AN INVESTIGATION INTO STARGAN CROSS-DOMAIN IMAGE GENERATION WITH DIFFERENT DATASETS

A PREPRINT

Uy, Mark Christopher Siy (2046559)
Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon
mcsuy@connect.ust.hk

Suen, Heung Ping (20271291)
Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon
hpsuenaa@connect.ust.hk

Kim, Jaeyeon(20528606)
Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon
jaeyeon.kim@connect.ust.hk

November 25, 2020

ABSTRACT

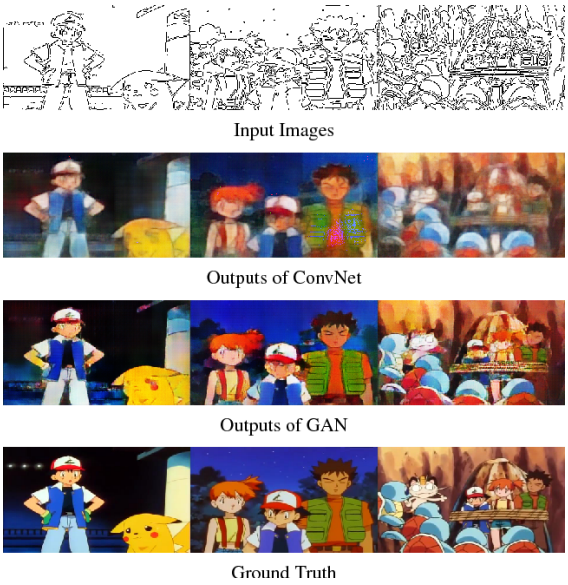
The field of artificially generated images has grown in great popularity within the past few years, with the release of a wide variety of different GANs[1] (Generative Adversarial Networks) models, such as WGAN-GP[2], CycleGAN[3]. While these GANs have demonstrated impressive results, they are primarily designed for one-to-one image translation, which means that resulting neural network is being trained for one specific domain translation. However, this could lead to a combinatorial increase in networks for every single domain translation required. StarGan[4] proposes a solution to this problem with very promising results. This paper desires to explore the solutions proposed by the paper and to see the effectiveness of the network on different datasets. The paper is split into three main sections. The first is an introduction to GANs, and a corresponding review of related literature and models; the next section will be a discussion of the innovations proposed by StarGAN; and the last section will focus on the group's innovations and work.

1 Introduction

GANs (Generative Adversarial Networks) were first introduced by Ian Goodfellow in his seminal paper[4]. They are an example of generative models, which are a class of models that are able to take a training set, consisting of samples drawn from a unknown distribution p_{data}, and learns to represent an estimate of that distribution somehow. The result is a probability distribution p_{model}. GANs are then able to draw sample data from this distribution to create new data points that are different from the original data points in the training set. As a result, GANs have been found to have a wide variety of different uses and applications from enhancing image resolution[5], to art creation (under the style and influence of a specific artist), and even towards animation rendering of a set of images[6]. GANs have also shown to be useful in reinforcement learning tasks, and have the added advantage of being able to work with *multi-modal* outputs.

Due to the wide number of cases mentioned above, GANs have become a very active and popular field of research. Since 2014, the number of GAN related literature and academic papers have sky rocketed with new models and improvements being made on a regular basis. The amount of attention and research focused on the development of GANs underlie the incredible potential that GANs have to create realistic and novel content, whether in image, text, or sound format. Some of these GAN generated images will be shown in the next page in figures 1 and 2.

Figure 1: Super Image Enhancement[5]



(a) Colorization of Images

(b) Painting Generation

Figure 2: GAN Based Artistic Renderings[6]

2 Review of Related Literature

2.1 The GAN Framework

In a simplistic way, the GAN framework can be described as a game between two participants. One of participants is called the generator, and the other is called the discriminator. The generator's goal is to create samples (fake images), from the same distribution as the training data in order to fool the discriminator. The discriminator examines samples from both the training data, and the generator created images in order to distinguish which image is fake or real.

Formally, GANs are a structured probabilistic model composed of latent variables z and observed variables x. The two players in the game are represented by two functions. The discriminator is a function D that takes x as input and uses parameters $\theta^{(D)}$. The generator is defined by a function G that takes z as input and uses $\theta^{(G)}$ as parameters.

Both players have cost functions that are defined in terms of both players' parameters. The discriminator wishes to minimize the cost function, $J^{(D)}(\theta^{(D)}, \theta^{(G)})$ and must do so while adjusting only the $\theta^{(D)}$ parameters. Similarly,

the generator wishes to minimize the cost function, $J^{(G)}(\theta^{(D)},\theta^{(G)})$, while adjusting only the $\theta^{(G)}$ parameters. The solution to the game is a tuple $(\theta^{(D)}, \theta^{(G)})$ that is both a local minimum of $J^{(D)}$ with respect to $\theta^{(D)}$, and a local minimum of $J^{(D)}$ with respect to $\theta^{(G)}$.

The explicit cost function of the generator is:

$$J^{(G)}(\theta^{(D)}, \theta^{(G)}) = -\frac{1}{2}\mathbb{E}_z \log D(G(z)) \tag{1}$$

The explicit cost function of the discriminator is:

$$J^{(D)}(\theta^{(D)}, \theta^{(G)}) = -\frac{1}{2}\mathbb{E}_{x\sim p_{data}} \log D(x) - \frac{1}{2}\mathbb{E}_z \log(1 - D(G(z))) \tag{2}$$

The model can thus be trained by performing gradient descent for each of the two cost functions per iteration or step.

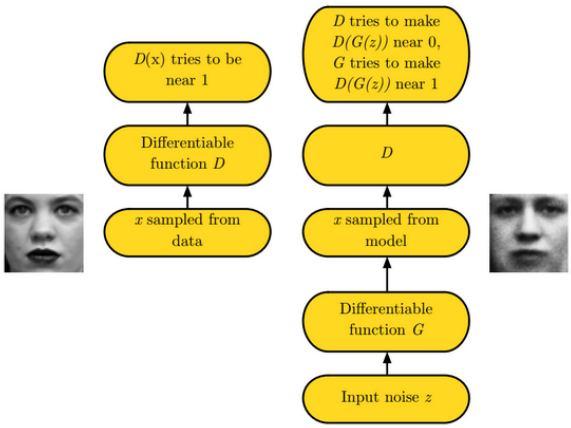


Figure 3: GAN Training Process

2.2 Related Work

Conditional GANs. Conditional GANs build upon classical GANs by only generating image samples from a specific class. In order to do this, some studies have provided both the discriminator and generator functions with class information from the training data [7],[8], [9]. Conditional GANs have also been shown to be effective and useful in a wide variety of tasks such as domain transfer [10], [11] , photo editing [12] and even super-resolution imaging[5].

Image-to-Image Translation. Recent work have achieved impressive results in image-to-image translation. [1],[13],[14] CycleGAN [1] and DiscoGAN [14] preserve key attributes between the in-put and the translated image by utilizing a cycle consistency loss. However, all these frameworks are only capable of learning the relations between two different domains at a time. Their approaches have limited scalability in handling multiple domains since different models should be trained for each pair of domains. This leads to combinatorial rise in complexity to learn every pair of domain relations.

(a) Aerial to Map Translation

(b) Doodle to Image Translation

Figure 4: Image to Image Translation: The two images above show the ability of GANs to transform one image to another by being able to learn features from another domain (Ground Truth)

2.3 Proposed Solution

The goal is to build a generator, G, that translates an image x to an output image y, given a target domain label c, i.e. $G(x, c) \rightarrow y$. As with other GANs, a discriminator D is introduced to compete with G. The model has three objectives:

1. **Indistinguishable Fake Images.** The images generated should be Indistinguishable from the real images. To achieve this, the adversarial loss

$$\mathcal{L}_{adv} = E_x[\log D_{src}(x)] + E_{x,c}[\log(1 - D_{src}(G(x, c)))], \tag{3}$$

   is adopted.

   The first term is the log likelihood of real images classified as real, and the second term is the log likelihood of fake images classified as fake. D learns to maximize L_{adv}, while G learns to minimize it.

   For practical purposes, Equation (3) leads to unstable training. The Wasserstain GAN [2] objective is used instead in implementation.

$$\mathcal{L}_{adv} = E_x[\log D_{src}(x)] - E_{x,c}[\log D_{src}(G(x, c))] - \lambda_{gp} E_{\hat{x},c}[(||\nabla \hat{x} D_{src}(\hat{x})||_2 - 1)^2] \tag{4}$$

2. **Correct Translated Domain.** The output images should be correctly translated and classified to the target domain c. Two conditions are necessary to achieve this. Firstly, the discriminator must classify the domain of an image correctly. Secondly, the generator must translate images to the correct domain.

   To satisfy the first condition, D learns to classify real images to their original domain, c'. The loss function is the log likelihood that real images are classified as their original domain.

$$\mathcal{L}^r_{cls} = E_{x,c'}[-\log D_{cls}(c'|x)] \tag{5}$$

   To satisfy the second condition, G learns to generate fake images that would be classified to the target domain by D. The loss function is the log likelihood that fake images are classified to the target domain c.

$$\mathcal{L}^f_{cls} = E_{x,c}[-\log D_{cls}(c|G(x, c))] \tag{6}$$

3. **Preserved Image Content** The translated images generated should be the same as the input image, except for attribute changes, e.g. the face should remain the same after translations between facial expressions. Given

the translated image $G(x, c)$ and the original image domain c', the generator should be able to reconstruct the original image x, i.e. G learns to minimize:

$$\mathcal{L}_{rec} = E_{x,c,c'}[||x - G(G(x, c), c')||_1] \tag{7}$$

which is the expected reconstruction loss as an L1 norm , after translating an x of domain c' to $G(x, c)$, and translating it back.

Full Loss Functions. L_G and $\mathcal{L}_D$, respectively the full objective functions for G and D are as follows:

$$\mathcal{L}_D = -\mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}^r_{cls} \tag{8}$$

$$\mathcal{L}_G = \mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}^f_{cls} + \lambda_{rec}\mathcal{L}_{rec} \tag{9}$$

where λ_{cls} and λ_{rec} are hyperparameters for the relative importance of classification loss and reconstruction loss respectively.

Training with Multiple Datasets. When training with multiple datasets, e.g. CelebA [15] and RaFD [16], complete label information is not available. Suppose an image x with domain label c' from CelebA is translated to $G(x, c)$, where c is from RaFD, translating $G(x, c)$ back to x is challenging because x's original label c' does not contain information related to labels from RaFD. To address the challenge, the following label vector is used instead of c used in training a single dataset:

$$\tilde{c} = [c_1, ..., c_n, m] \tag{10}$$

where n is the number of datasets, c_i is a one-hot vector for labels of the i-th dataset, and m is n-dimensional one-hot mask vector representing which term c_i is active. In the example above, to translate $G(x, c)$ back to x, m is set to $[1, 0]$ to represent that information from RaFD should be ignored.

2.4 Model Architecture

The model architecture is adopted from CycleGAN. G has three convolutional layers for down-sampling, six residual blocks for the bottleneck layers and a up-sampling part consisting of two decov-layers and one convolutional layer. The model is trained with Adam optimizer.

2.5 Results Compared with Existing Models

In the StarGAN paper experiments are conducted with the CelebA and RaFD datasets. Important results from the papers are cited as follows:

To compare with other models, two surveys were conducted using Amazon Mechanical Turk (AMT). In the surveys, surveyees are asked to choose the best image from images generated by different models. The results show that for multi-domain image translations, StarGAN outperforms other models significantly in terms of image quality.

Method	H+G	H+A	G+A	H+G+A
DIAT	20.4%	15.6%	18.7%	15.6%
CycleGAN	14.0%	12.0%	11.2%	11.9%
IcGAN	18.2%	10.9%	20.3%	20.3%
StarGAN	**47.4%**	**61.5%**	**49.8%**	**52.2%**

Table 1: AMT perceptual evaluation for ranking different models on a multi-attribute transfer task. H: Hair color; G: Gender; A: Aged.

Also, StarGAN uses much fewer number of parameters than the existing models.

Cross-dataset learning is working successfully, with the mask vector controlling which attributes to ignore.

Method	Classification error	# of parameters
DIAT	4.10	52.6M × 7
CycleGAN	5.99	52.6M × 14
IcGAN	8.07	67.8M × 1
StarGAN	**2.12**	**53.2M × 1**
Real images	0.45	-

Table 2: Classification errors [%] and the number of parameters on the RaFD dataset.

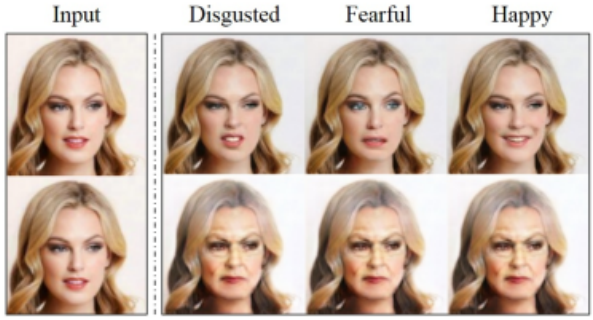


Figure 5: AMT perceptual evaluation for ranking different models on a multi-attribute transfer task. H: Hair color; G: Gender; A: Aged.

3 Group's Innovation and Study

3.1 Result Reproduction with Different Data-sets

For result reproduction, we use the CelebA and UTKFace [17] datasets. UTKFace is a public dataset with over 20,000 images, labels include age, gender and race. For learning purpose, those older than 65 years old are classified as aged, while others are classified as young.

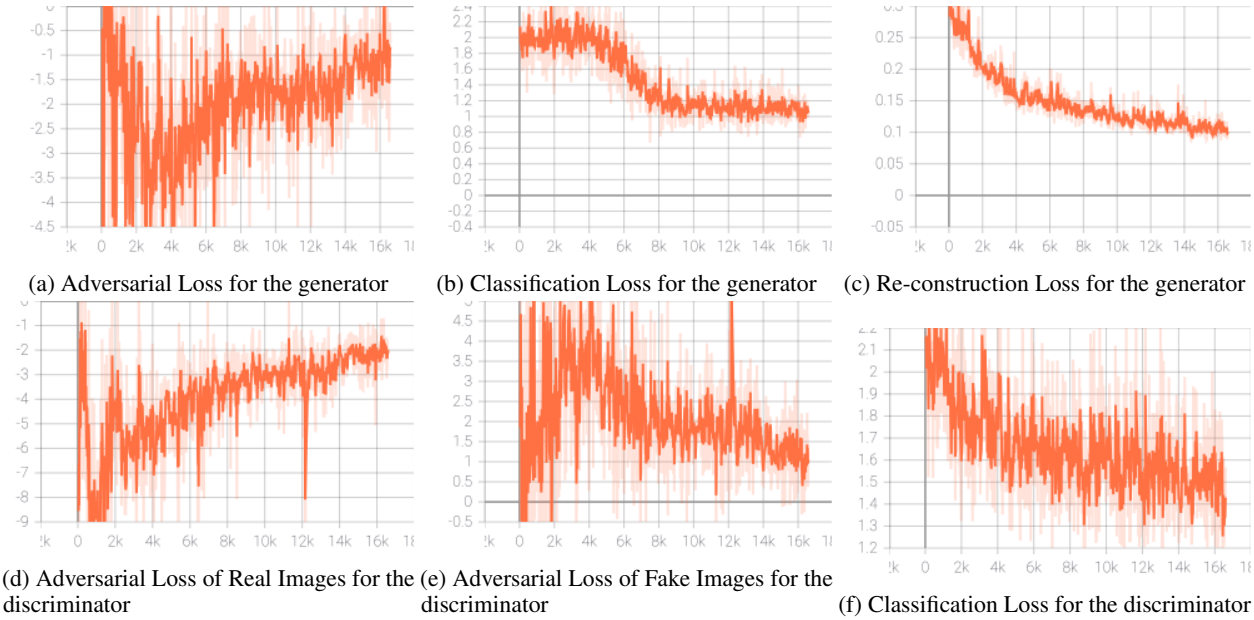
StarGAN is used on CelebA and UTKFace separately, and then both datasets are used together for multi-dataset training. Results are as follows:

Training with UTKFace

The images generated have sharp edges, and perceptually many of the generated images are of good qualities, e.g. translating the woman's image to the 'male' domain is quite successful.

Figure 6: Images generated by StarGAN using UTKFace Dataset, from left to right: original figure, aged, asian, black, female, indian, male, white, young (16000 iterations of training)

However, the generator has the potential to perform better, since the adversarial loss for the generator has been going up, while that for the discriminator has been going down, meaning that the discriminator is outperforming the generator. This might be due to insufficient training iterations due to limited computational resources



(a) Adversarial Loss for the generator

(b) Classification Loss for the generator

(c) Re-construction Loss for the generator

(d) Adversarial Loss of Real Images for the discriminator

(e) Adversarial Loss of Fake Images for the discriminator

(f) Classification Loss for the discriminator

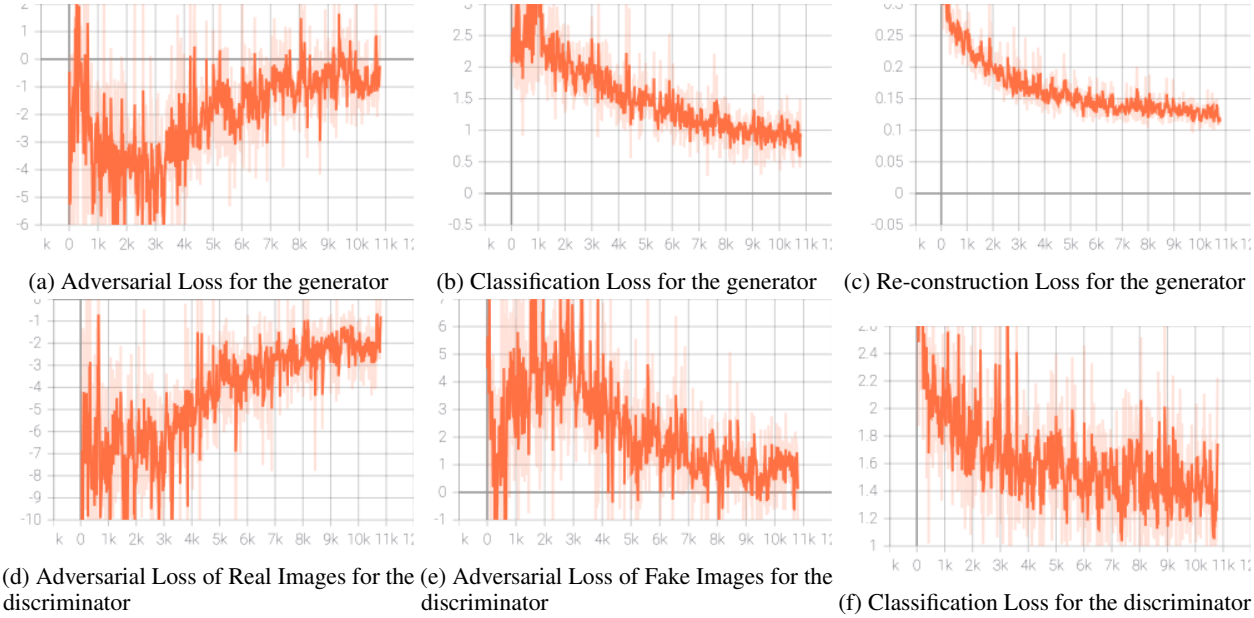
Figure 7: Loss Functions During Training with UTKFace Dataset

Cross Training with CelebA and UTKFace

Many of the images generated still have sharp edges. However perceptually they don't look as good as that generated using a single dataset. One possible explanation is that training across different datasets required more iterations as it is more difficult to learn from different datasets. The model has to learn how to interpret the mask vector m, and the right generalizations across two datasets.

Figure 8: Images generated by cross-training with CelebA and UTKFace. From left to right: original figure, black, female, indian, male, white, young (10000 iterations of training)

Once again due to hardware and computing resources limitations, we are unable to train all the datasets with 200,000 iterations, which is the number of iterations implemented in the official StarGAN's implementation. The reduced number of iterations means that the reproduced results are not the best images generated from the model. While the classification loss and the reconstruction losses have been dropping steadily over time, adversarial loss actually increases, proving that the discriminator is improving near the end of the 10,000 iterations, and the generator might be able to decrease the loss again as the number of iterations increases.



(a) Adversarial Loss for the generator (b) Classification Loss for the generator (c) Re-construction Loss for the generator

(d) Adversarial Loss of Real Images for the discriminator (e) Adversarial Loss of Fake Images for the discriminator (f) Classification Loss for the discriminator

Figure 9: Loss Functions During Training with CelebA and UTKFace Datasets

However, they still demonstrate that multi-domain Image-to-Image translation is working correctly, for human face translations, proving that StarGAN indeed provides a good solution to the problem .

3.2 Performance under Different Parameters

We experiment two parts which are GAN types and the number of residual blocks.

GAN types. StarGAN uses the adversarial losses and gradient penalty of WGAN-GP. We changed the adversarial generator loss, the adversarial discriminator loss ,and gradient penalty according to DRAGAN[18], and LSGAN[19]). WGAN-GP and DRAGAN adapt gradient penalty differently to WGAN and GAN. WGAN-GP checks gradient at a random point. On the contrary, DRAGAN checks gradient at any point which is close to the real sample. LSGAN uses least Square loss in order to reduce the distance of fake data in decision boundary. The loss functions are as follows.

1. **The value function of WGAN-GP**

$$\mathcal{L}_D^{WGAN-GP} = E[D(x)] - E[D(G(z))] + \lambda E[(|\nabla D(ax - (1 - aG(z)))| - 1)^2] \tag{11}$$

$$\mathcal{L}_G^{WGAN-GP} = E[D(G(z))] \tag{12}$$

2. **The value function of DRAGAN**

$$\mathcal{L}_D^{DRAGAN} = E[log(D(x))] + E[log(1 - D(G(z)))] + \lambda E[(|\nabla D(ax - (1 - ax_p))| - 1^2] \tag{13}$$

$$\mathcal{L}_G^{DRAGAN} = E[log(D(G(z)))] \tag{14}$$

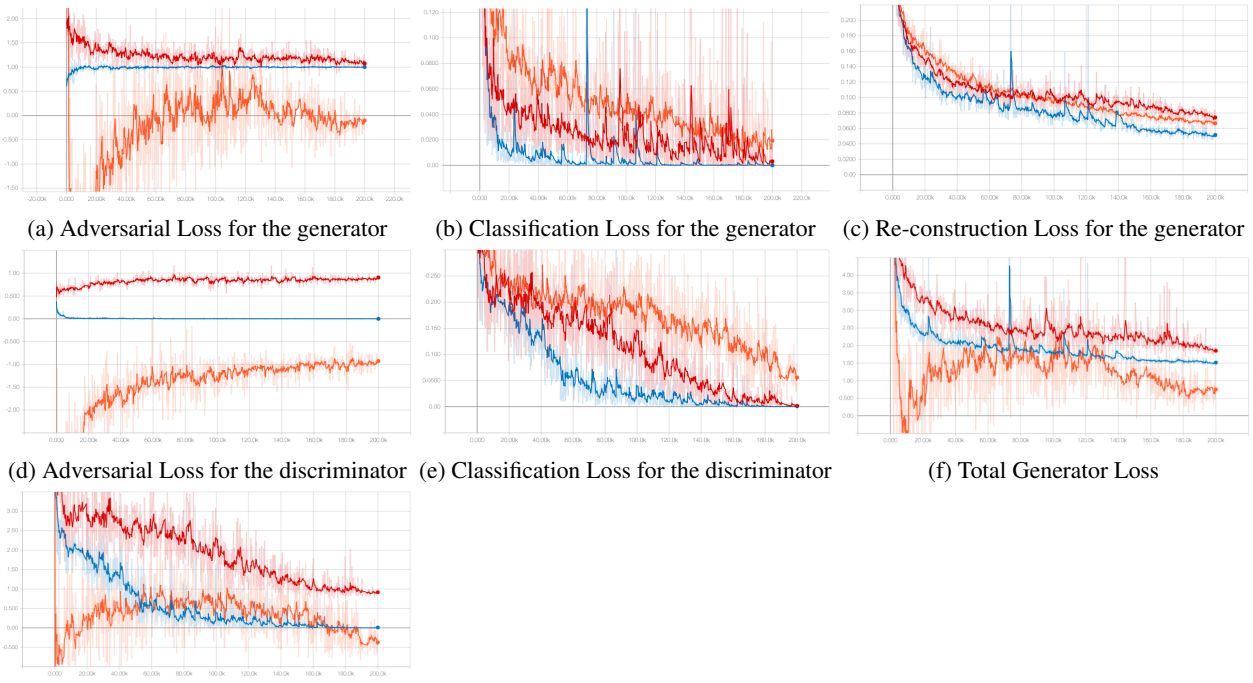
3. **The value function of LSGAN**

$$\mathcal{L}_D^{LSGAN} = E[(D(x) - 1^2)] + E[D(G(z))^2] \tag{15}$$

$$\mathcal{L}_G^{LSGAN} = E[(D(G(z)) - 1)^2] \tag{16}$$

We trained the CelebA dataset with 200, 000 iteration, respectively. We identify the fake images of WGAN-GP are most realistic, and LSGAN cannot make generate images with good quality(See 10). Although the total generator loss and discriminator loss of three GANs are decreased gradually, there is a difference in the adversarial losses. We confirm the model using WGAN-GP shows the best training and next is DRAGAN's model. However, the adversarial losses of LSGAN do not change and it means the model cannot train with data. It is the reason that LSGAN makes worse fake images(See 11). We speculate that the gradient penalty is one of the reasons not to work for LSGAN or the loss function of LSGAN cannot update the gradient due not to expressing the distance between real and fake images.

Figure 10: Facial attribute transfer results on the CelebA dataset by GAN types



(a) Adversarial Loss for the generator (b) Classification Loss for the generator (c) Re-construction Loss for the generator

(d) Adversarial Loss for the discriminator (e) Classification Loss for the discriminator (f) Total Generator Loss

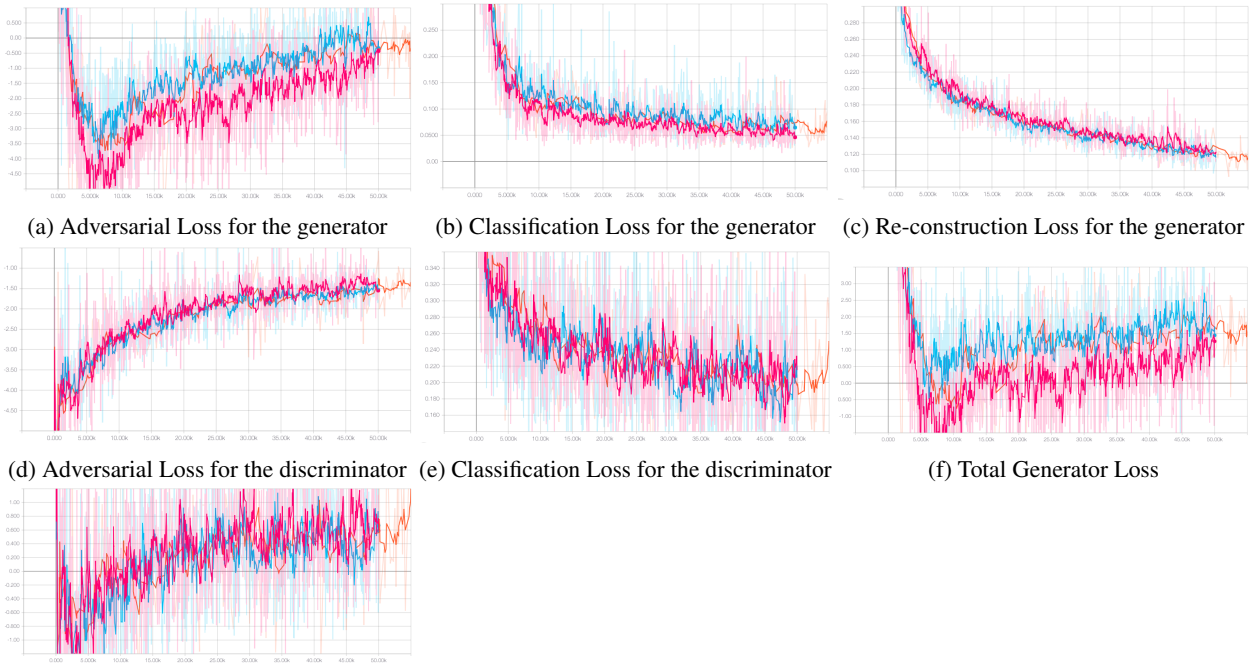
(g) Total Discriminator Loss

Figure 11: Loss functions on the celebA dataset by GAN: WGAN-GP(orange), DRAGAN(Red) and LSGAN(Blue)

The number of residual blocks. We changed the number of residual blocks(3, 6, 20). According to the original paper, the generator network architecture is adapted from CycleGAN. The residual blocks are part of the bottleneck layer in the generator network for transforming the original information. Moreover, the residual block ensures that the output vector cannot deviate from the original image very much by adding previous layers to later layers. In the CycleGAN paper, they use more residual blocks for higher resolution images. Hence, we expect the more residual blocks can improve the generator and can retain the character of input such as volume and aspect. We train the CelebA dataset with 50,000 iterations, respectively. Although the adversarial loss of generator goes decreased, as the number of block

Figure 12: Facial attribute transfer results on the CelebA dataset by the number of residual blocks

increases, the shape and value of the generator loss and discriminator loss are similar.(See 13) In addition, the samples don't make much difference.(See 12) The lack of iteration is one reason for this result. Because the losses increased gradually during the training and 50,000 iterations and we confirm this iteration is not enough compared with the graph of 200,000 iterations. Therefore, we have to train more than 200,000 iterations for checking the influence of the number of residual blocks.



(a) Adversarial Loss for the generator (b) Classification Loss for the generator (c) Re-construction Loss for the generator

(d) Adversarial Loss for the discriminator (e) Classification Loss for the discriminator (f) Total Generator Loss

(g) Total Discriminator Loss

Figure 13: Loss functions on the celebA dataset by GAN: 3 blocks(blue), 6 blocks(orange) and 20 blocks(pink)

3.3 Cross-Training with Non-human-face Data Sets

In this section, we explore the performance of the original starGAN model when used with other data sets. Unlike the UTK or Radboud data set, which are both human face data sets, this section focused on trying to potentially integrate human and animal faces by co-training the CelebA data set and an all animal data set, which is the KTH-ANIMALS[21] data set.

Observations about the Data Set The data set has 17 different classes of animals. However, for the experiment, only 8 out of these 17 classes we selected. This is because the paper also used 8 features from the Radboud data-set. The 8 classes are the following: Bear, Cougar, Cow, Coyote, lion, panda, penguin, and sheep. The data-set is also quite small with approximately 80 to 100 images for each animal. Due to the small size of the data set, a 90-10 train-test split was used.

Training Results The results of training can be seen in Figure 14. Due to the heavy requirements of the data set, and the GAN model, only 80,000 iterations were able to be accomplished rather than the full 200,000 proposed in the original starGAN model. Looking at the performance in figure 14, some interesting observations can be seen. The discriminator is clearly performing worse than the generator. This is probably due to the difference between the two data sets being used since the discriminator is finding it hard to generalize from one data set to another. This is reflected in the giant fluctuations in the adversarial loss for the real images. It can also be seen that the adversarial loss for fake images seems to be increasing after 50,000 epochs, which signifies to us some over-fitting is occurring. This might be a product of the small data-set, which was used. Overall, it seems that the model does not work too well when trying to co-learn from completely different domains, i.e. animal features and human faces.

Cross Training with CelebA and KTH-ANIMALS

The last 8 images in each row correspond to the 8 different classes of animals: Bear, Cougar, Cow, Coyote, Lion, Panda, Penguin, Sheep. It is clear that the generated images are nowhere as clear or as nice as those generated in the original paper Many of the images generated still have sharp edges, with a confusing mix of colors. One possible explanation for these colors is that the color of the animals is one of the primary features learned by the generator. Overall, all images are still quite humanoid and don't resemble much the animal they are supposed to except for the panda. The panda seems to show decent results in that the resulting image does indeed resemble a panda, with the black eyes and thick nose. One interesting observation is the dominance of the color green for the sheep and cow generated images, which may come from the grass surroundings of the animal in the images. As a result, it seems that more image pre-processing should have been performed. This is definitely one direction for further research in the future. Overall though, quality is far from what is desired, meaning that the model has not learned the right generalizations across two data sets.

Figure 14: Images generated by cross-training with CelebA and KTH-ANIMALS. From left to right: original, Black-Hair, Blond-Hair, Brown-Hair, Male, Young, Bear, Cougar, Cow, Coyote, Lion, Panda, Penguin, Sheep (80000 iterations of training)

4 Conclusion

The investigation into StarGAN shows that the model is capable of learning multi-domain Image-to-Image task successfully across different sets of images that are homogeneous with each other, e.g. human faces. However, cross-training with datasets that are not homogeneous with each other, e.g. human faces and non-human faces are not as effective. StarGAN may be more useful for mapping between subtle image features, e.g. facial expressions, than transferring features between distinct datasets.

5 Contributions from Team Members

UY, Mark Christopher(20466559)
Uy, Mark Christopher is responsible for finding a new data-set, and using it to co-train the StarGan model with celebA

data-set. He also did necessary image pre-processing work. He also evaluated the results by training the model on the cloud.

SUEN, Heung Ping(20271291)
SUEN, Heung Ping is responsible for reproducing the results by using the UTKFace, which was not used by the original research team, as well as cross-training the UTKFace Dataset with CelebA dataset. He evaluated the results based on a smaller number of iterations.

Jaeyeon, Kim(20528606)
Jaeyeon Kim is responsible for performance under Different Parameters. She experiments the difference of Loss and gradient penalty according to GAN types. She also changed the number of residual blocks in the generator for confirming the improvement of the generator. She also evaluated the results by training the model.

References

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks", *ArXiv e-prints*, arXiv:1406.2661, arXiv:1406.2661, Jun. 2014. arXiv: 1406.2661 [stat.ML].

[2] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN", *ArXiv e-prints*, arXiv:1701.07875, arXiv:1701.07875, Jan. 2017. arXiv: 1701.07875 [stat.ML].

[3] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks", *CoRR*, vol. abs/1703.10593, 2017. arXiv: 1703.10593. [Online]. Available: http://arxiv.org/abs/1703.10593.

[4] 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", *ArXiv e-prints*, arXiv:1711.09020, arXiv:1711.09020, Nov. 2017. arXiv: 1711.09020 [cs.CV].

[5] 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. arXiv: 1609.04802. [Online]. Available: http://arxiv.org/abs/1609.04802.

[6] K. Nazeri and E. Ng, "Image colorization with generative adversarial networks", *CoRR*, vol. abs/1803.05400, 2018. arXiv: 1803.05400. [Online]. Available: http://arxiv.org/abs/1803.05400.

[7] A. Odena, C. Olah, and J. Shlens, "Conditional Image Synthesis With Auxiliary Classifier GANs", *ArXiv e-prints*, arXiv:1610.09585, arXiv:1610.09585, Oct. 2016. arXiv: 1610.09585 [stat.ML].

[8] A. Odena, "Semi-Supervised Learning with Generative Adversarial Networks", *ArXiv e-prints*, arXiv:1606.01583, arXiv:1606.01583, Jun. 2016. arXiv: 1606.01583 [stat.ML].

[9] M. Mirza and S. Osindero, "Conditional generative adversarial nets", *CoRR*, vol. abs/1411.1784, 2014. arXiv: 1411.1784. [Online]. Available: http://arxiv.org/abs/1411.1784.

[10] Y. Taigman, A. Polyak, and L. Wolf, "Unsupervised cross-domain image generation", *CoRR*, vol. abs/1611.02200, 2016. arXiv: 1611.02200. [Online]. Available: http://arxiv.org/abs/1611.02200.

[11] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks", *CoRR*, vol. abs/1703.05192, 2017. arXiv: 1703.05192. [Online]. Available: http://arxiv.org/abs/1703.05192.

[12] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, "Neural photo editing with introspective adversarial networks", *CoRR*, vol. abs/1609.07093, 2016. arXiv: 1609.07093. [Online]. Available: http://arxiv.org/abs/1609.07093.

[13] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks", *CoRR*, vol. abs/1611.07004, 2016. arXiv: 1611.07004. [Online]. Available: http://arxiv.org/abs/1611.07004.

[14] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks", in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, Eds., ser. Proceedings of Machine Learning Research, vol. 70, International Convention Centre, Sydney, Australia: PMLR, Jun. 2017, pp. 1857–1865. [Online]. Available: http://proceedings.mlr.press/v70/kim17a.html.

[15] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild", in *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[16] O. Langner, R. Dotsch, G. Bijlstra, D. H. J. Wigboldus, S. T. Hawk, and A. van Knippenberg, "Presentation and validation of the radboud faces database", *Cognition & Emotion*, vol. 24, no. 8, pp. 1377–1388, Dec. 2010. DOI: 10.1080/02699930903485076. [Online]. Available: https://doi.org/10.1080/02699930903485076.

[17] Z. Zhang, Y. Song, and H. Qi, "Age progression/regression by conditional adversarial autoencoder", in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017.

[18] N. Kodali, J. D. Abernethy, J. Hays, and Z. Kira, "How to train your DRAGAN", *CoRR*, vol. abs/1705.07215, 2017. arXiv: 1705.07215. [Online]. Available: http://arxiv.org/abs/1705.07215.

[19] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, and Z. Wang, "Multi-class generative adversarial networks with the L2 loss function", *CoRR*, vol. abs/1611.04076, 2016. arXiv: 1611.04076. [Online]. Available: http://arxiv.org/abs/1611.04076.

[20] Joint Visual Vocabulary for Animal Classification- Heydar Maboudi Afkham, Alireza Tavakoli Tar ghi, Jan-oluf Eklundh, and Andrzej Pronobis In Proceedings of the International Conference on Pattern Recognition (ICPR08), Tampa, FL, USA, December 2008.