, Generative Adversarial Networks (GANs) has achieved great success even when it is used in the unsupervised domain. In this work, we aim to translate cartoon images to photorealistic manifold using GAN. We apply several state-of-the-art models to perform this task; however, they fail to perform good quality translations. We observe that shallow difference between these two domains causes this issue. Based on this idea, we propose a method *toon2real*, based on CycleGAN model for image translation from cartoon domain to photorealistic domain. To make our model efficient, we implemented Spectral Normalization which added stability in our model. We demonstrate our experimental results and show that our proposed model has achieved the *lowest Fréchet Inception Distance score* and better results compared to other state-of-the-art techniques, such as UNIT. We also took help of human evaluation system where our output were given 4.08 out of 5.

# Contents

# List of Figures

# List of Tables

# Chapter 1

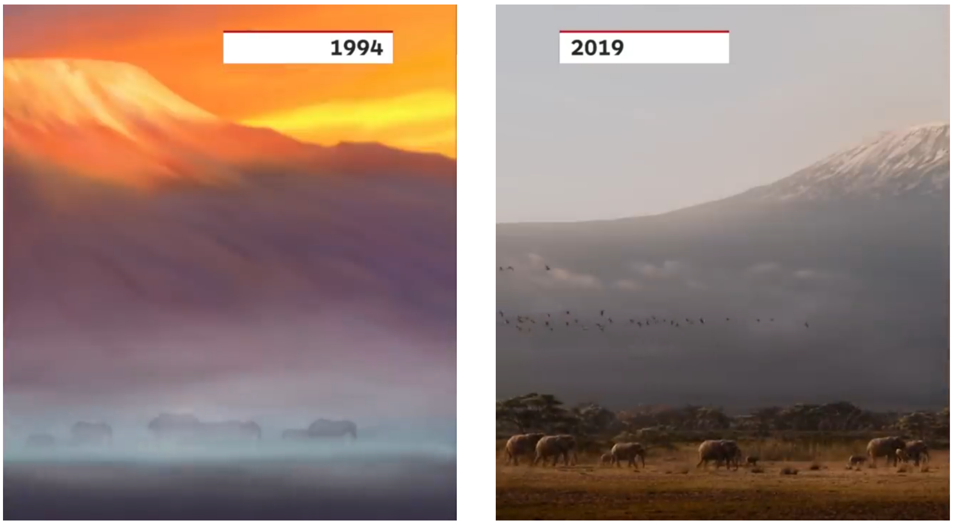
# Introduction

## 1.1 Research Domain

Movies serve as one of the most popular sources of entertainment for human beings. Cartoons, undeniably, held a large part of entertainment industry in this modern day world. While watching them, a curiosity might be prompted in our mind: *How enchanting it would be to see our favourite cartoons become realistic? What if the adventure of Chihiro from 'Sprited Away (2001)' is rendered in a real-life setup? Or the journey of Carl from 'Up (2009)'?*

We reckon the above fantasy crosses most of the cartoon lovers' minds once in a while; however, making this happen in reality is not an easy task. For instance, an upcoming movie *'The Lion King (2019)'*—remake of one of the most popular animated movies *'The Lion King (1994)'*—costs four times the original one [5,6][See Figure 1.1]. The reason is that the new movie is a live-action version of the animated one based on Computer-Generated Imagery (CGI) which is a costly task to perform [7]. Moreover, the time and the labour required to generate an image are also high.

In this paper, we consider the above subject as our research problem and we attempt to propose a time & cost effective solution. We aim to input a cartoon image and to produce its realistic version automatically. Hence, we present a technique called ***"toon2real"***—a Generative Adversarial Networks (GANs) [8] based approach—that translates cartoons to realistic images. There has been some tremendous researches on image-to-image translation using GANs such as [3, 9–12]; however, to our knowledge there hasn't been any research on generating realistic images from cartoon images yet. The closest research on translating cartoon images to realistic images has been touched by *Li etal* [13] which only covers the face generation part of the task. Besides, *Tomei et al* translates art images to realistic domain in their work [14], where each object of an image from the cartoon is mapped with the same objects from images of realistic domain. Moreover, the CartoonGAN [15]—a motivation



(a) (b)

Figure 1.1: Here, we can see a scene of *The Lion King(1994)* along with the scene from *The Lion King(2019)* which is made using CGI.

behind our work—converts real image to cartoon; performing contrariwise is not a solution to our problem as the detail preservation from real to cartoon is not similar for vice versa. Hence, translating cartoons to realistic images is much harder because the cartoon images are smoothed out and their details are very trivial while compared to realistic images.

In this paper, we apply a technique which is based on CycleGAN [3] to achieve desired goal. We demonstrate our results and, in addition, we compare it with the UNIT method used by *Liu et al* [10].

In the next section, we will discuss more on how we came to take decision to work on generating cartoon images from real images.

## 1.2 Motivation

Ever since **Ian Goodfellow** published his first paper on GAN [8], research on this field has taken a huge spike. Since then, GANs have achieved great results in various image generation tasks, which are image super-resolution [16], image-to-image translation [3], [9], [4], text-to-image synthesis [17], [18] etc.

Among all, Image-to-Image translation has reached another dimension with the help of GANs. There has been tremendous activies on this field. There have been various researches which generate dogs from cats or man from woman[See 1.2]. Recently, a research on super resolution paper has opened the door for recreating the once beloved games of 90s' again into high resolution game. From Figure. 1.3, we can see the example where images from two old games are translated into higher resolution. Another research which has created a new kind of branch of GAN is *Karras et al* [2]. This paper works on generating realistic human faces. It just not only works on that. It creates faces by merging faces from different people. From Figure. 1.4, we can see that, a image of face is created by using three diffent

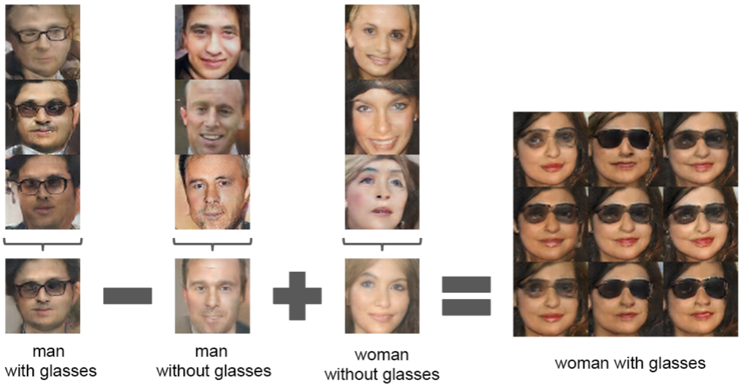
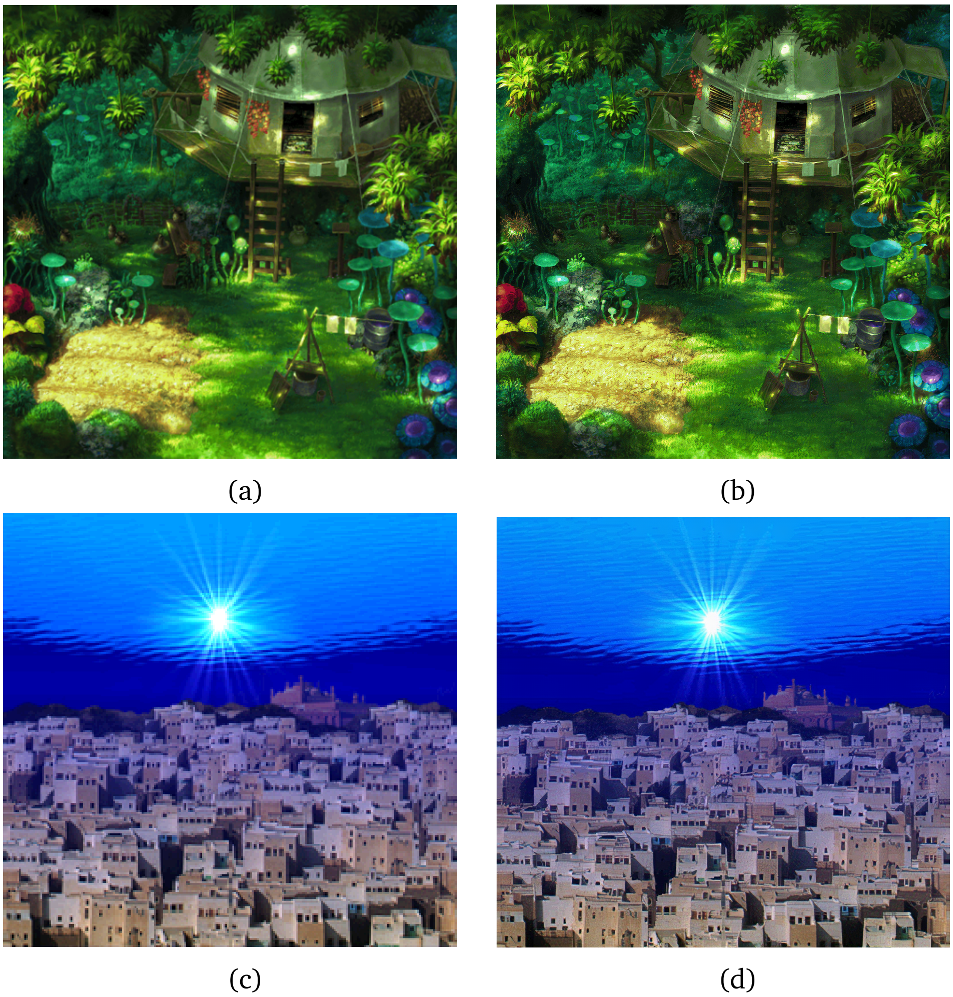


Figure 1.2: New women face are generated by learning from image distribution of **Man with glasses**, **man without glasess** and **woman without glasses**. [1]


(a)

(b)

(c)

(d)

Figure 1.3: Low resolution old games are translated into high resolution images. Here, Figure 1.3a is the old resulation game which is translated into high resolation version in Figure 1.3b. Same is done for Figure 1.3c and Figure 1.3d respectively.

images of face.

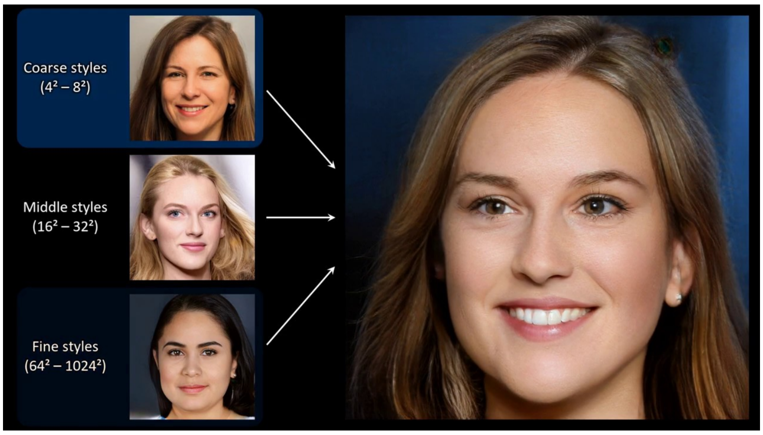


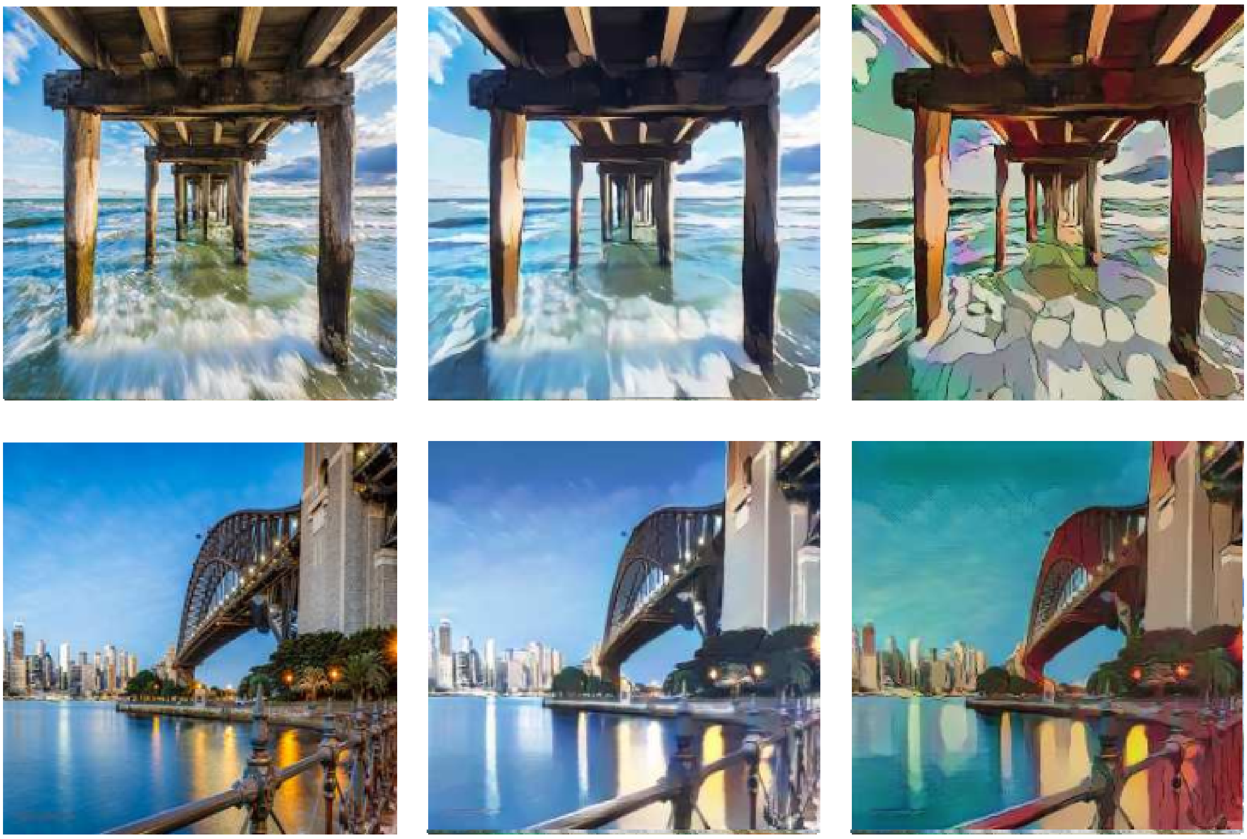
Figure 1.4: A new face is created by using three different faces. StyleGAN technique made it possible. [2]

One of the first researches which translated unpaired images successfully was [3]. It uses the technique of Cycle Consistency Loss. One of the main motivation of coming up with our research was when we learned about CycleGAN and how amazing it is. Before the research on it, image-to-image translation of unpaired images was out of reach. Later after the research of cyclegan, there have been some tremendous researches going on this field.

Using the technique, there has been another paper which is focused on the technique of generating Photo real images from cartoon images. This technique is uses an edge smoothing filter to create sharp edges of cartoon from photo-real images. We can see examples of real to cartoon translation of this paper from Figure. 1.5, where this technique is applied for *two* different styles.

Learning about this paper has influenced us a great deal to work cartoon to real translation. However, ours is not as straightforward as this one because — Cartoons have less details than photo real images. So, while translating from real images to cartoon images, it doesn't bother the model for some lack of image detail. However, in our case, we need to translate images of less details into images of high details, which makes our task difficult.

In the next section, we will discuss more on how we overcame the difficulties while translating cartoon images to photo real images.



(a) Real-world image.

(b) Shinkai style transformation.

(c) Hayao style transformation.

Figure 1.5: CartoonGAN: This technique is applied on 2 images for 2 different style — Shinkai and Hayao Style where (a) is input image and (b), (c) are their **Shinikai** and **Hayao** version respectively.

# 1.3 Contribution

## 1.3.1 Training Necessaries

As we need large dataset to train generative adversarial networks along with atleast *two* deep networks, it demands the need of heavy computation and heavy training time. So, to train our model, we used `Nvidia GeForce GTX 1060` with a dedicated ram of 6*GB*. It took us almost 5 days to train CycleGAN model and 3 days to train UNIT model.

## 1.3.2 Dataset Development

Previously, pix2pix [19] model was used to learn a mapping from input to output images using paired dataset. It used conditional adversarial network to transform images from one domain to another. Also similar works were generating photos from sketches [20] or from

semantic structures [21]. Though they generated great results, obtaining paired dataset was arduous and time-consuming. So we used unpaired dataset for our thesis work.
As deep learning is data hungry, initially, for *realistic* domain, we scraped scenery images from *Flickr* and many other sources which were tagged as *scenery, sunrise, sunset, sea, sky* & *beach* and collected around 7*K* samples. Besides, for *cartoon* domain, we extracted images from various Japanese anime movies. We extracted the scenery images from these movies consisting of sunsets, sea, sky, trees etc. We excluded the frames which are darker to see, and the first and last few frames—as the introductory and credits part tend to be textual in a movie. After hand-picking the appropriate images, in order to approximate with the size of the *realistic* domain, we collected images from more than 15 cartoon movies and clips, consisting the genres of *romance, spiritual, war, supernatural* & *science-fiction*. For both the domain, images were of 128 × 128 dimension.

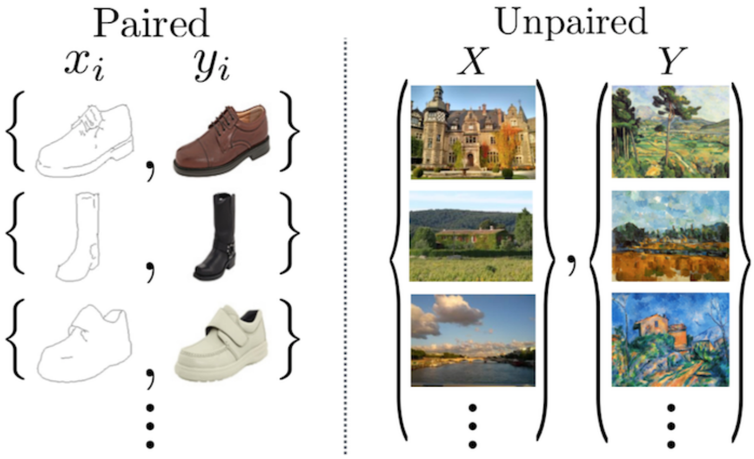


Figure 1.6: *Paired* training data on left and *unpaired* training data on right. Paired training data consists of correspondence between $x_i$ and $y_i$. For our thesis, we use unpaired training dataset where there is no correspondence between two sets.

### 1.3.3 Approaches for our work

Our thesis consists of training 7*k* samples and finally, with that trained model, we transform a cartoon image to a real world image domain. The steps we explored are described below and shown in Fig 1.7.

- We use CycleGAN [3] for cartoon-to-real world image transformation.
- For weight normalization, we exploited Spectral Normalization technique introduced by Miyato et al. [22].
- Finally we compared our work with another state-of-the-art model UNIT [4]. We show that our work achieves better result than UNIT framework.

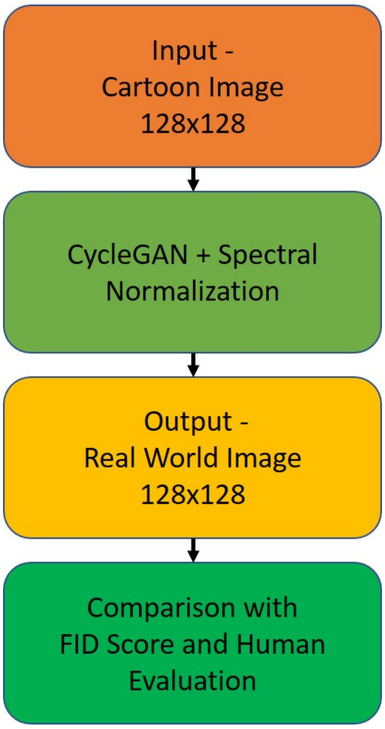


Figure 1.7: Process model for our thesis. We provide an input(128 × 128 image for the CycleGAN model. For the weight normalization technique we combine spectral normalization. Finally the result images are compared with *FID* score and human evaluation.

We published our work on *International Conference on Innovation in Engineering and Technology(ICIET)* 2018 as a poster paper. We also stood $1^{st}$ *runner-up* in *MindSparks '19* for poster presentation.

## 1.4 Thesis Organization

In this research, we have explored the idea of generating photo-real images from cartoon images and for that, we used methods such as UNIT, CycleGAN etc. We discussed about our exploration in the following sequence:

- Firstly, in Chapter 2, we discuss briefly about GANs and its variations.
- Later, in Chapter 3, we discuss about our method of work and its sequence.
- In Chapter 4, we discuss about our results and compare it with other techniques. We also evaluated our results in this section.
- Lastly, on Chapter 5, we take a look at our limitations and discuss about those. We also discussed about what we are planning to do in future.

In the next chapter we start our study with the brief introduction of Generative Adversarial Networks.

# Chapter 2

# Background

In this chapter, we demonstrate the in-depth contents about GAN [8], CycleGAN [3], UNIT [4] and CartoonGAN [15]. Finally we discuss about the spectral normalization technique [22] we implemented in contrast with CycleGAN [3].

## 2.1 Generative Adversarial Network

Generative Adversarial Network(GANs) [8] has become a global phenomenon in deep learning algorithms. The algorithm uses two networks, Generators and Discriminators, in a minimax algorithm situation where both of them tries to outperform another in a significant task e.g. image generation [1] [23], image editing [24], text2image [17], image inpainting [25], image-to-image translation tasks [3] [19] etc.

**Total Loss:** Full objective of GAN function is:

$$\min_{G}\max_{D} V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{2.1}$$

where Generator G tries to generate images, whereas discriminator D discriminates the output whether it is fake or real. In the objective function, Generator tries to maximize the value of $D(G(z))$ such that it can fool the discriminator,and thus the gap between real and fake becomes minimum. The discriminator tries to maximize the term of $[logD(x)]$ and for the 2nd term of $[log(1-D(G(z))]$, discriminator tries to minimize it to 0, which means that the discriminator tries to recognize if the image is generated or real.

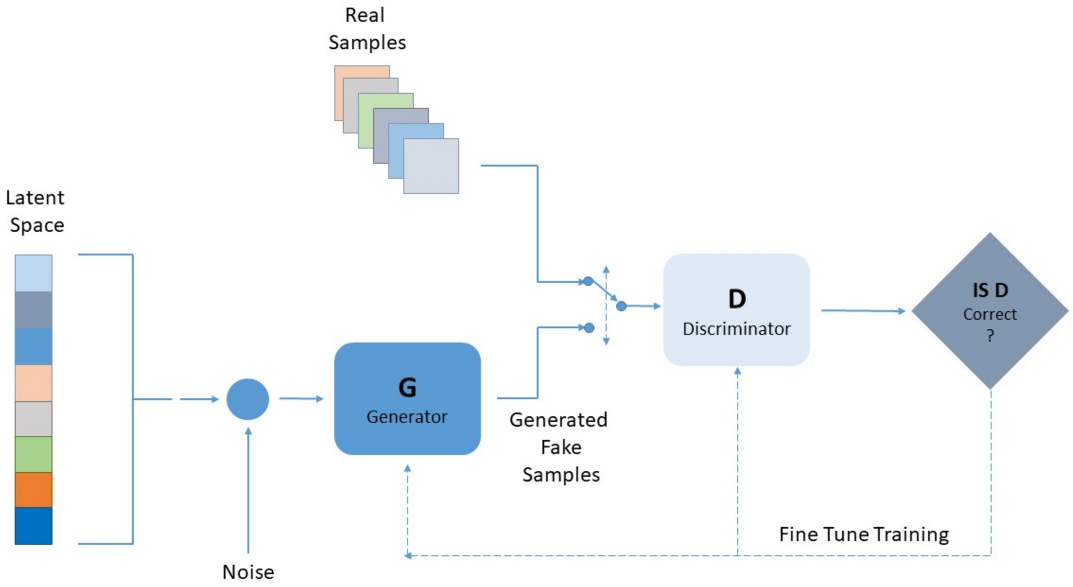


Figure 2.1: High level representation of generative adversarial networks. *G* stands for generative network and *D* for discriminative. The generator tries to generate image from random noise and the discriminator probes if the result image is true or fake? Both of them continuously work together to assemble a real sample.

Algorithm 1 optimizes the equation of GAN function given in 2.1.

**Algorithm 1:** Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, $k$, is a hyperparameter. $k = 1$ is chosen.

```
for number of training iterations do
    for k steps do
        • Sample minibatch of $m$ noise samples $(z^{(1)}, ..., z^{(m)})$ from noise prior $p_g(z)$.
        • Sample minibatch of $m$ examples $(x^{(1)}, ...x^{(m)})$ from data generating distribution $p_{data}(x)$.
        • Update the discriminator by ascending its stochastic gradient:
              $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} [log D(x^{(i)}) + log(1 - D(G(z^{(i)})))]$    (2.2)
    end for
        • Sample minibatch of $m$ noise samples $(z^{(1)}, ..., z^{(m)})$ from noise prior $p_g(z)$.
        • Update the generator by descending its stochastic gradient:
              $\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} log(1 - D(G(z^{(i)}))).$    (2.3)
```

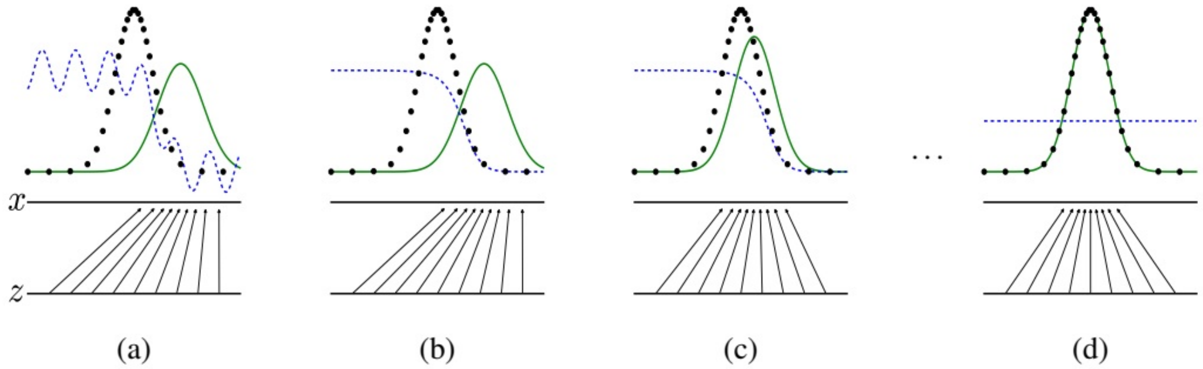


Figure 2.2: Generative adversarial networks are trained by continuously updating the discriminative distribution ($D$, blue, dashed line) so that it discriminates between samples from data (black, dotted line) $p_x$ and from generative distribution $p_g(G)$ (green, solid line). The lower horizontal line defines the domain from which $z$ is sampled. The horizontal line above is part of the domain $x$. (a) Let's assume an adversarial pair near convergence: $p_g$ is similar to $p_{data}$ and discriminator $D$ is a partially accurate classifier, initially. (b) In the inner loop of the algorithm $D$ is trained to discriminate samples from data. (c) After an update to Generator $G$, gradient of discriminator $D$ will reform $G(z)$ to flow to portions that are more likely to be classified as data. (d) After several steps of training, if $G$ and $D$ have enough capacity, they will reach a point at which both cannot improve.

## 2.2 CartoonGAN

### 2.2.1 Adversarial loss

Like equation 2.1, adversarial loss is applied to both generative $G$ and discriminative $D$ network. However, simply training the discriminator $D$ won't be enough to distinguish cartoon images. Because cartoon images have clear edges, therefore an output image without clearly reproduced edges is likely to confuse the discriminator trained with this loss.

So the author preprocessed the images of training data where $s_{data}(e) = \{e_i | i = 1..M\} \in E$ by removing clear edges in $S_{data}(c)$, where $C$ and $E$ are the cartoon images and cartoon-like images without clear edges, respectively. To make clear edges, first of all a canny edge detector is applied to detect edge pixel. Secondly, the regions are dilated and finally a Gaussian smoothing is applied in the dilated regions.

So the full adversarial loss can be defined as:

$$\begin{aligned}\mathscr{L}_{adv}(G,D) = \mathbb{E}_{c_i \sim s_{data}(c)}[logD(c_i)] + \mathbb{E}_{e_j \sim s_{data}(e)}[log(1-D(e_j)]+ \\ \mathbb{E}_{p_k \sim s_{data}(p)}[log(1-D(G(p_k)))].\end{aligned} \tag{2.4}$$

### 2.2.2 Content loss

Another loss is introduced in the paper known as content loss to preserve the semantic content of the input image. This can also be represented as CycleGAN's [3] *cycle consistency loss* which is shown on the later sections. In CartoonGAN [15], the authors adopt the high-level feature maps in VGG network [26] pre-trained by [27] which has shown good object preservation. So the content loss is:

$$\mathcal{L}_{con}(G,D) = \mathbb{E}_{p_i} \sim s_{data}(p)[||VGG_l(G(p_i)) - VGG_l(p_i)||_1] \tag{2.5}$$

where $l$ refers to the feature maps of a specific VGG layer. The authors definied their content loss using the $l_1$ spare regularization of VGG feature maps between the input image and the generated cartoon image.

## 2.3 CycleGAN

While many researchers have produced groundbreaking results such as [19] , [20], [21] on image-to-image translation using paired data, there hasn't been much successful research using unpaired data. To resolve this case, CycleGAN [3] has played an influencial role by presenting an approach which translates one image of domain $X$ to another domain $Y$ without any paired training data. This translation is based on an assumption that if an image, $x_i$ from domain $X$ can generate a new image a new image $y_i$ of another domain $Y$, eventually, the generated image, $y_i$ can be mapped to $X$ by generating a new image $\hat{x}$ where $x_i = \hat{x}$. To sum up, if $G$ is the generator which translates into domain $Y$ and $F$ is for the next translation, we can write it as following -

$$G(x_i) = y_i, where\, x_i \in X, y_i \in Y \tag{2.6}$$

$$F(y_i) = \hat{x}, where\, \hat{x} \in X \tag{2.7}$$

Let's break down the ideas that were used to make it a successful research and discuss them one by one.

### 2.3.1 Adversarial Loss

Previously, we have known about *Goodfellow et al.* [8] and how it has revolutionized the future of AI. As mentioned in section 4.1, we know that GAN [8] architecture works are based on *Adversarial Loss* which is just an extension of *Binary Cross-Entropy Loss*. However, in the case of CycleGAN [3], although *Adversarial Loss* has been used, *Binary Cross Entropy Loss* is not used. The reason is more related to the training inconsistency of GAN [8]. From *Mao et al.*, it is known that using *Least Squares Loss* shows more stability in training for CycleGAN [3] than using *Binary Cross-Entropy Loss*. So, equation of Adversarial loss turns into the following equation, where $c$ is an image from domain $C$ and $r$ from $R$:

$$For\,Generator\,G,\, \mathscr{L}_{G_{adv}} = \frac{1}{m}\sum_{i=1}^{m}(1-D_r(G(c)))^2$$

$$For\,Generator\,F,\, \mathscr{L}_{F_{adv}} = \frac{1}{m}\sum_{i=1}^{m}(1-D_c(F(r)))^2$$

$$\mathscr{L}_{adv} = \mathscr{L}_{G_{adv}} + \mathscr{L}_{F_{adv}} \tag{2.8}$$

However, using only *Adversarial Loss* is not enough to get the best result. This loss is *under-constraint* as it only limits the output to be of a specific domain and fails to limit a closely related output with respect to input. From fig 4.1, we can see its demonstration. The researchers of CycleGAN [3] uses an addition loss *Cycle Consistency Loss* to limit the output to be closely related to the input.

### 2.3.2 Cycle-Consistency Loss

The idea of *Cycle Consistency* goes way back. It is an idea of determining the transitivity of two images where second image is the reconstruction of the first image. This transitivity is here denoted as *Loss* in our case, where we use two *Cycle Consistency Loss* as *Forward & Backward Cycle Consistency Loss*.
For our *Cartoon-to-real* translation, if image, $c$ of domain $C$ is to be translated into domain $R$, through *Generator G*, there must be another *Generator F* to translate the newly translated image $G(c)$ into $\hat{c}$ with a view to reconstructing $c$. From figure 4.3, if this *Cycle Consistency Loss* is called *Forward - Cycle Consistency Loss*, the opposite is called *Backward - Cycle Consistency Loss*. It is defined using the following equations -

$$Forward\,Consistency\,Loss, \mathscr{L}_{f\_cyc} = \frac{1}{m}\sum_{i=1}^{m}(F(G(c))-c)$$

$$Backward\,Consistency\,Loss, \mathscr{L}_{b\_cyc} = \frac{1}{m}\sum_{i=1}^{m}(G(F(r))-r)$$

A question may arise on why using *Cycle Consistency Loss* solves the *under-constraint* issue. The intuition is that for a general mapping of two images, *Adversarial Loss* is great. However, it won't be able to specify the best image of the domain which should be mapped to the first image. On the other hand, *Cycle Consistency Loss* can do this job. Let's think of a scenario, where someone wants to translate *a garden image* to *Monet Painting* using GAN. To his surprise, he finds out that when used only *Adversarial Loss*, the machine translates the garden image with a random monet painting, kind of like figure 4.1 and on the other hand, when used *Cycle Consistency Loss*, it shows the same contents of the garden which seems to be painted by *Monet*. After some time, he finds out that, as *Cycle Consistency Loss* minimizes the reconstruction loss of the image, the machine is bound to choose an image which is pretty similar to the garden image, as its loss must be the least. So, we can say that, *Cycle Consistency Loss* binds the code to find out the best monet painting to be stylized.

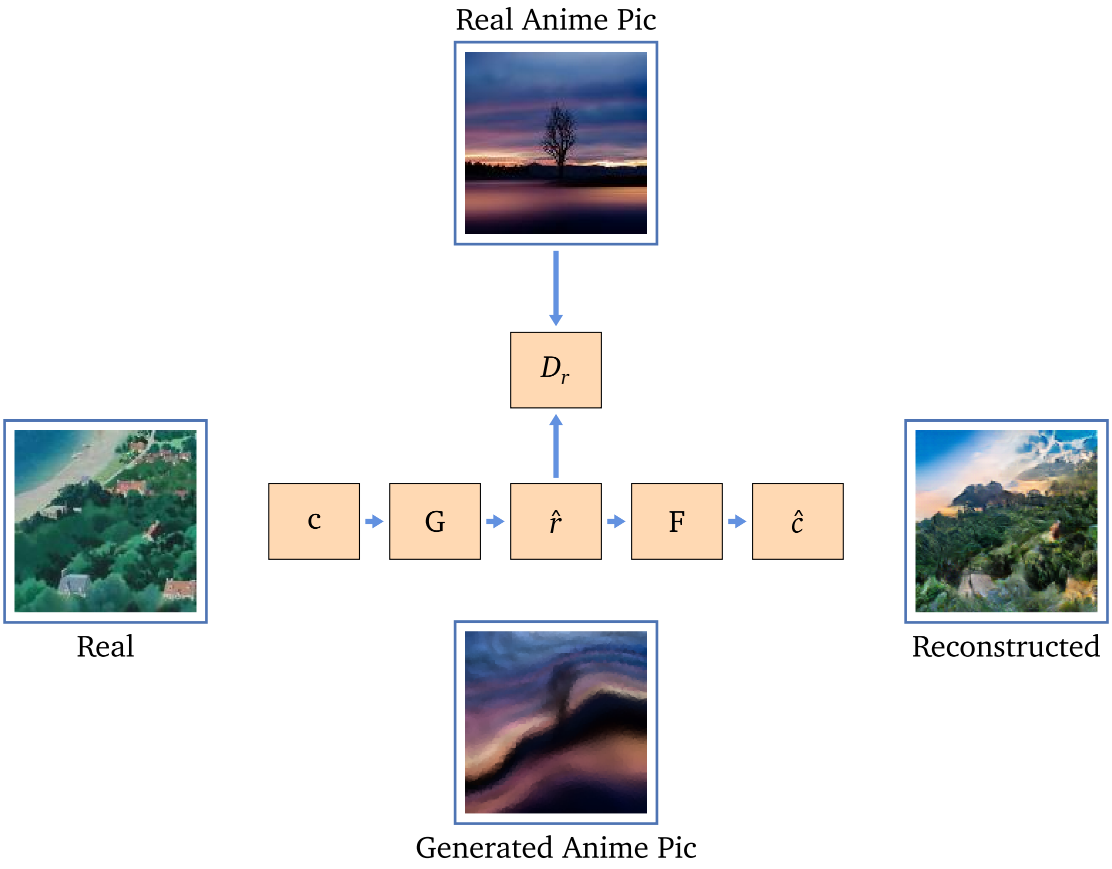


Figure 2.3: Architecture Of CycleGAN

$$\mathscr{L}_{cyc} = \mathscr{L}_{f\_cyc} + \mathscr{L}_{b\_cyc} \tag{2.9}$$

**Total Loss:**

So, the total loss is -

$$\mathscr{L}(G, F, D_c, D_r) = \mathscr{L}_{G_{adv}} + \mathscr{L}_{F_{adv}} + \lambda\mathscr{L}_{cyc}$$

where $\lambda$ controls the relative importance of the two objectives.

## 2.4 UNIT

Researchers of UNIT [4] worked on an assumption known as shared latent space assumption, where a pair of corresponding images in different domains can be mapped to a same latent representation in a shared-latent space. The framework combined a variational autoencoder(VAE) [28] and generative adversarial network (GAN). The adversarial training objective enforces the generator to transform corresponding images in two domains, while the variational autoencoders(VAE) [28] make correlations between translated images with input images in the respective domain.

### 2.4.1 VAE-GAN

Variational Autoencoders (VAEs) [28] is a generative model that encodes the input data. A variational autoencoder consists of an encoder, a decoder, and a loss function. The Encoder, $E$ is a neural network that takes in the distribution and outputs a hidden representation z that is usually of a much smaller dimension. The Decoder, $D$ is another neural network whose input is the hidden representation, $z$. The Decoder reconstructs the data using the input vector.

The encoder-generator pair $\{E_1, G_1\}$ constitutes a $VAE$ for the domain $X_1$ , termed $VAE_1$. For an input image $x1 \in X_1$, the $VAE_1$ first maps $x_1$ to a code in a latent space $Z$ via the encoder $E_1$ and then decodes a random-perturbed version of the code to reconstruct the input image via the generator $G_1$.

$$\mathcal{L}_{VAE_1}(E_1, G_1) = \lambda_1 KL(q_1(z_1|x_1)||p_\eta(z)) - \lambda_2 \mathbb{E}_{z_1 \sim q_1(z_1|x_1)}[log p_{G_1}(x_1|z_1)]. \tag{2.10}$$

$$\mathcal{L}_{VAE_2}(E_2, G_2) = \lambda_1 KL(q_2(z_2|x_2)||p_\eta(z)) - \lambda_2 \mathbb{E}_{z_2 \sim q_2(z_2|x_2)}[log p_{G_2}(x_2|z_2)]. \tag{2.11}$$

### 2.4.2 Cycle-Consistency

The shared-latent space assumption defines the cycle-consistency constraint in one way. The authors used this constraint in their framework to further regularize the mapping between unsupervised image-to-image translation.

A VAE-like objective function is used to model the cycle-consistency constraint, given in equation 2.12 and 2.13.

$$\mathcal{L}_{CC_1}(E_1, G_1, E_2, G_2) = \lambda_3 KL(q_1(z_1| x_1) || p_\eta(z)) + \lambda_3 KL(q_2(z_2|x_1^{1\to 2})||p_\eta(z)) - \lambda_4 \mathbb{E}_{z_2 \sim q_2(z_2| x_1^{1\to 2})}[log p_{G_1}(x_1| z_2)]. \tag{2.12}$$

$$\mathcal{L}_{CC_2}(E_2, G_2, E_1, G_1) = \lambda_3 KL(q_2(z_2| x_2) || p_\eta(z)) + \lambda_3 KL(q_1(z_1|x_2^{2\to 1})||p_\eta(z)) - \lambda_4 \mathbb{E}_{z_1 \sim q_1(z_1| x_2^{2\to 1})}[log p_{G_2}(x_2| z_1)]. \tag{2.13}$$

The negative log-likelihood term ensures that the translated image resembles the input and the KL terms penalize the hidden representation $z$ in the cycle consistency. The hyper-parameters $\lambda_3$ and $\lambda_4$ control the weights of the two different objective terms.

**Total Loss:** Finally, from the previous subsections, all the learning problems of the $VAE_1$, $VAE_2$, $GAN_1$ and $GAN_2$ are jointly solved in the equation 2.14.

$$\mathcal{L}(E_1, G_1, D_1, E_2, G_2, D_2) = \underset{E_1,E_2,G_1,G_2}{min} \underset{D_1,D_2}{max} \mathcal{L}_{VAE_1}(E_1, G_1) + \mathcal{L}_{GAN_1}(E_2, G_1, D_1) + \mathcal{L}_{CC_1}(E_1, G_1, E_2, G_2) + \mathcal{L}_{VAE_2}(E_2, G_2) + \mathcal{L}_{GAN_2}(E_1, G_2, D_2) + \mathcal{L}_{CC_2}(E_2, G_2, E_1, G_1). \tag{2.14}$$

## 2.5 Spectral Normalization

Spectral normalization [22] normalizes the spectral norm of the weight matrix W so that it satisfies the Lipschitz constraint, $\sigma(W) = 1$:

$$\bar{W}_{SN} := W/\sigma(W). \tag{2.15}$$

In the paper implementation, the computation using SGD for updating weight $W$ is cheaper than the calculation of the forward and backward propagations on neural networks. Algorithm 2 refers to the algorithm used for spectral normalization to update weight $W$.

**Algorithm 2:** SGD with spectral normalization

- Initialize $\tilde{\mathbf{u}}_l \in R^{d_l}$ for $l = 1, ..., L$ with a random vector (sampled from isotropic distribution).

- For each update and each layer l:

  1. Apply power iteration method to an unnormalized weight $W^l$:

  $$\tilde{\mathbf{v}}_l \leftarrow (W^l)^T \tilde{\mathbf{u}}_l / \left\|(W^l)^T \tilde{\mathbf{u}}_l\right\|_2 \tag{2.16}$$

  $$\tilde{\mathbf{u}}_l \leftarrow (W^l) \tilde{\mathbf{v}}_l / \left\|(W^l)^T \tilde{\mathbf{v}}_l\right\|_2 \tag{2.17}$$

  2. Calculate $\tilde{W}_{SN}$ with the spectral norm:

  $$\tilde{W}^l_{SN}(W^l) = W^l / \sigma(W^l), where\, \sigma(W^l) = \tilde{\mathbf{u}}_l^T W^l \tilde{\mathbf{v}}_l \tag{2.18}$$

  3. Update $W^l$ with SGD on mini-batch dataset $D_M$ with a learning rate $\alpha$:

  $$W^l \leftarrow W^l \leftarrow \alpha \nabla_{W^l} l(\tilde{W}^l_{SN}(W^l), D_M) \tag{2.19}$$

# Chapter 3

# Methodology

In this chapter, we discuss the different approaches we have accumulated for our model *toon2real*.

## 3.1 Unpaired Image-to-Image Translation

As we had to use unpaired dataset, we used our idea on various unpaired image-to-image translation technique which are discussed in the following subsections.

### 3.1.1 CycleGAN

CycleGAN is the state of the art algorithm which translates between domains without paired examples. In our thesis, we exploit our work at the level of sets: we are given cartoon domain $X$ and real world domain $Y$. We train a mapping $G : X \rightarrow Y$ so that the output, $\hat{y} = G(x)$. *Adversarial training* is used to classify $\hat{y}$ apart from $y$. In theory, adversarial training's objective is to get an output distribution over $\hat{y}$ such that the empirical distribution $p_{data}(y)$ is matched. In order to do that, optimal generator $G$ translates the domain $X$ to domain $\hat{Y}$ such that it matches approximately to domain $Y$. However, adversarial training is not enough to persuade $G$ to translate, as there are infinite mappings to domain $Y$ that will result in same distribution. However if we do modify the adversarial training, it leads to the common problem of GAN *mode collapse*, where all input images map to same output image [8].

**Cycle Consistency loss:** To limit the infinite mappings of domain $Y$, *cycle consistency loss* is used. The intuition behind this property is that if we translate, e.g. a sentence from English to French, and then translate it back from French to English, we should arrive back at the original sentence [29]. So, an image generated from an input can be reconstructed back to the input again such that $x = F(G(x))$, where $F$ and $G$ are generators and $x$ is the input, and thus it is able to map an image of target domain which is as close as possible to the image of input domain. Combining this loss with adversarial loss yields our full objective.

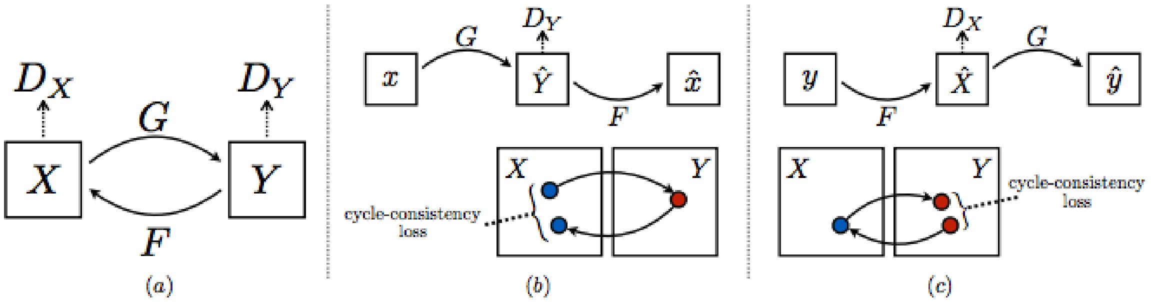


Figure 3.1: (a) CycleGAN [3] model contains two mapping functions $G : X \rightarrow Y$ and $F : Y \rightarrow X$, and discriminators $D_Y$ and $D_X$. $D_Y$ persuades $G$ to translate $X$ into outputs similar in distribution in domain $Y$, and vice versa for $D_X$ and $F$. Two *cycle consistency loss* is used to maintain the distribution of transforming $X$ to $Y$ and back to $X$ so that no content loss occurs: (b) forward cycle-consistency loss: $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$, and (c) backward cycle-consistency loss: $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$.

**Reducing Model Oscillation:** To prevent the model from changing drastically from iteration to iteration, Shrivastava et al. [30] proposed a technique to feed the discriminators with a history of generated images, rather than just the ones produced by the latest generative networks. To do this, a pool is kept to store the 50 most recently generated images.

**Network Architecture:** The architecture of generative networks is adopted from Johnson et al. [16] which have shown admirable results for style transfer and super resolution task. The network consists of two stride-2 convolutions, several residual blocks [] and two fractionally strided convolutions with stride $\frac{1}{2}$. As our image resolution is 128 × 128 so we used 6 blocks of residual net. Similar to Johnson et al. [16] instance normalization [31] is used.

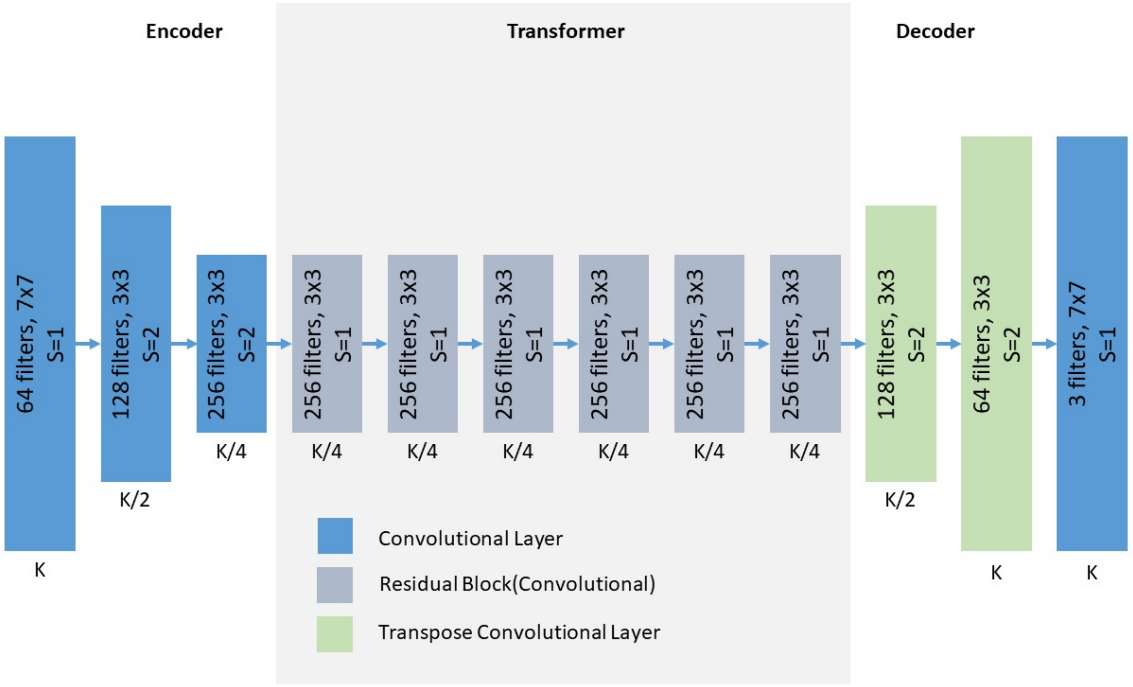


Figure 3.2: Generative Network of CycleGAN [3].

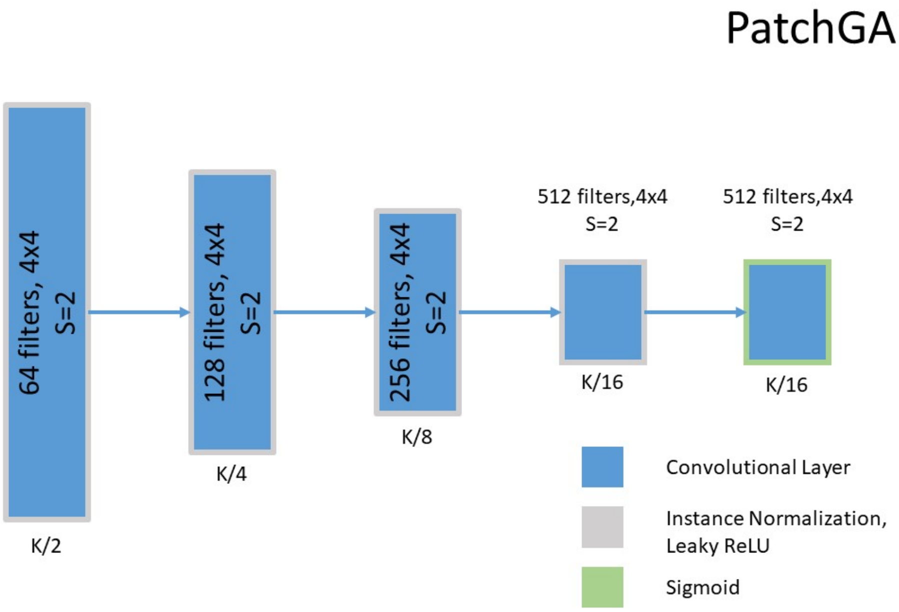


Figure 3.3: Discriminative Network of CycleGAN [3].

For the discriminator networks, shown in Fig 3.3, 70 × 70 PatchGANs [11], [32], [33] is used, which aimed to classify if the 70 × 70 overlapping image patches are realistic or fake. Patch-level discriminator has fewer parameters than a full-image discriminator, and can be applied to arbitrarily-sized images.

### 3.1.2 UNIT

We compare our work with UNIT [4] framework. Liu et al. [4] proposed this framework based on shared latent assumption, where a pair of corresponding images in different domains can be mapped to a same latent representation in a shared-latent space. The framework combined a variational autoencoder(VAE) and generative adversarial network (GAN). The adversarial training objective enforces the generator to transform corresponding images in two domains, while the variational autoencoders(VAE) make correlations between translated images with input images in the respective domain.
Larsen et al. [34] proposes to combine Variational AutoEncoder (VAE) [28] with GAN [8] to exploit both of their benefits, as GAN can generate sharp images but often miss some modes while images produced by VAE [28] are blurry but have large variety. This method is used in UNIT framework [4]. The VAE part regularize the encoder $E$ by imposing a prior of normal distribution (e.g. $z \sim N(0,1)$). Also, VAE-GAN [34] proposes to represent the reconstruction loss of VAE in terms of the discriminator $D$.

A higher level representation of UNIT framework is shown in Fig 3.4. In fig 3.4 $E_1, E_2$ defines encoder, $G_1, G_2$ defines generator and $D_1, D_2$ defines the discriminative function.

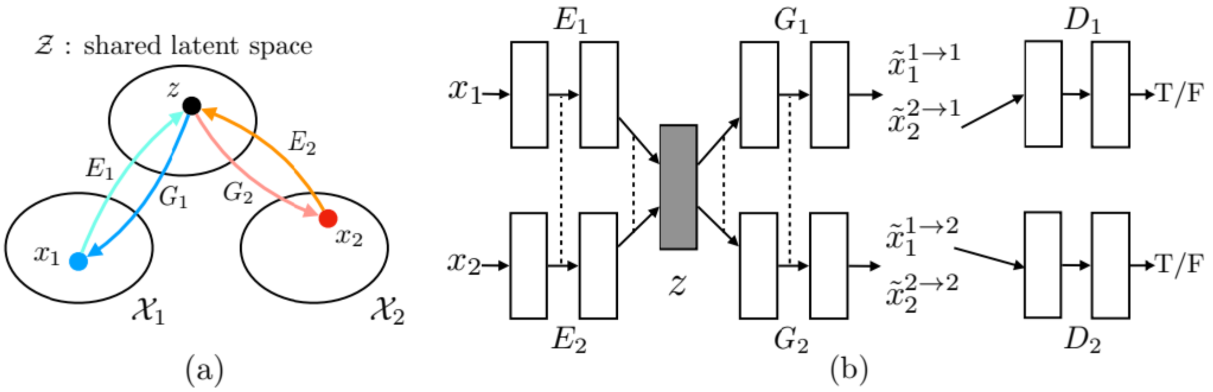


Figure 3.4: (a) The authors [4] dictate a shared latent space assumption where a pair of corresponding images $(x_1; x_2)$ in two different domains $X_1$ and $X_2$ can be mapped to a same latent code $z$ in a shared-latent space $Z$. Two encoders $E_1$ and $E_2$ are used to map images to latent codes. Also, two generative functions $G_1$ and $G_2$ maps latent codes to images. (b) The author's UNIT framework. $E_1 E_2 G_1$ and $G_2$ are represented using CNNs and implement the shared-latent space assumption using a weight sharing constraint where the connection weights of the last few layers in $E_1$ and $E_2$ are tied (illustrated using dashed lines) and the connection weights of the first few layers in $G_1$ and $G_2$ are tied. Here, $\tilde{x}_1^{1\to1}$ and $\tilde{x}_2^{2\to2}$ are self-reconstructed images and $\tilde{x}_1^{1\to2}$ and $\tilde{x}_2^{2\to1}$ are transformation of one domain to another. Two discriminative networks $D_1$ and $D_2$ are used for the respective domains for evaluating whether the translated images are realistic or not.

**Network Architecture** The network architecture used for the unsupervised image-to-image translation experiments is given in Fig 3.5 and 3.6. 8 residual blocks is used in UNIT [4] framework. Fig 3.5 shows the architecture of generator and 3.6 shows the architecture

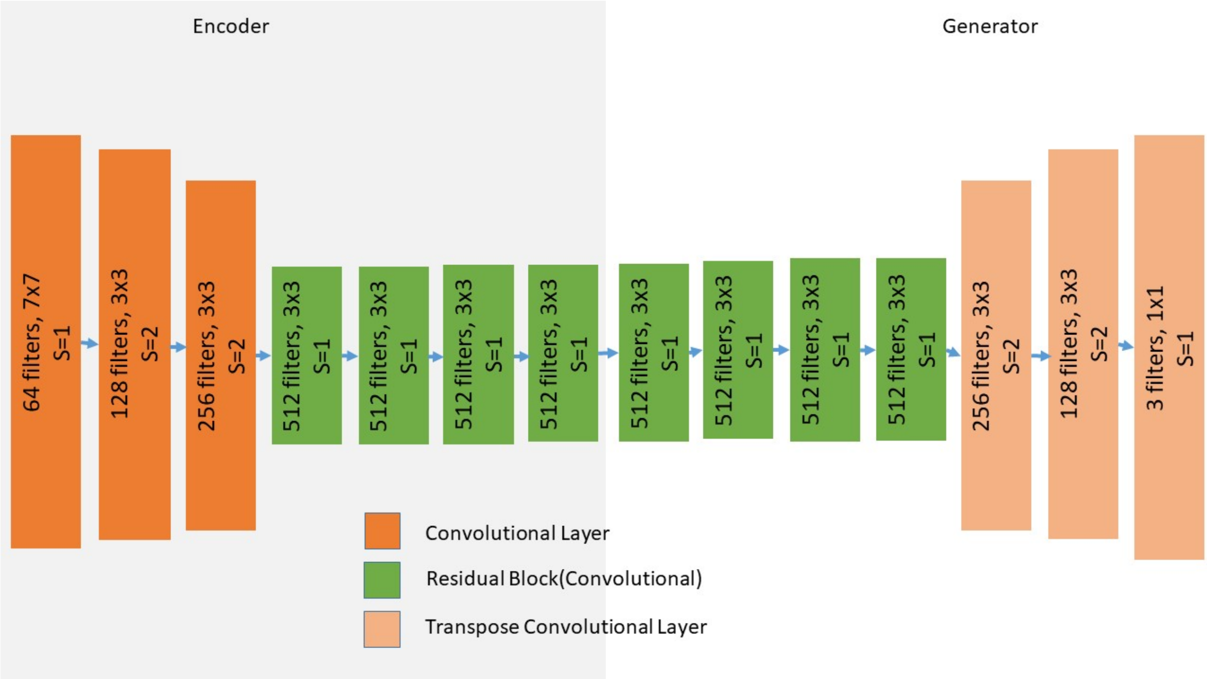


Figure 3.5: Generative Network of UNIT [4] framework.

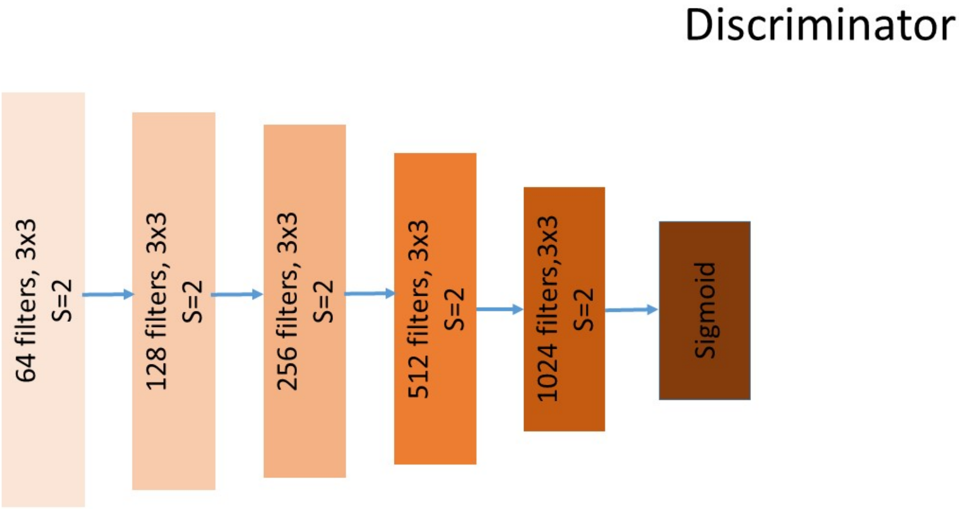


Figure 3.6: Discriminative Network of UNIT [4] framework.

of discriminator.

## 3.2 Using Spectral Normalization

One of the challenges in the training of generative adversarial networks [8] is the lack of stability. Miyato et al. [22] proposed a novel weight normalization technique called *spectral normalization* to stabilize the training of discriminator $D$. Spectral normalization normalizes the spectral norm of the weight matrix $W$ such that it satisfies the Lipschitz constraint $\sigma(W) = 1$. The technique is computationally inexpensive and the implementation is simple. The technique only requires Lipschitz constraint, one of the hyper-parameters of network to be tuned. The technique improves the quality of generated images better than weight normalization [35] and gradient penalty [36] used previously.

# Chapter 4

# Results and Comparison

## 4.1 Evaluation

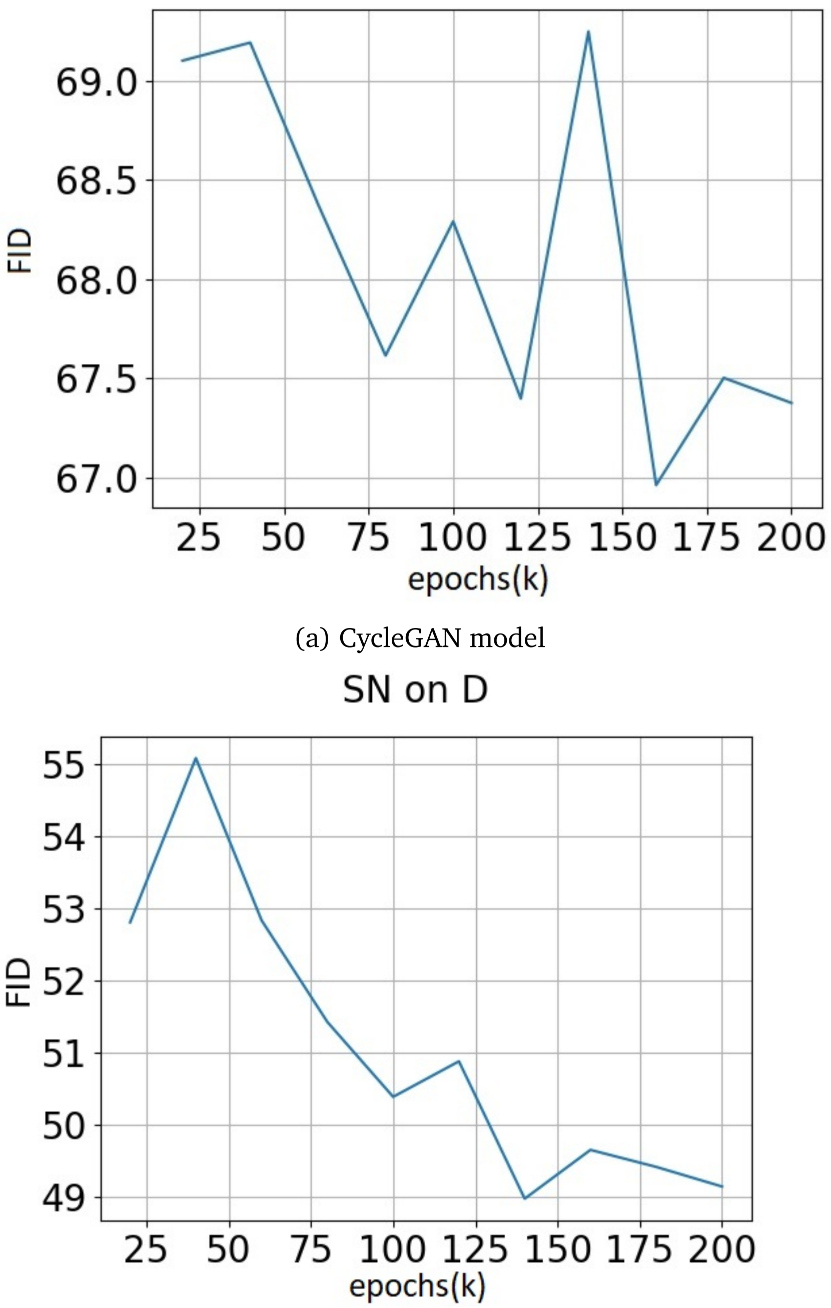
In this section, we discuss about the evaluation methods that we used for our research. We used *three* different evaluation methods which are *Fréchet Inception Distance*, *Stabilization Evaluation* and *Human Evaluation*. In the following *three* subsections, we will discuss about this.

### 4.1.1 Evaluation metric

We chose the Fréchet Inception Distance (FID) [37] for quantitative evaluation. As FID score measures the difference between the generated dataset and the target dataset, it has shown more consistency with human evaluation. it calculates the Wasserstein-2 distance between the translated image and the real world images from an intermediate layer of an Inception-v3 network. Lower the FID score, the closer the distance between translated image and real domain images. As our task is image-to-image translation where we want our output to have the content of input cartoon images and the style of real-world images, we calculated a weighted average between them, where we used 80% weight for target data and 20% weight for input data. From Table 4.1 we can see that our work has shown the least FID score compared to other state of the art models.

Table 4.1: Fid scores of ours & UNIT model.

| Models | Our Work | UNIT |
|---|---|---|
| FID | **48.4225** | 55.9214 |



(a) CycleGAN model

(b) Ours

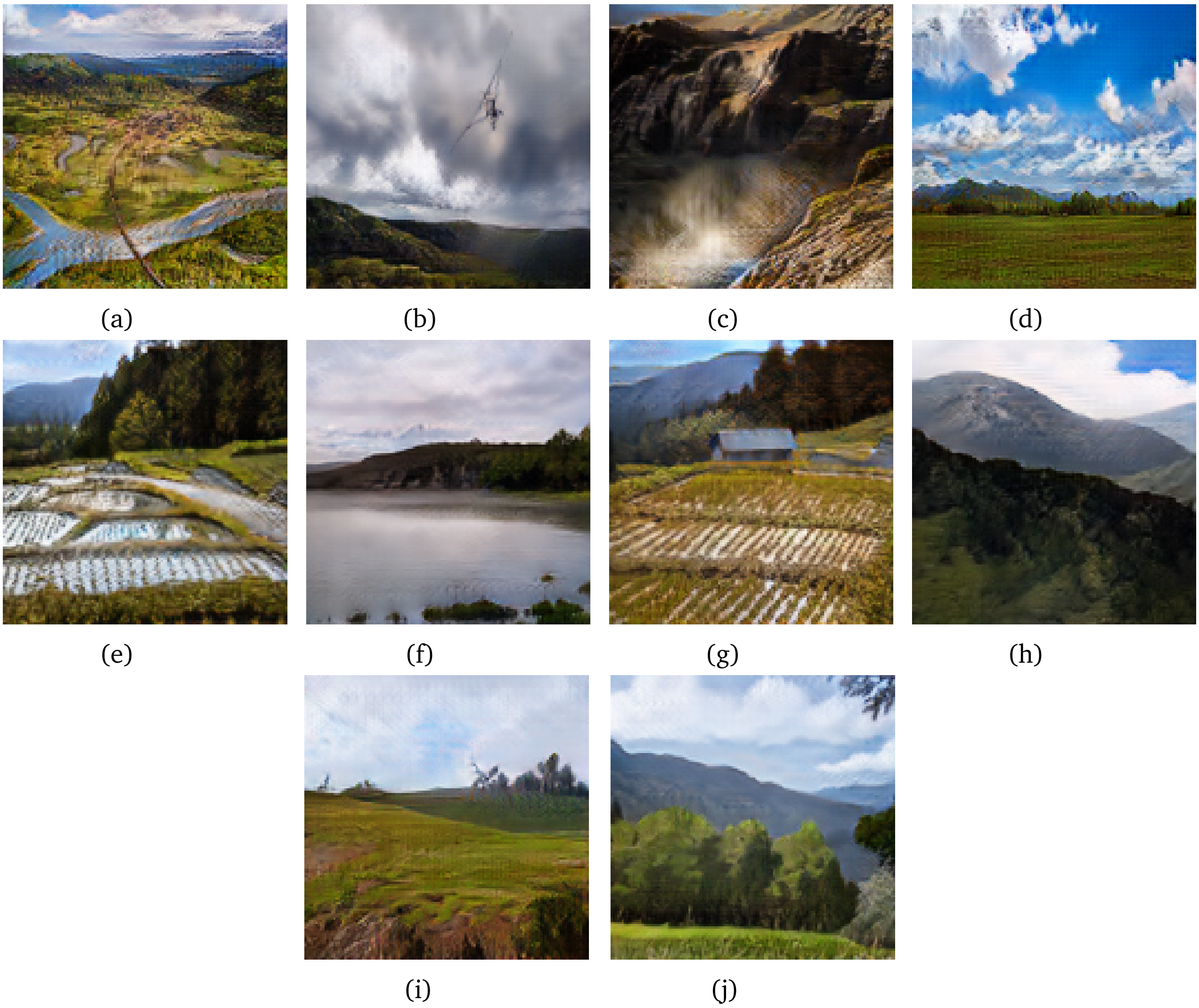
Figure 4.1: Here, FID scores for CycleGAN (a) and for our method (b) are shown from 20 epochs up to 200 epochs.

### 4.1.2 Evaluation of stabilization technique

By utilizing spectral normalization technique on discriminator network shown in Figure 4.1b, we started to gain lower FID score from the very initial of training compared to baseline model, which is implemented based on CycleGAN [3] model. Spectral normalization is used on discriminator network on baseline model which is shown in 4.1b. From 4.1b, the quality of transforming images doesn't improve monotonically during training. For example, the FID score of our work starts to drop at the 37th epoch. On the contrary, baseline model's FID score starts to rise after 125th epoch and it crosses the initial FID scores, whereas in our work, the scores didn't rise like the baseline model did. From this, we can clarify that we achieved a more stabilized model and better scores. We can also clarify from Figure 4.1 that the stabilization technique also takes fewer training epochs to achieve better scores.

### 4.1.3 Human evaluation

We randomly selected 10 images from our cartoon-to-real world domain transformation and evaluated them by creating a survey on the social media platform. More than 100 peoples gave a scoring on each of the individual images, on the number of 5.0. The images are shown in Fig 4.2 and the score for each image are shown in Table 4.2.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j)

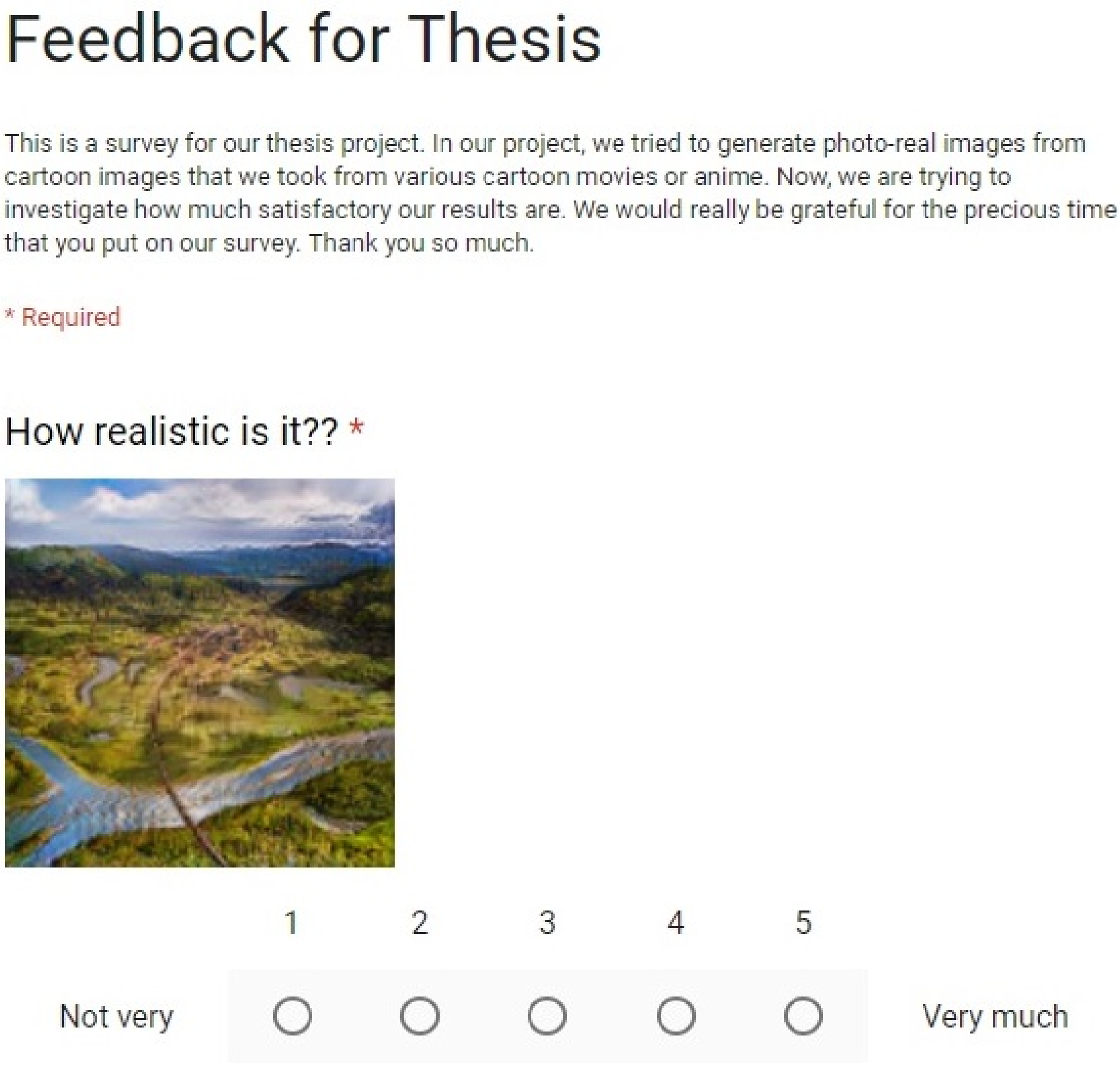
Figure 4.2: 10 images taken for human evaluation. All the samples are the result of our cartoon-to-real world domain transformation.

Table 4.2: Average score of each individual images of Fig 4.2. The score is rated on 5.0.

| Image No | Score | Image No | Score |
|---|---|---|---|
| 4.2a | 3.028169014 | 4.2f | 4.323943662 |
| 4.2b | 3.436619718 | 4.2g | 4.197183099 |
| 4.2c | 2.971830986 | 4.2h | 4.070422535 |
| 4.2d | 3.802816901 | 4.2i | 4.098591549 |
| 4.2e | 3.957746479 | 4.2j | 3.746478873 |

From the table we can conclude that, 90% of our images have more than 3.0 rating. Our model have thus produced decent results for cartoon-to-real translation task.

The screenshot of our google form is shown in Fig 4.3.



Feedback for Thesis

This is a survey for our thesis project. In our project, we tried to generate photo-real images from cartoon images that we took from various cartoon movies or anime. Now, we are trying to investigate how much satisfactory our results are. We would really be grateful for the precious time that you put on our survey. Thank you so much.

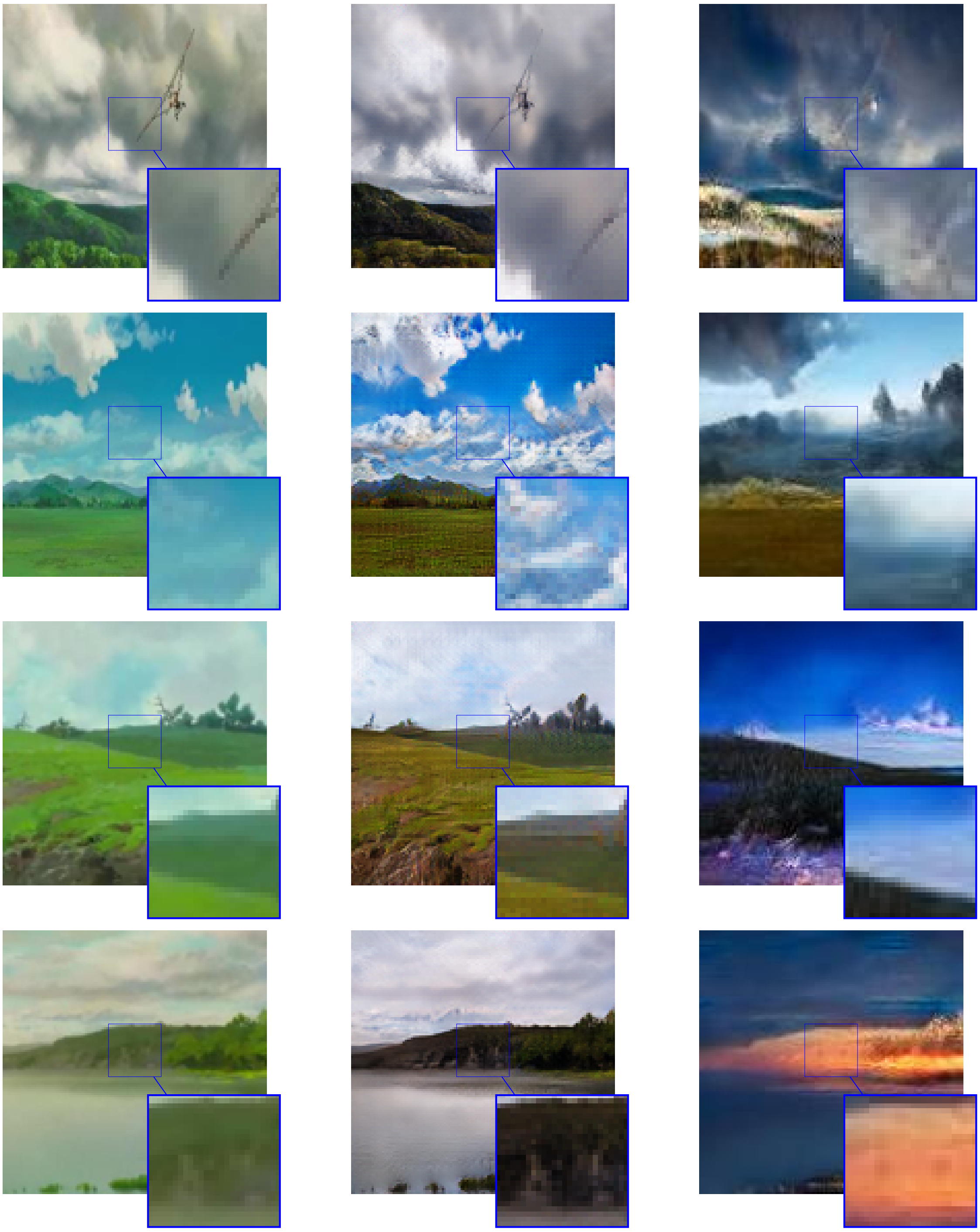
* Required

How realistic is it?? *

| | | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | |
| Not very | ○ | ○ | ○ | ○ | ○ | Very much |

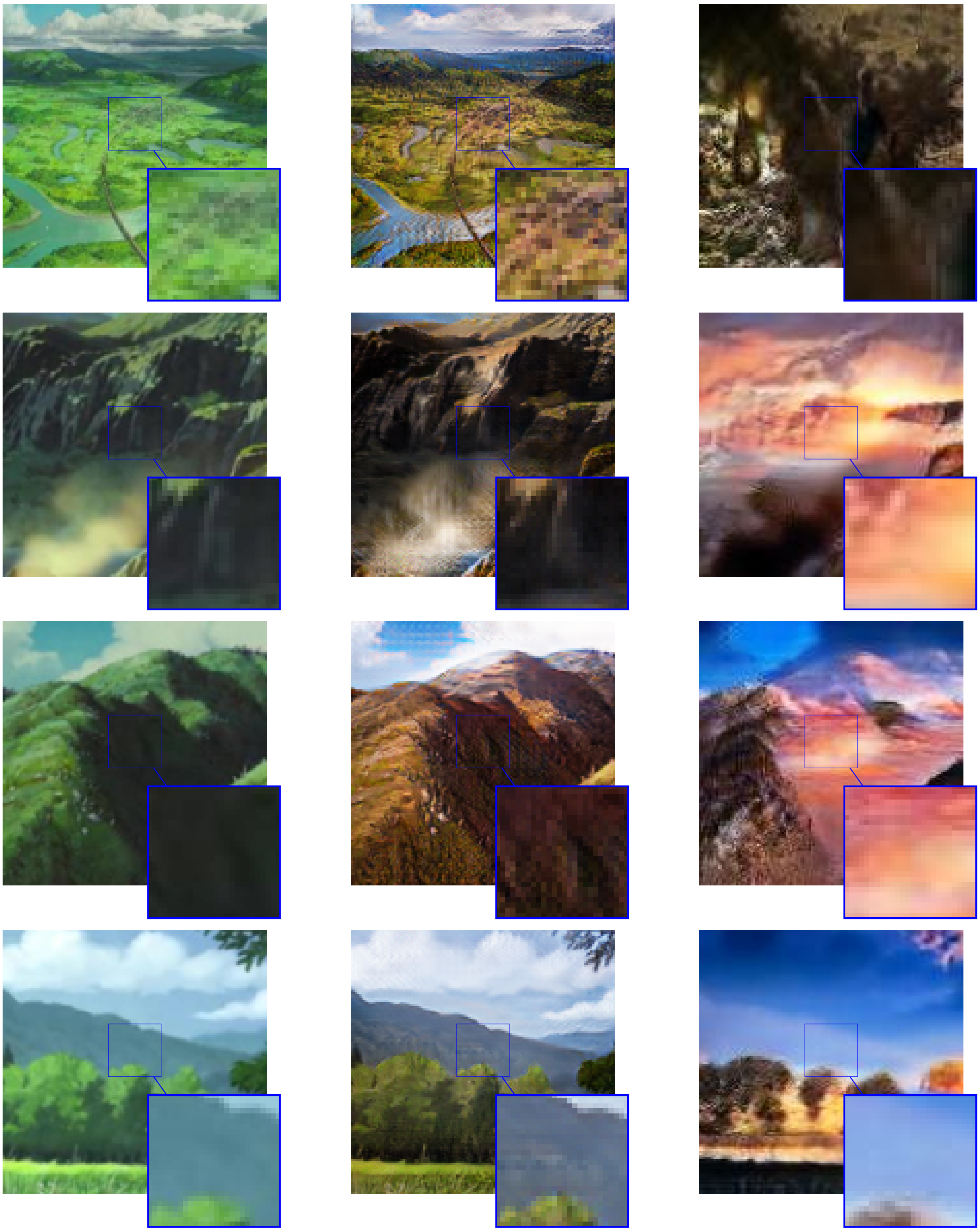
Figure 4.3: Screenshot of our Google form on the evaluation of our model.

## 4.2 Results of our work



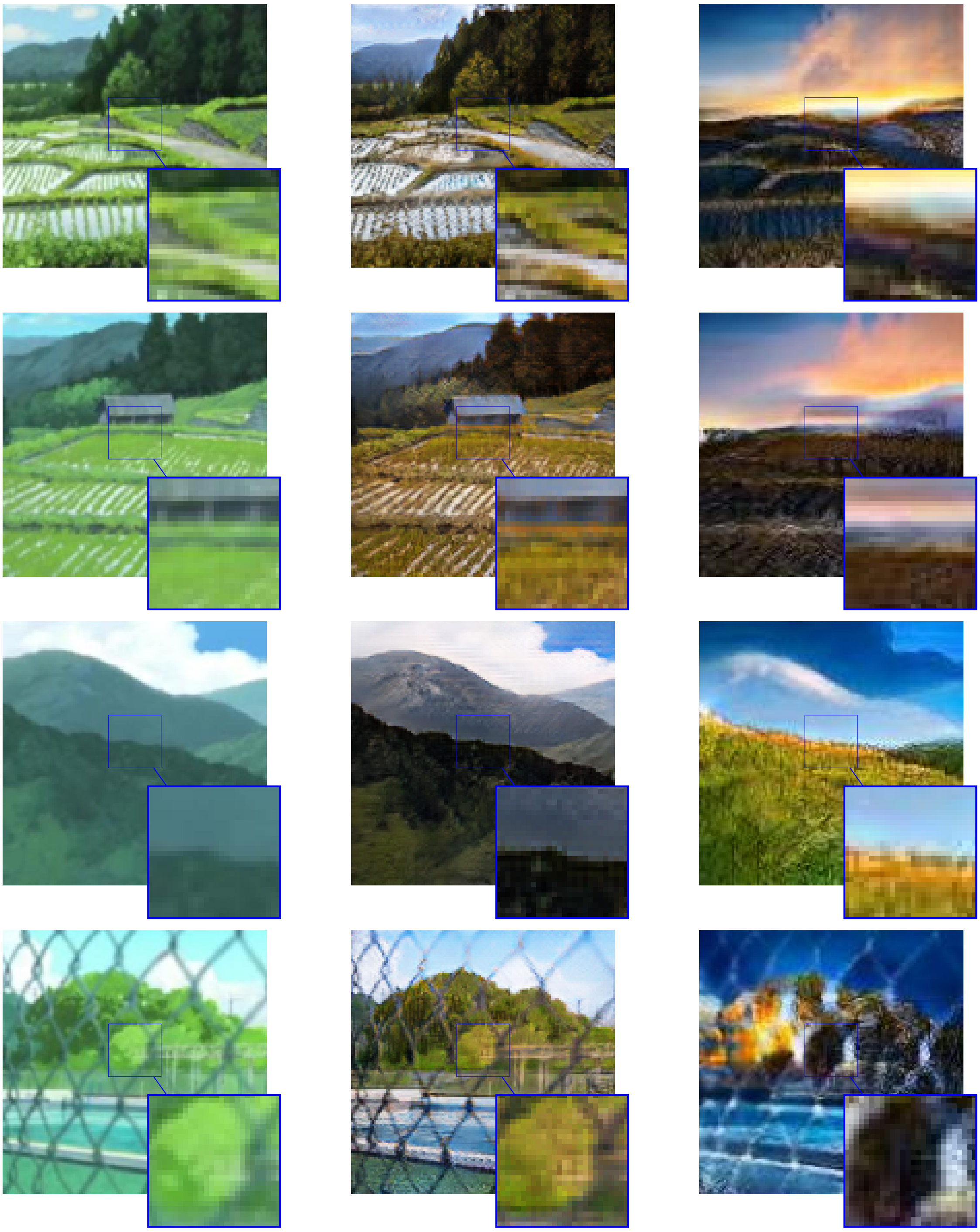
(a) input (b) toon2real (c) UNIT

Figure 4.4: Detailed comparisons in terms of contrast and content preservation. (a) Input images of cartoon scenes (a portion is amplified inside red bounding box for better observation). (b) *Result of our toon2real*: shows more contrast on content, compared to other works. (c) *Result of UNIT* [4] which shows the lacking of content than (b).



(a) input (b) toon2real (c) UNIT

Figure 4.5: Collective set $I$: more collective samples of *toon2real* in comparison with UNIT.



(a) input (b) toon2real (c) UNIT

Figure 4.6: Collective set $II$: more collective samples of *toon2real* in comparison with UNIT.

# Chapter 5

# Limitations and Future Work

## 5.1 Limitations

Despite achieving a better FID score than other techniques, this method still lacks in achieving a perfect image to image translation. On the following sub sections we will discuss more about the limitations.

### 5.1.1 Presence of Realism

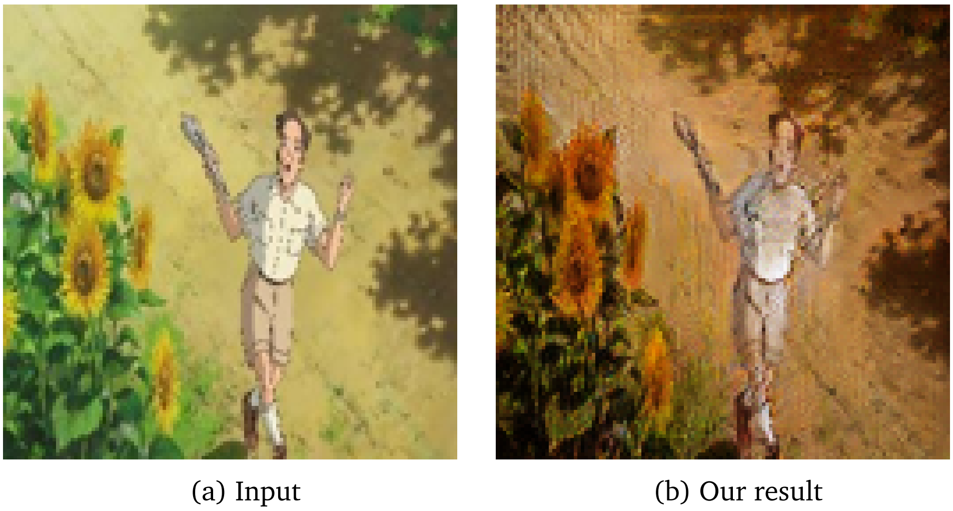
This technique fails to make a cartoon image realistic when there is nothing at all realistic in the cartoon image. We know that any watery things such as sea doesn't seem realistic in cartoon images. In that case, the model will fail to generate any realistic image. This happens because of geometric structure of the sea in realistic images. Cartoon images fail to catch that.

### 5.1.2 Human Structure

One of the biggest drawback of this model is that it fails to translate cartoon human figure into realistic images. From Figure 5.1 we can see the example of where the the model failed to generate a realistic version of the human in input cartoon image.

### 5.1.3 Dataset

Another drawback is that to create a large enough dataset, we curated images from various sources. However, the GAN technique loose some of its sharpness because of learning so many different distributions of the image.



(a) Input

(b) Our result

Figure 5.1: An unsuccessful case (b) of our method where it fails to translate the human figure for the input image (a).

## 5.2 Future works

Our future plan is to lessen our current limitations by investigating more geometry and content aware model to improve the texture so that the gap with the photorealistic domain decreases. In addition to FID score and human involved evaluation, we have plans to arrange perceptual evaluation processes to asses the correctness of your outcomes. These are discussed in details in the following subsections.

### 5.2.1 Segmentationally Aware

To improve the score, we have plans to train the datasets semantically, which is instead of training the entire images, we will train their segmented versions. Hence, the objects, e.g. trees, from cartoon domain will be segmented and mapped to the similar objects (trees) in the real domain like it done in [14,38].

### 5.2.2 Variational Discriminator Bottleneck

Consider the equation given in 2.1. Peng et al. [39] proposed a *bottleneck loss* added to 2.1 by which necessary amount of information is passed through discriminator network which then allows the generator to improve on the most discerning differences between real and fake samples. This achieved significant advantage in training and in generating images. We believe that by incorporating VDB in our work we can improve our results and training progress too.

### 5.2.3 Dataset

As it is mentioned in 5.1.3, we will try to minimize the the variation in our dataset as much as possible which will limit the learning distribution of GANs.

# References

[1] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, vol. abs/1511.06434, 2015.

[2] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," *arXiv*, Dec 2018.

[3] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networkss," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[4] M. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," *CoRR*, vol. abs/1703.00848, 2017.

[5] Aramide-Tinubu, "'The Lion King': Is the Live-Action Film Disney's Most Expensive Movie?," *Cheat Sheet*, Nov 2018.

[6] T. Bacon, "Lion King's CGI Has Changed In New Trailer: Here's How It's Different," *ScreenRant*, Feb 2019.

[7] "Animation On A Budget - Pixelbox Visual Design," Jun 2015. [Online; accessed 15. Mar. 2019].

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[9] X. Yu, X. Cai, Z. Ying, T. Li, and G. Li, "Singlegan: Image-to-image translation by a single-generator network using multiple generative adversarial learning," in *Asian Conference on Computer Vision*, 2018.

[10] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Advances in Neural Information Processing Systems*, pp. 700–708, 2017.

[11] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

[12] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *arXiv preprint arXiv:1812.04948*, 2018.

[13] J. Li, "Twin-gan–unpaired cross-domain image translation with weight-sharing gans," *arXiv preprint arXiv:1809.00946*, 2018.

[14] M. Tomei, M. Cornia, L. Baraldi, and R. Cucchiara, "Art2real: Unfolding the reality of artworks via semantically-aware image-to-image translation," *arXiv preprint arXiv:1811.10666*, 2018.

[15] Y. Chen, Y.-K. Lai, and Y.-J. Liu, "Cartoongan: Generative adversarial networks for photo cartoonization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9465–9474, 2018.

[16] J. Johnson, A. Alahi, and F. Li, "Perceptual losses for real-time style transfer and super-resolution," *CoRR*, vol. abs/1603.08155, 2016.

[17] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. N. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," *CoRR*, vol. abs/1612.03242, 2016.

[18] S. E. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," *CoRR*, vol. abs/1605.05396, 2016.

[19] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 8798–8807, 2018.

[20] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays, "Scribbler: Controlling deep image synthesis with sketch and color," *CoRR*, vol. abs/1612.00835, 2016.

[21] L. Karacan, Z. Akata, A. Erdem, and E. Erdem, "Learning to generate images of outdoor scenes from attributes and semantic layouts," *CoRR*, vol. abs/1612.00215, 2016.

[22] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *CoRR*, vol. abs/1802.05957, 2018.

[23] E. L. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep generative image models using a laplacian pyramid of adversarial networks," *CoRR*, vol. abs/1506.05751, 2015.

[24] J. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," *CoRR*, vol. abs/1609.03552, 2016.

[25] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," *CoRR*, vol. abs/1604.07379, 2016.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," *CoRR*, vol. abs/1409.0575, 2014.

[28] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[29] R. W. Brislin, "Back-translation for cross-cultural research," *Journal of cross-cultural psychology*, vol. 1, no. 3, pp. 185–216, 1970.

[30] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," *CoRR*, vol. abs/1612.07828, 2016.

[31] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *CoRR*, vol. abs/1607.08022, 2016.

[32] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," *CoRR*, vol. abs/1604.04382, 2016.

[33] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," *CoRR*, vol. abs/1609.04802, 2016.

[34] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," *arXiv preprint arXiv:1512.09300*, 2015.

[35] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," *CoRR*, vol. abs/1602.07868, 2016.

[36] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *CoRR*, vol. abs/1704.00028, 2017.

[37] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, G. Klambauer, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a nash equilibrium," *CoRR*, vol. abs/1706.08500, 2017.

[38] S. Mo, M. Cho, and J. Shin, "Instagan: Instance-aware image-to-image translation," *arXiv preprint arXiv:1812.10889*, 2018.

[39] X. B. Peng, A. Kanazawa, S. Toyer, P. Abbeel, and S. Levine, “Variational discriminator bottleneck: Improving imitation learning, inverse rl, and gans by constraining information flow,” *CoRR*, vol. abs/1810.00821, 2018.