Landmark-Assisted Star-GAN: Image to Bitmoji Avatar Generation

Sean Tonthat
Department of Electrical and Computer Engineering
UC San Diego
stonthat@ucsd.edu

Zeting Luan
Department of Electrical and Computer Engineering
UC San Diego
zluan@ucsd.edu

Abstract

Domain adaptation in machine learning is the task of learning features of one domain distribution and applying it to another domain distribution. Much of attention is on adapting similar domains through various GAN architectures, focusing on changing physical attributes such as gender, hair color, etc. In the case of adapting very different domains such as real life images of people and their corresponding cartoon representations, the purpose is to create as accurate a representation in the target domain. For each of these two different types of domain adaption, only one adaption is being performed. The ability to perform domain adaption to two or more domains is rarely explored as in adapting from domain A to domain B and then to domain C, which results in larger mean discrepancy between domains. To explore this problem, we propose combining the two popular architectures of StarGAN and CycleGAN in order to perform multiple domain adaptation. This model seeks to modify the physical attributes of a person from a real image and then output a Bitmoji avatar of this altered image, performing two domain adaptations.

https://github.com/seton-develops/ECE228_FINAL_PROJECT_GROUP24

1 Introduction

Ever since Goodfellow proposed the generative adversarial networks(GAN) [1], there has been a huge interest in GAN regarding synthetic images generation. But in recent years, machine learning communities have grown interest in domain adaptation using GANs. Several methods focus on data augmentation to improve the adaptation from source domain to target domain. Huang and Lin *et al* proposed AugGAN[2] to deal with the issue that many GAN based models perform poorly on preserving image-objects and maintaining translation consistency. Choi *et al* presented a data augmentation method based on GAN utilizing self-ensembling[3]. Zhu and Park *et al* proposed Cycle-Consistent Adversarial Networks (CycleGAN)[4] using cycle consistency loss to make sure the information consistent between source domain and target domain. These methods successfully translated information from the source domain to the target domain. However, in comparison, our goal is to maximize the distance distribution between source domain and target domain. Therefore, we proposed a GAN-based model that can translate two domains with a larger discrepancy.

The goal of this project is to create a combined architecture of two GAN frameworks: StarGAN and CycleGAN. Each model by itself is capable of a single domain adaptation. Of particular interest is unsupervised translating between real images of faces to cartoon avatars. We adopt the approach for the human to Bitmoji domain from Wu et. al.[5] by using a Landmark Consistency Loss and Local Discriminator Loss. Our Landmark-Assisted Star-GAN seeks to combine the landmark domain

adaption of StarGAN with the real to image domain adaption of a landmark-assisted CycleGAN by combining both in an end-to-end manner.

Our contributions to this project included:

- Developing a method to filter our non-forward facing images from training set using Perspective-n-point
- Implementing a version of CycleGAN from scratch
- Designing a TCDCN based architecture to detect the coordinates of facial landmarks
- Designing a CNN-based local discriminator to predict whether the landmarks generated are representative of the human and cartoon data sets
- Developing a live-demonstration that can take a webcam image and convert it to a Bitmoji avatar

2 Related Works

2.1 StarGAN

StarGAN[6] was proposed by Choi *et al*, which is a unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. StarGAN addressed the difficulty that most models have when handling two or more domains. They introduced simultaneous training of multiple data sets and learned the mappings between all domains using only a single generator.

StarGAN has three distinct types of losses to optimize its single generator and discriminator. The first is the adversarial loss proposed by the GANs[1] model so that the discriminator learns how to distinguish the real and fake images. For the second, Choi *et al* proposed a domain classification loss to train the generator and the discriminator separately in order to correct classify the source domain and the target domain. The third is a reconstruction loss intended to minimize the adversarial and classification losses. Our project seeks to retain the architecture of StarGAN as the basis for the first half of our Landmark Assisted StarGAN combined architecture.

2.2 CycleGAN

CycleGAN[4] was introduced by Zhu *et al* in 2017. They proposed a novel method for unpaired image-to-image translation, which vanilla GANs struggle with, through the new model CycleGAN. The CycleGAN model are capable to learn the distributions from two domains. They achieved excellent results in domain adaptation using CycleGAN such as Zebra $\rightarrow$ Horse, Apple $\rightarrow$ Orange tasks.

CycleGAN used two sets of generators and discriminators so that the images generated from domain $X \rightarrow Y$can be reconstructed back to domain X. The cycle consistency loss was introduced as the L-1 norm between the original image and the reconstructed image. By minimizing the cycle consistency loss, it creates two mapping that matches the features between two domains. However, CycleGAN was found to be unable to translate two domains that have a large mean discrepancy [4].

2.3 Landmark Assisted CycleGAN

The Landmark Assisted CycleGAN [5] was proposed by Wu and Gu *et al* in 2019. They adopted facial landmark to assist CycleGAN[4] to achieve translation between real-face images and cartoon images. They proposed a landmark consistency loss to guide the training of local discriminators for facial features such as eyes, nose, and mouth for their CycleGAN-based model. This minimizes deformations of these landmarks in the final Bitmoji output.

Wu and Gu *et al* [5] pre-trained a U-Net like landmark regressor to generate the facial landmark coordinates from both the human-face domain and the cartoon domain. By minimizing the landmark consistency loss, the model kept the landmark in both domains at the same position, which makes

sure the facial features from both domains located in the same area. Their approach were capable to generate high-quality cartoon faces while keeping the majority of the facial attributes from the human faces. In our work, we also plan on employing a landmark consistency loss; however, our attempts to replicate the U-Net proposed by Wu and Gu *et al*[5] has thus far been unsuccessful as noted in the Landmark Regressor section of this report.

3 Methods

3.1 Problem Formulation and Motivation

Our goal was to train a model that was capable of doing two step domain adaptation by translating the human image outputted from the pre-trained StarGAN model to a Bitmoji representation. Since CycleGAN failed to maintain the facial features such as the color, position, and shape of the eyes, nose, and mouth during the translation, we incorporated the landmark consistency loss and local discriminator from [5] to assist the two-step domain adaptation process. Our task was translating images from human domain X to cartoon domain Y.

Let $\{x_i\}_{i=1}^N$ and $\{y_i\}_{i=1}^N$ be the training samples from X and Y respectively. If the model was used for two steps domain adaptation, $\{x_i\}_{i=1}^N$ would be the output samples from the pre-trained StarGAN model. l be the landmark coordinates label provided by the dataset. Our objective contains four terms: adversarial losses[1], cycle consistency loss[4] from CycleGAN, and landmark consistency loss[5] and local discriminator loss[5] from Landmark-Assisted CycleGAN. The losses from CycleGAN reinforced the criteria that the images generated from X to Y look like images from Y, and that the losses from Landmark-Assisted CycleGAN maintain the consistency of the facial features between X and Y.

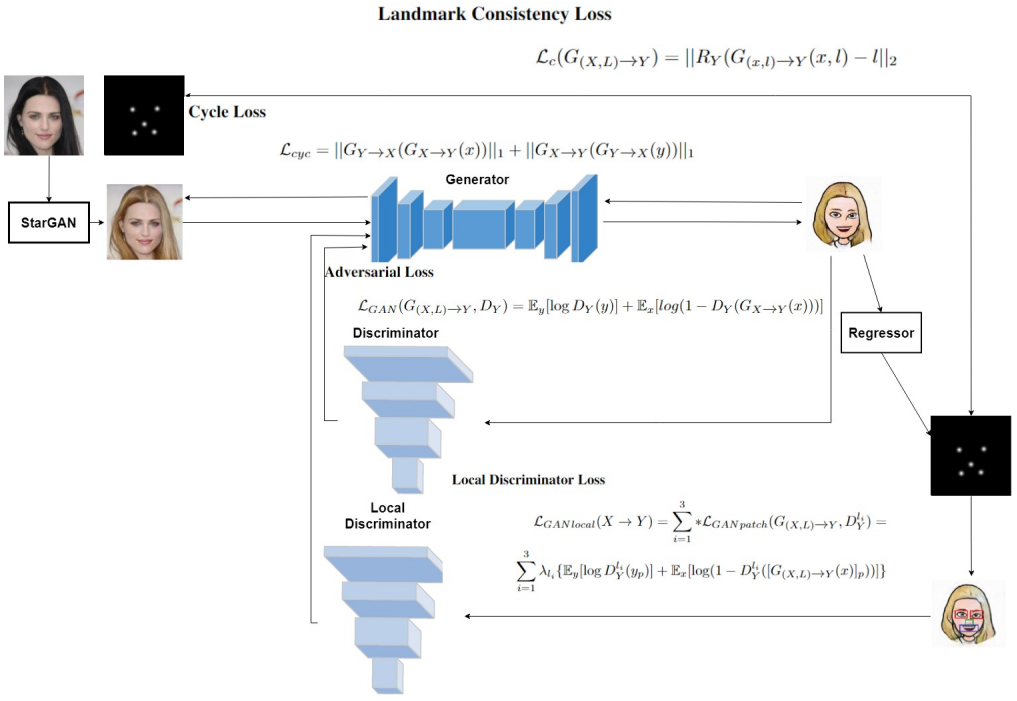

Figure 1: Diagram demonstration of Landmark-Assisted StarGAN

The half-cycle process ($X \to Y$) was demonstrated in Figure 1. First, one image from the CelebA dataset[7] was inputted into the pre-trained StarGAN model to get $x_1 \in \{x_i\}_{i=1}^N$. x_1 was then inputted into generator $G_{(X,L)\to Y}$ to generate a rough image. Similar to GAN model, by feeding this rough image to the discriminator D_Y the model learned how to generate a realistic cartoon image.

Adversarial loss was generated in this stage. Then this same image was fed into the pre-trained regressor to generate landmark coordinates of the facial attributes (eyes, nose, and mouth). The landmark consistency loss was then calculated using the L-2 norm between inputted landmark label $y_1 \in \{y_i\}_{i=1}^{N}$ and the generated landmark coordinates. The generated image was then cropped into two eyes patches, one nose patch, and one mouth patch (two patches for the eye were combined into one patch) based on the generated landmark coordinates. The cropped patches were fed into the eyes, nose, and mouth local discriminator respectively so that the model could learn the cartoon facial features by minimizing the local discriminator loss. The other half of the cycle ($Y \rightarrow X$) had the same structure. Similar to CycleGAN, Cycle Consistency Loss was Incorporated in the full-cycle process.

3.2 CycleGAN losses

CycleGAN, as the baseline of our model, introduced a cycle consistency loss in addition to the adversarial loss from original GAN model. Equation 1 shows the adversarial loss and equation 2 shows the cycle consistency loss.

3.2.1 Adversarial Loss

Adversarial loss[1] was applied so that the discriminator D_Y was able to decide whether the images generated from the generator $G(X, L)$ were fake or not. The generator G tried to minimize $\mathcal{L}_{GAN}(G_{(X,L)\rightarrow Y}, D_Y)$ while D tried to maximize it. By training the model in an adversarial manner, G learned how to generate more realistic images in domain Y while the discriminator learned how to decide whether the images were similar as the real images from the input.

$$\mathcal{L}_{GAN}(G_{(X,L)\rightarrow Y}, D_Y) = \mathbb{E}_y[\log D_Y(y)] + \mathbb{E}_x[log(1 - D_Y(G_{X\rightarrow Y}(x)))] \tag{1}$$

3.2.2 Cycle Consistency Loss

CycleGAN learns the forward and backward mapping simultaneously. The distance between the reconstructed images and the input images was minimized using L-1 norm to maintain cycle consistency. Cycle consistency loss[4] was introduced to prevent the learned mappings contradicting each other.

$$\mathcal{L}_{cyc} = ||G_{Y\rightarrow X}(G_{X\rightarrow Y}(x))||_1 + ||G_{X\rightarrow Y}(G_{Y\rightarrow X}(y))||_1 \tag{2}$$

3.2.3 CycleGAN Full Objective

The full objective of CycleGAN is described in equation 6. By multiplying λ_{cyc} with the cycle consistency loss, the parameters could be tuned to adjust the weight of the loss.

$$\mathcal{L}(G_{X\rightarrow Y}, G_{Y\rightarrow X}, D_X, D_Y) = \mathcal{L}_{GAN}(G_{X\rightarrow Y}, D_Y) + \mathcal{L}_{GAN}(G_{Y\rightarrow X}, D_X) + \lambda_{cyc}\mathcal{L}_{cyc} \tag{3}$$

3.3 Landmark Consistency Loss

The Landmark Consistency Loss (LCL) reinforced the criteria that the coordinates of the landmarks between the human and cartoon domain must remain the same. As different facial expressions are in part due to the location of these landmarks, the LCL helps to map the expression of the original to the cartoon. The LCL calculated the L2 Loss between the landmark coordinates of the original human image l and $R_Y(G_{(x,l)\rightarrow Y}(x, l))$, which was the coordinate map predicted by the regressor with the generated cartoon image as the input.

$$\mathcal{L}_c(G_{(X,L)\rightarrow Y}) = ||R_Y(G_{(x,l)\rightarrow Y}(x, l)) - l||_2 \tag{4}$$

3.4 Local Discriminator Loss

The Local Discriminator Loss (LDL) was the adversarial loss for the eyes, nose, and mouth patches. Using the coordinates returned by the regressor for the generated human and generated cartoon, we extracted the patches of the local landmarks. Following Wu *et al* [5], the eyes were combined together into one image while the nose and mouth were kept separate. Notably, our implementation padded each patch to become a 64x64 image prior to inserting them into their respective local discriminators. The dimensions of each patch before padding were 32×32 for eyes, 28×24 for the nose, and 23×40 for the mouth as specified by Wu *et al* [5].

λ_l is the coefficient representing how much the Local Discriminator Loss was weighted into the total objective function. $D_Y^{l_i}(y_p)$ was the result of the local discriminator with the real image input. $D_Y^{l_i}([G_{(X,L)\to Y}(x)$ was the result of the local discriminator with the generated image as the input.

$$\begin{aligned}\mathcal{L}_{GANlocal}(X \to Y) = \sum_{i=1}^{3} * \mathcal{L}_{GANpatch}(G_{(X,L)\to Y}, D_Y^{l_i}) = \\ \sum_{i=1}^{3} \lambda_{l_i} \mathbb{E}_y[\log D_Y^{l_i}(y_p)] + \mathbb{E}_x[\log(1 - D_Y^{l_i}([G_{(X,L)\to Y}(x)]_p))]\end{aligned} \tag{5}$$

3.5 Full Objective

$$\begin{aligned}\mathcal{L}(G_{X\to Y}, G_{Y\to X}, D_X, D_Y) = \mathcal{L}_{GAN}(G_{X\to Y}, D_Y) + \mathcal{L}_{GAN}(G_{Y\to X}, D_X) \\ +\lambda_{cyc}\mathcal{L}_{cyc} + \lambda_c(\mathcal{L}_c(G_{(X,L)\to Y}) + \mathcal{L}_c(G_{(Y,L)\to X})) \\ +\mathcal{L}_{GANlocal}(X \to Y) + \mathcal{L}_{GANlocal}(Y \to X)\end{aligned} \tag{6}$$

3.6 Landmark Regressor

As noted, the role of the regressor is to identify the landmark coordinates. Compared to the approach of Wu *et al* [5], this project uses a TCDCN architecture, based on Zhang et al's TCDCN[8], rather than a U-Net one. This was due to memory limitations and lack of computational power. Both the human and cartoon domains require a separately trained regressor.

4 Results

4.1 Image Pre-Processing

Following the example of Wu et. al.[5], we selected images from the CelebA dataset that featured the subject facing-forward as opposed to side or profile pictures. Our approach to filtering out the involves the Perspective-n-Point algorithm. We applied the landmark coordinates for the eyes, mouth, and nose provided in the CelebA dataset as our 2D coordinates and used well-known values for the camera matrix and 3D model coordinates. OpenCV has an implementation of Perspective-n-Point that returns a rotation and translation matrix that can map our 2D landmarks onto a 3d plane. We then projected a point [0, 0,1000] in that 3D plane using the rotation and translation matrices and found the angle between our nose landmark and that projected point to find the angle of the head pose.

The result is a crude estimation of the Euler angle yaw, which helped us filter out side-profile images. This method shrinks the original 202,599 dataset down to a smaller 62504 forward-facing images, which is greater in size than the selected 37,794 images of Wu *et al* [5]. Following this, we resized the forward facing images to 128x128.

Bitmojis, which are commonly used in social media, are ideal for the cartoon domain due to the variety of visages and physical features available. This allows Bitmojis to accurately capture the expressiveness of real-life. Our bitmoji dataset was downloaded directly from kaggle[9]. This data

Figure 2: Comparison: Profile Image (left) vs Front-Facing Image (right)

set required no preprocessing steps other than resizing to 128x128. The data set is also notably not the same set used by Wu *et al* [5].

4.2 Landmark Regressor Results

As noted, we implemented Zhang et al's TCDCN[8] for landmark detection.

Our results with this model are summarized below:

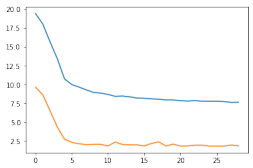

Figure 3: Training and Validation Loss Curve for CelebA.

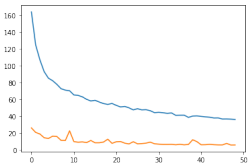

Figure 4: Training and Validation Loss Curve for Bitmoji.

Figure 5: Figures 3 and 4 show the training curves with both the training loss (blue) and validation loss (orange) for the human and cartoon regressors.

The test loss and the average MSE for CelebA and Bitmoji were 1.9, 4.39% and 6.1, 10.45% respectively. Based on our high test set accuracy and low MSE loss, the TCDCN should be competent enough to recognize the facial landmarks for the landmark consistency loss in place of the U-Net proposed by Wu *et al* [5]. The landmark generation results are shown in Figure 6 and Figure 7. The green dot represents the landmark ground-truth labels from the data set, while the red cross represents the landmark generated by the regressor.

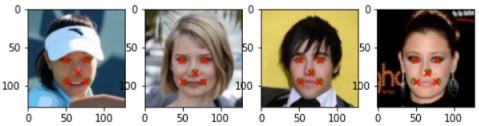

Figure 6: Landmark Detection Results from CelebA

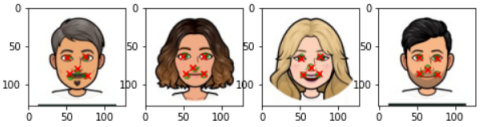

Figure 7: Landmark Detection Results from Bitmoji

4.3 Experiment Settings and Hyper-Parameters

We split the dataset into training and testing samples using the 80/20 split.

Prior to building our combined architecture, we planned to build both the StarGAN and Landmark-assisted Cycle GAN separately. Thus, we trained an instance of StarGan on the CelebA dataset with the target domain being 'blonde hair'.

We adopted the two-stage strategy in [5]. We first trained the basic CycleGAN model with the landmark consistency loss with 60 epochs. We observed a pattern of over-fitting multiple runs after 60 epochs, indicating that the model was fully trained. The first training stage trained the model

to generate a coarse result but was unable to capture the details of the facial features at thus stage. Therefore, at the second stage, we added the local discriminator to the model and trained for 40 more epochs.

We set the batch size to 1, and learning rate to 0.00001. For loss hyper-parameter, we set $\lambda_{cyc} = 10$, $\lambda_c = 1$, $\lambda_{l_i} = 0.3$. These represent non-trainable coefficients that determine how much each loss is weighted into the objective function.

4.4 Results analysis and ablation study

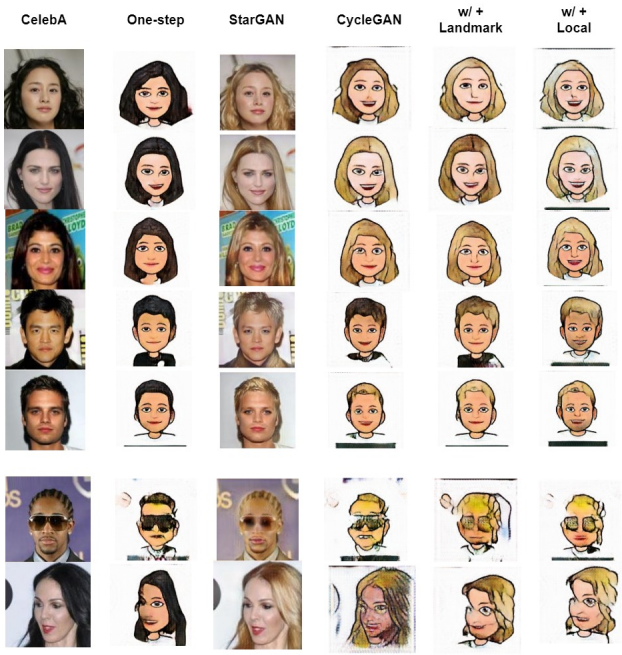

Figure 8: Ablation Study. Column 1 (left-most) is the original image from CelebA dataset and column 3 is the result after inputting it into blonde StarGAN. Columns 4-6 contain the vanilla CycleGAN, CycleGAN + Landmark Consistency Loss, CycleGAN + Landmark Consistency Loss + Local Discriminator Loss. Column 2 is the one-step domain adaptation from the original CelebA to the final cartoon.

As shown in Figure 8, the quality of the generated images improved gradually from baseline (column 2) to our full model (column 6). To determine the effect each loss had on the overall model, the model was trained by progressively adding the different losses (columns 4-6) for the ablation study. By comparing the results of one-step domain adaptation and two-step domain adaptation (column 2 and 6), the model successfully translated the human image to cartoon image by retaining the facial expression, hairstyles, etc.

The last two rows showcased the limitations of our model. Our model had lower performance on darker skin tones and non-frontal facing images. Obstructions to the face such as glasses and complex backgrounds also hindered results as they were rarely present in the Bitmoji dataset. Additionally, StarGAN adds a light-skin bias when converting to blond and may unintentionally change the perceived race as a result when converting to a Bitmoji.

	CycleGAN	Landmark-Assisted StarGAN
FID score	121	98

Table 1: FID score

The Fréchet Inception Distance (FID) score [10] was used to evaluate our model against the baseline. FID is a common standard to evaluate the performance of GAN models by comparing the similarity between two set of images, which were the image test set and the generated image set. In our setting, we calculated the FID score in our test set of 697 samples. The result is shown in table 1. Note that a smaller score is better as it indicates more similarities to the validation set. In conclusion, we achieved a better/lower score compared to the baseline CycleGAN.

5 Limitation and Future work

In conclusion, the Landmark Assisted StarGAN achieved reasonable success with multiple domain adaptations with the majority of images resembling Bitmoji representations of the original subjects. Future implementations will expand the dataset to include more samples of subjects with darker skin tone and non-frontal facing images for better generalization and robustness. Additionally, we plan to introduce regularization techniques during the pre-processing stage focusing on reducing the effects of different lighting and backgrounds for a more robust model.

5.1 Individual Member Contributions

The contributions of Zeting Luan are summarized below:

- Served as lead designer. Researched possible approaches to implement for the project
- Advised in matters regarding Latex and Pytorch syntax.
- Created base model for CycleGAN
- Co-Developed the regressor model architecture with Sean Tonthat
- Integrated the landmark consistency loss into the CycleGAN implementation
- Trained the pre-trained StarGAN implementation with the project CelebA dataset
- Co-Developed the live-demonstration script with Sean Tonthat

The contributions of Sean Tonthat are summarized below:

- Created idea for project
- Served as project manager by organizing meetings, setting deadlines, and dividing workload
- Performed all data set Pre-Processing including filtering out non-frontal images
- Designed architecture for local discriminators
- Co-Developed the regressor model architecture with Zeting Luan
- Integrated the Local Discriminator Loss into the CycleGAN implementation
- Co-Developed the live-demonstration script with Zeting Luan

Both members put an equal amount of writing into the proposal, poster, report, and final paper though it should be noted that Zeting's knowledge of Latex was instrumental to the professional look of the papers. Each member contributed equally in terms of preparation for every meeting.

References

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[2] Sheng-Wei Huang, Alex Lin, Shu-Ping Chen, Yen-Yi Wu, Po-Hao Hsu, and Shang-Hong Lai. *AugGAN: Cross Domain Adaptation with GAN-Based Data Augmentation: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part IX*, pages 731–744. 09 2018.

[3] Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation, 2019.

[4] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2017.

[5] Ruizheng Wu, Xiaodong Gu, Xin Tao, Xiaoyong Shen, Yu-Wing Tai, and J iaya Jia. Landmark assisted cyclegan for cartoon face generation, 2019.

[6] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. 2017.

[7] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[8] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 94–108, Cham, 2014. Springer International Publishing.

[9] Mozafari M. Bitmoji faces, version 1. `https://www.kaggle.com/datasets/mostafamozafari/bitmoji-faces`, 2020. retreived May 5, 2022.

[10] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500, 2017.

6 Appendex

6.1 Details of Regressor and Local Discriminator

6.1.1 Regressor Architecture

The architecture for the landmark regressor is summarized below:

Layer	Output Size	Kernel/Pooling Size
Inputlayer	(128,128,3)	(-,-)
Conv2D-1	(124,124,20)	(5,5)
MaxPool-2	(62,62,20)	(2,2)
Conv2D-3	(58,58,48)	(5,5)
MaxPool-4	(29,29,48)	(2,2)
Conv2D-5	(27,27,64)	(3,3)
MaxPool-6	(13,13,64)	(2,2)
Conv2D-7	(11,11,80)	(3,3)
Flatten-8	(9680)	(-,-)
Dropout-9	(9680)	(-,-)
FCL-10	(256)	(-,-)
Dropout-11	(256)	(-,-)
Output	(10)	(-,-)

Table 2: Landmark Regressor architecture

6.1.2 Local Landmark Discriminator Architecture

The architecture for the landmark regressor is summarized below:

Layer	Output Size	Kernel/Pooling Size
Inputlayer	(64, 64, 3)	(-,-)
Conv2D-1	(32, 32, 32)	(4,4)
LeakyReLU-2	(32, 32, 32)	(-,-)
Conv2D-3	(16, 16, 64)	(4,4)
BatchNorm2D-4	(16, 16, 64)	(-,-)
LeakyReLU-5	(16, 16, 64)	(-,-)
Conv2D-6	(8, 8, 128)	(4,4)
BatchNorm2D-7	(8, 8, 128)	(-,-)
LeakyReLU-8	(8, 8, 128)	(-,-)
Conv2D-9	(4, 4, 256)	(4,4)
BatchNorm2D-10	(4, 4, 256)	(-,-)
LeakyReLU-11	(4, 4, 256)	(-,-)
Conv2D-12	(1,1,1)	(4,4)
Sigmoid-13	(1,1,1)	(-,-)
Flatten-14	(1)	(-,-)
Output	(1)	(-,-)

Table 3: Landmark Regressor architecture