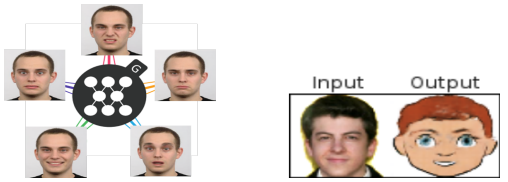
Facial Expression Generation

Ssu-Yu, Chang (106062555) **Zhi-Kuan, Wang (106062601)**
{adam9500370, moopene2017}@gmail.com

Abstract

We want to synthesize images with different emotions for certain person by multi-domain image-to-image emotion transfer. We augment StarGAN with perceptual loss, residual connection, and some GAN tricks. In the experiments, we evaluate facial expression recognition on baseline and our method, the performance of our method is better than the baseline in recognition evaluation. Our code is available at https://github.com/adam9500370/facial_expression_generation.

1 Introduction



(a) Emotion transfer. (b) From real to virtual.

Figure 1: Main ideas in our project.

In real world, we may only see someone's face with some part of emotions. We may never know someone's face with other emotions. We can only imagine what someone's face with certain emotion is by self-experiences from a variety of facial expressions of other people.

In this project, we want to synthesize images with different emotions for certain person by unpaired multi-domain image-to-image emotion transfer method [1]. We focus on generating images with emotional attributes among 7 kinds of facial expressions (Angry, Disgust, Fear, Happy, Neutral, Sad, Surprise).

Furthermore, we also want to generate images from real human to virtual character [2] with emotion transfer. It is more challenging to transfer two kinds of domains, not only emotion attributes, but also translations from real human to virtual character.

Our contribution can be listed as followings:

- Perceptual loss in D
- Skip (residual) connection in G
- Some GAN tricks (Dropout + LReLu in G)

2 Related work

StarGAN. StarGAN [1] is capable of learning mapping among multiple domains and have shown remarkable results in multi-domain image-to-image translation. Similar to typical Generative Ad-

versarial Network, a StarGAN consists of a generator G and a discriminator D. G takes input images and target domain as input, and output fake images which make D can't distinguish from real images and classifies them to target domain. D distinguish between real and fake images, and besides, it classify the real images to its original domain.

Perceptual Loss. Recently, perceptual loss [3] has been proved that it can better reconstruct fine details compared to methods trained with per-pixel loss. Instead of measuring the similarity in the image space, measuring in a feature space is more reasonable. Therefore, we follow the suggestion and apply perceptual loss.

Skip Connection. In order to learn residual features between different emotions, We use skip-connection to connect mirrored layers in the down-sampling and up-sampling stacks [3].

3 Our Method

We augment original StarGAN [1] with the following techniques, including perceptual loss in discriminator, skip connection in generator, and some GAN tricks in generator.

3.1 StarGAN

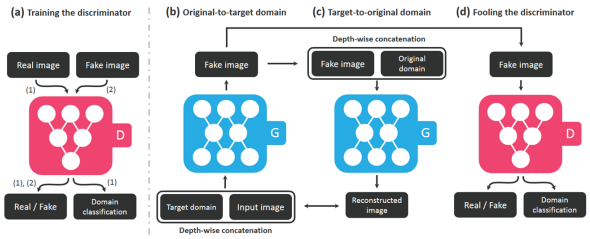


Figure 2: Overview of StarGAN model.

StarGAN is a multi-domain image-to-image translation method, using only a single generator and a single discriminator (shown in Fig. 2).

Adopted loss in StarGAN can be listed as followings:

- Adversarial loss (WGAN objective with gradient penalty)

$$\begin{aligned}\mathcal{L}_{adv} = \mathbb{E}_x[D_{src}(x)] - \mathbb{E}_{x,c}[D_{src}(G(x,c))] \\ - \lambda_{gp}\mathbb{E}_{\hat{x}}[(\|\nabla_{\hat{x}} D_{src}(\hat{x})\|_2 - 1)^2]\end{aligned} \tag{1}$$

- Domain classification loss

$$\begin{aligned}\mathcal{L}^r_{cls} &= \mathbb{E}_{x,c'}[-\log D_{cls}(c'|x)] \\ \mathcal{L}^f_{cls} &= \mathbb{E}_{x,c}[-\log D_{cls}(c|G(x,c))]\end{aligned} \tag{2}$$

- Pixel-wise reconstruction (cycle-consistency) loss

$$\mathcal{L}^{pix}_{rec} = \mathbb{E}_{x,c,c'}[\|x - G(G(x,c),c')\|_1] \tag{3}$$

- Full Objective function

$$\begin{aligned}\mathcal{L}_G &= \mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}^f_{cls} + \lambda_{rec}\mathcal{L}_{rec} \\ \mathcal{L}_D &= -\mathcal{L}_{adv} + \lambda_{cls}\mathcal{L}^r_{cls}\end{aligned} \tag{4}$$

3.2 Perceptual Loss in D

We apply reconstruction constraint from image space to feature space. We use pixel-wise reconstruction error (Eq. 3) between real images and reconstructed images before discriminator becomes robust. We replace feature-wise reconstruction error with pixel-wise reconstruction error

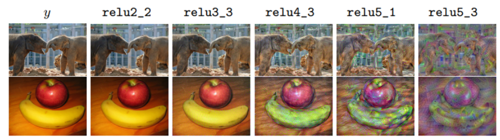


Figure 3: Experiments of perceptual loss applied at different layers in D from [4].

Layer	Input → Output Shape	Layer Information
Input Layer	$(h, w, 3) \rightarrow (\frac{h}{2}, \frac{w}{2}, 64)$	CONV-(N64, K4x4, S2, P1), Leaky ReLU
Hidden Layer	$(\frac{h}{2}, \frac{w}{2}, 64) \rightarrow (\frac{h}{4}, \frac{w}{4}, 128)$	CONV-(N128, K4x4, S2, P1), Leaky ReLU
Hidden Layer	$(\frac{h}{4}, \frac{w}{4}, 128) \rightarrow (\frac{h}{8}, \frac{w}{8}, 256)$	CONV-(N256, K4x4, S2, P1), Leaky ReLU
Hidden Layer	$(\frac{h}{8}, \frac{w}{8}, 256) \rightarrow (\frac{h}{16}, \frac{w}{16}, 512)$	CONV-(N512, K4x4, S2, P1), Leaky ReLU
Hidden Layer	$(\frac{h}{16}, \frac{w}{16}, 512) \rightarrow (\frac{h}{32}, \frac{w}{32}, 1024)$	CONV-(N1024, K4x4, S2, P1), Leaky ReLU
Hidden Layer	$(\frac{h}{32}, \frac{w}{32}, 1024) \rightarrow (\frac{h}{64}, \frac{w}{64}, 2048)$	CONV-(N2048, K4x4, S2, P1), Leaky ReLU
Output Layer (D_{src})	$(\frac{h}{64}, \frac{w}{64}, 2048) \rightarrow (\frac{h}{64}, \frac{w}{64}, 1)$	CONV-(N1, K3x3, S1, P1)
Output Layer (D_{cls})	$(\frac{h}{64}, \frac{w}{64}, 2048) \rightarrow (1, 1, n_d)$	CONV-(N(n_d), K$\frac{h}{64}$x$\frac{w}{64}$, S1, P0)

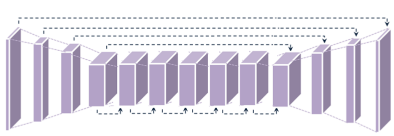
perceptual loss

Figure 4: Perceptual loss in D.

after 40 epochs. Refer to [4] (shown in Fig. 3), we extract low-level and mid-level features in the first 4 layers of discriminator to measure perceptual loss between real images and reconstructed images (shown in Fig. 4). We formulate equation as Eq. 5.

$$\mathcal{L}_{rec}^{feat} = \mathbb{E}_{x,c,c'}[\| \sum_l D_l(x) - \sum_l D_l(G(G(x,c),c')) \|_1] \tag{5}$$

3.3 Skip Connection in G



Part	Input → Output Shape	Layer Information
Down-sampling	$(h, w, 3 + n_c) \rightarrow (h, w, 64)$	CONV-(N64, K7x7, S1, P3), IN, ReLU
	$(h, w, 64) \rightarrow (\frac{h}{2}, \frac{w}{2}, 128)$	CONV-(N128, K4x4, S2, P1), IN, ReLU
	$(\frac{h}{2}, \frac{w}{2}, 128) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	CONV-(N256, K4x4, S2, P1), IN, ReLU
Bottleneck	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), IN, ReLU
Up-sampling	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{2}, \frac{w}{2}, 128)$	DECONV-(N128, K4x4, S2, P1), IN, ReLU
	$(\frac{h}{2}, \frac{w}{2}, 128) \rightarrow (h, w, 64)$	DECONV-(N64, K4x4, S2, P1), IN, ReLU
	$(h, w, 64) \rightarrow (h, w, 3)$	CONV-(N3, K7x7, S1, P3), Tanh

Figure 5: Skip connection between all down-sampling and up-sampling layers in G.

Original StarGAN use several residual blocks at bottleneck. Refer to [3], we also want to directly learn residual features between different emotions by applying skip connection between all down-sampling and up-sampling layers with the same output shapes (shown in Fig. 5).

3.4 Dropout + LReLu in G

Refer to ganhacks [5], we use Dropout layers (50%) in generator at both training and test phase to provide noise to inputs. In addition, we replace all ReLu layers in generator with LeakyReLu layers to avoid sparse gradients.

At first, we augment Dropout layers after the first and the last non-linear activation layers in generator for StarGAN with dropouts tricks version. We just try to replace the above positions of Dropout layers with only augmenting more Dropout layers at bottleneck in generator, e.g., layers in residual blocks after non-linear activations (similar to [6]), for StarGAN with skip connection and dropouts tricks version.

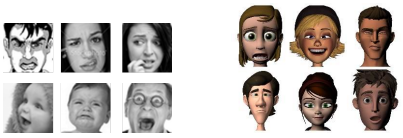


Figure 6: Examples of FER2013 (left) and FERG-DB (right) datasets.

4 Experiment

4.1 Datasets

We find two datasets, FER2013 [7] and FERG-DB [8], for our experiments (examples shown in Fig. 6). The images in the two datasets are grouped into 7 types of expressions (Angry, Disgust, Fear, Happy, Neutral, Sad and Surprise).

FER2013. Facial expression recognition 2013 challenge (FER2013) is a database of 48x48 grayscale real human pictures with annotated facial expressions. The database contains 28709 for training and 7178 for test.

FERG-DB. Facial Expression Research Group Database (FERG-DB) is a database of 256x256 RGB stylized virtual characters with annotated facial expressions. The database contains 55767 annotated face images of six stylized characters. The characters were modeled using the MAYA software and rendered out in 2D to create the facial expression images.

Table 1: Training data distribution for 7 types of facial expressions on FER2013 and FERG-DB.

Emotion Type	Ratio in FER2013 (%)	Ratio in FERG-DB (%)
Angry	13.92	16.44
Disgust	1.52	15.37
Fear	14.27	13.30
Happy	25.13	13.14
Neutral	17.29	12.44
Sad	16.82	13.68
Surprise	11.05	15.63

4.2 Experimental Results on FER2013

Qualitative results. Fig. 7 shows that some examples of results generated by different models. From Tab. 1, we know the data distribution on FER2013 training set is very unbalanced, especially the proportion of "Disgust" is less than 2%. Some of our generated results for "Happy" and "Surprise" look well. However, some of our generated results, especially for "Disgust" emotion, are unnatural. There are some and ghosting artifacts in generated images.

Training losses are shown in Fig. 8. We can observe that both feature-wise and pixel-wise reconstruction losses decrease to near 0 in our modified StarGAN with perceptual loss. Compare our modified version with the baseline (original StarGAN), classification loss converges earlier and most of the changes of losses over epochs are smaller than the original one.

Qualitative evaluation. We evaluate our results by facial expression recognition. We train a emotion recognition classifier by ResNet18 model on FER2013 training set, and evaluate generated images by different generators on test set by the same classifier. Training process information and confusion matrices of all models are shown in Fig. 9, and the performance of each model is reported in Tab. 2.

The classifier seems to overfit to samples of training set, since training loss is very low and training accuracy is near 100%. Since evaluation on real FER2013 test set is about 64%, this dataset is challenging, only high accuracy for "Happy" and "Surprise". It is hard to recognize facial expressions in original test images, especially for "Sad". We evaluate generative models by their generated

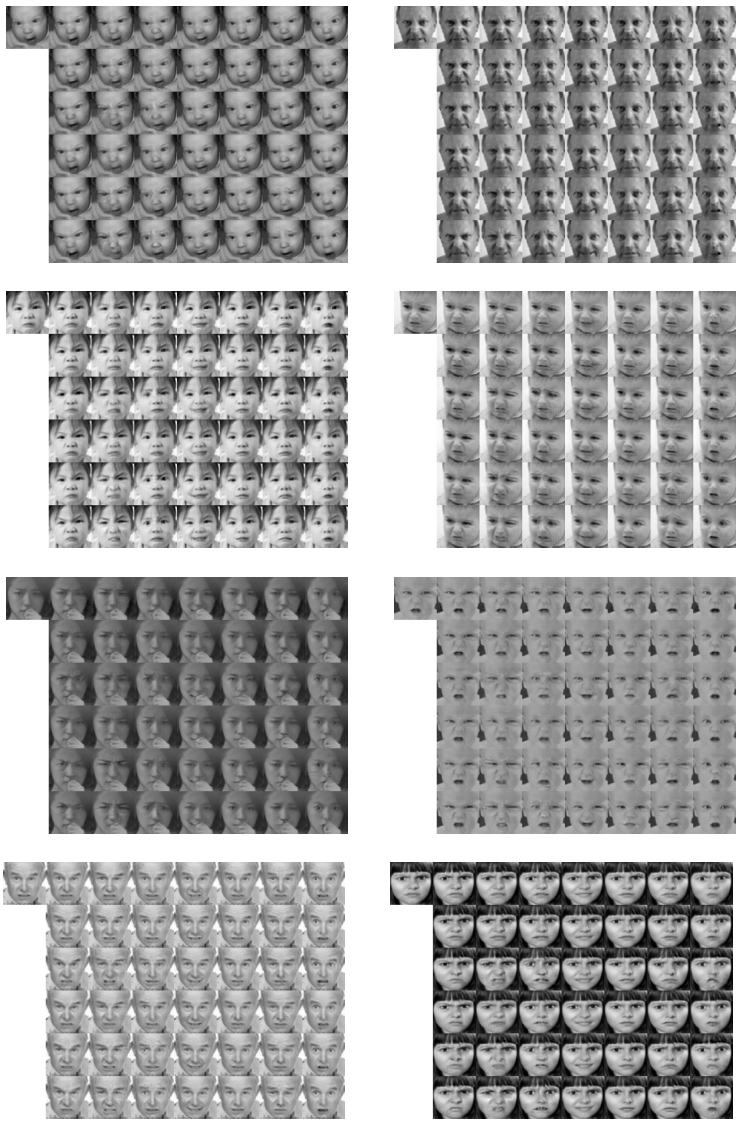
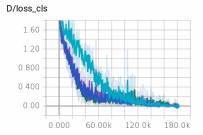


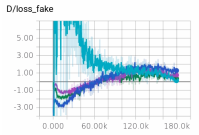
Figure 7: Generated images from FER2013 test set by different generators; from left to right: Input, Angry, Disgust, Fear, Happy, Neutral, Sad, Surprise; from top to bottom (refer to the order in Tab. 2): (1) original StarGAN, (2) StarGAN + perceptual loss, (3) StarGAN + perceptual loss + skip connection, (4) StarGAN + dropouts tricks, (5) StarGAN + skip connection, (6) StarGAN + skip connection + dropouts tricks.

images on test set for all 7 types of facial expressions. Therefore, the number of generated test images for each generator is 7x than the number of real images in original test set.

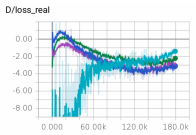
From Tab. 2, we can observe that the best generative model is StarGAN with skip connection and dropouts tricks in generator. StarGAN augmented with skip connection and dropouts tricks can significantly improve the performance in recognition evaluation. However, perceptual loss for low-level and mid-level features leads to degrade the performance in recognition evaluation, in spite of the fact that feature-wise and pixel-wise reconstruction losses decrease to near 0 in the versions of StarGAN augmented with perceptual loss. We also try feature matching in the official DiscoGAN implementation [9]. However, feature matching between real images and reconstructed images does not improve and not degrade the performance, it is ineffective. In the future, we may try other feature matching metrics to improve the poor performance in the aspect of feature-wise matching distribution.



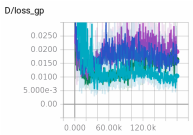
(a) Domain classification loss for D.



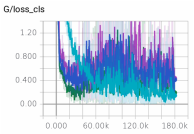
(b) Adversarial (fake) loss for D.



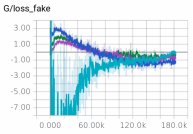
(c) Adversarial (real) loss for D.



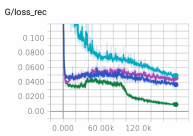
(d) Adversarial (gradient penalty) loss for D.



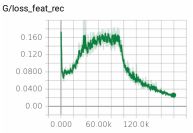
(e) Domain classification loss for G.



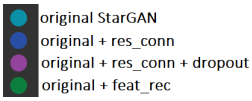
(f) Adversarial (fake) loss for G.



(g) Pixel-wise reconstruction loss for G.

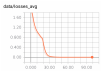


(h) Feature-wise reconstruction loss for G.

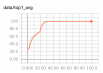


(i) Legend for each color.

Figure 8: Losses over epochs for training GAN on FER2013.

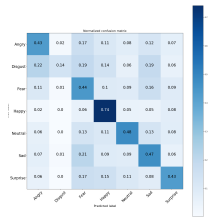


(a) Training loss over epochs.

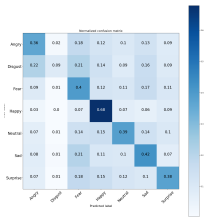


(b) Training accuracy over epochs.

(c) Original real images on test set.

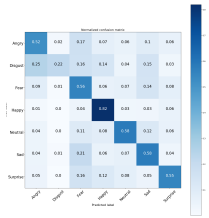


(d) Generated test images by original StarGAN.



(e) Generated test images by StarGAN with perceptual loss.

(f) Generated test images by StarGAN with perceptual loss and skip connection.



(g) Generated test images by StarGAN with dropouts tricks.

(h) Generated test images by StarGAN with skip connection.

(i) Generated test images by StarGAN with skip connection and dropouts tricks.

Figure 9: Information for training evaluation model (a-b) and confusion matrix on (generated) FER2013 test set (c-i).

4.3 Experimental Results on FER2013 + FERG-DB

Qualitative results. We also want to synthesize images with emotion transfer from real human to virtual character. We jointly train models on FER2013 and FERG-DB datasets by adding extra mask vector which represents images of which dataset to be generated. Since the emotion labels of

Table 2: Evaluation of generated images on FER2013 test set by facial expression recognition.

Input Images	Accuracy (%)
(c) Original real images on test set	64.433
(d) original StarGAN	44.729
(e) StarGAN + perceptual loss	38.823
(f) StarGAN + perceptual loss + skip connection	52.902
(g) StarGAN + dropouts tricks	54.759
(h) StarGAN + skip connection	57.091
(i) StarGAN + skip connection + dropouts tricks	**72.828**

the two datasets are the same, we just classify 7 types of facial expressions and real/fake images in discriminator, e.g., we do not classify the same emotion between the two datasets.

Fig. 10 shows that results generated by jointly training original StarGAN and our modified StarGAN with skip connection and dropouts tricks in generator. Jointly training losses are shown in Fig. 11, we can observe that in our modified version, classification loss converges earlier and most of the changes of losses over epochs are smaller than the original one. However, adversarial loss for discriminating real/fake in both original StarGAN and modified StarGAN is very high, which means that there is a gap between real images and generated images. Therefore, the generated results look not well, especially transfer from one dataset domain to the other dataset domain. The generated results seem to only learn the separate background color and foreground color information in the two datasets in original StarGAN, and it is hard to learn residual difference between different emotions with different dataset property (real human vs. virtual character) in our modified StarGAN. The reasons why the generated results look not well may come from the in two different kinds of datasets (real human vs. virtual character). In the future, we may need to separate generator and discriminator to get individual information, or to consider cross generation between the two datasets when training.

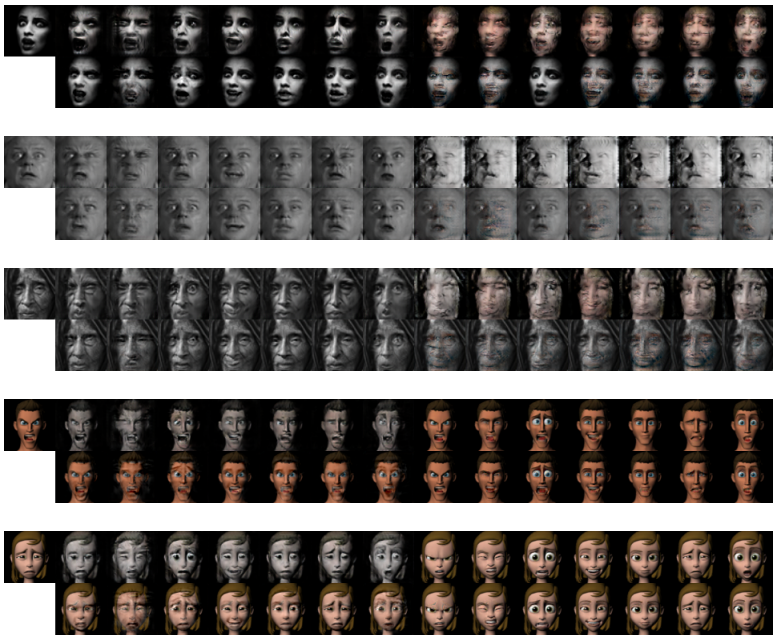
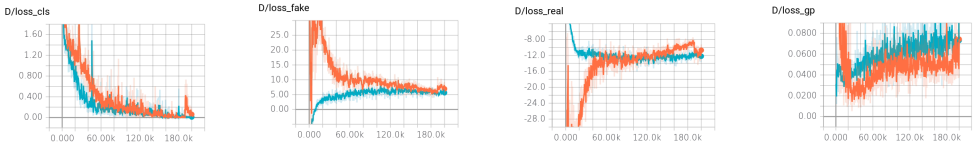


Figure 10: Generated images from FER2013 and FERG-DB by joint training different generators; from left to right: Input, FER2013_Angry, FER2013_Disgust, FER2013_Fear, FER2013_Happy, FER2013_Neutral, FER2013_Sad, FER2013_Surprise, FERG-DB_Angry, FERG-DB_Disgust, FERG-DB_Fear, FERG-DB_Happy, FERG-DB_Neutral, FERG-DB_Sad, FERG-DB_Surprise; from top to bottom: (1) original StarGAN, (2) StarGAN + skip connection + dropouts tricks.

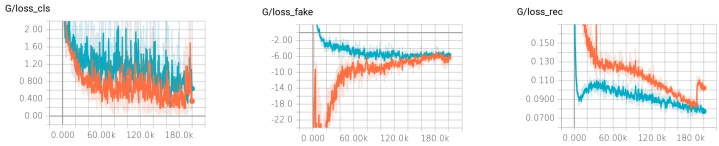


(a) Domain classification loss for D.

(b) Adversarial (fake) loss for D.

(c) Adversarial (real) loss for D.

(d) Adversarial (gradient penalty) loss for D.



(e) Domain classification loss for G.

(f) Adversarial (fake) loss for G.

(g) Pixel-wise reconstruction loss for G.

Figure 11: Losses over epochs for jointly training on FER2013 and FERG-DB (orange: original StarGAN, blue: our modified StarGAN).

5 Conclusion

We augment StarGAN with several techniques, including perceptual loss in discriminator, residual connection in generator, and some GAN tricks. We test experiments with the above techniques and most of the quality results by our method are better than the results generated by baseline. In addition, we evaluate facial expression recognition on baseline and our method, the performance of our method outperformed the baseline.

In the future, we will try to improve the quality results of generated images with different facial expressions from real human to virtual character. We may combine our method with XGAN [2] architecture and concepts to generate images of the other dataset domain and use constraint on certain semantic consistency. In addition, refer to Contrast-GAN [10], we may apply semantic-aware adversarial discriminator to learn semantic contrast between different emotions, and use masks in part of face to learn more specific region changes between different emotions.

References

[1] 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. *arXiv preprint arXiv:1711.09020*, 2017.

[2] Amlie Royer, Konstantinos Bousmalis, Stephan Gouws, Fred Bertsch, Inbar Mosseri, Forrester Cole, and Kevin Murphy. Xgan: Unsupervised image-to-image translation for many-to-many mappings. *arXiv preprint arXiv:1711.05139*, 2017.

[3] Chaoyue Wang, Chang Xu, Chaohui Wang, and Dacheng Tao. Perceptual adversarial networks for image-to-image transformation. *arXiv preprint arXiv:1706.09138*, 2017.

[4] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016.

[5] Soumith Chintala, Emily Denton, Martin Arjovsky, and Michael Mathieu. How to train a gan? In *Neural Information Processing Systems (NIPS)*, 2016.

[6] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[7] Ian Goodfellow, Dumitru Erhan, Pierre-Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, Yingbo Zhou, Chetan Ramaiah, Fangxiang Feng, Ruifan Li, Xiaojie Wang, Dimitris Athanasakis, John Shawe-Taylor, Maxim Milakov, John Park, Radu Ionescu, Marius Popescu, Cristian Grozea, James Bergstra, Jingjing Xie, Lukasz Romaszko, Bing Xu, Zhang Chuang, and Yoshua Bengio. Challenges in representation learning: A report on three machine learning contests, 2013.

[8] Deepali Aneja, Alex Colburn, Gary Faigin, Linda Shapiro, and Barbara Mones. Modeling stylized character expressions via deep learning. In *Asian Conference on Computer Vision*, pages 136–153. Springer, 2016.

[9] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1857–1865, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[10] Xiaodan Liang, Hao Zhang, and Eric P. Xing. Generative semantic manipulation with contrasting gan. In *Neural Information Processing Systems (NIPS)*, 2017.