CS 229 Project Final Report
A Method for Modifying Facial Features

Boning Zheng Meixian Zhu Jing Bo Yang
b7zheng mxzhu jingboy

(Dated: December 2018)

Category: Computer Vision

Many present day facial recognition systems focus on making verification or identification facial feature invariant. While these systems are highly effective for fully automated tasks, manual facial recognition is still required in numerous real life scenarios that involve comparing against ID photos. Our goal is to build a system that is capable of transforming faces to include or exclude glasses and beard. Such a system should be able to handle a wider range of facial features with small modifications. A few network structures have been tested for this purpose and we have found that CycleGAN[1] is the most capable compared to other vanilla GAN systems. Generated images from test set are presented and their inception scores [2] are analyzed. Details regarding characteristics of these generated images are also included in our discussion. Potential future improvements could involve making our system more generic or introducing semi-supervised learning to expand usable data sources. Source code for this project is available on Github.

I. INTRODUCTION

There have been significant improvement in our capability to identify and verify human faces over the past few years. Device makers are already taking advantage of such development by equipping their latest phones and tablets with AI co-processor and powerful image processing algorithms. However, the recent trend has mostly focused on making facial identification and verification invariant to facial features. These works certainly help machine recognize human faces, however, most humans are interested in seeing people in the natural state, without any facial disguise.

A system that can recover undisguised faces could be helpful for criminal investigation. In particular, witnesses should be able to make use of these processed images to identify the criminal among a series of ID photos, which typically include no disguise, or in person among a number of held suspects. People utilizing online dating apps could also utilize this system to reveal the real person behind facial disguise, a feature that many find useful.

We build on current work related to GAN-based style transform methods that are commonly employed for applying facial disguise. Recent works have demonstrated much success in related areas [3], [4]. Our method make use of similar machine learning techniques but aim to swap input and output of those algorithms to achieve our purpose.

We train our generative neural network using a facial disguise database from Hong Kong Polytechnic University [5] and CelebA [6]. We have experimented with increasingly more complex generative adversarial models and obtained images with expected improvement in quality. The best results were achieved using CycleGAN [1]. Inception scores [2] are incorporated into this project as a way to numerically evaluate quality of generated images.

II. RELATED WORK

The seminal paper on GANs was first published in 2014 [7]. Since then, GAN's have experienced wide success in rendering novel realistic images and image style transfer. The core of the framework is composed of two models, a *generator* and a *discriminator*. The generator (G) is trained to reproduce genuine target images from a specified input, while the discriminator (D) is trained to differentiate from generated images to naturally sampled images. The end goal is for the generator to produce increasingly realistic images of the target distribution, and for the discriminator to pick up on the most subtle differences between real and fake images. The training objective is expressed as a min-max problem through an adversarial loss function:

$$L_{GAN} = \mathop{\mathbb{E}}_{y\sim p_{data}(y)}[\log D_y(y)] + \mathop{\mathbb{E}}_{x\sim p_{data}(x)}[\log(1-D_y(G(x)))]$$

where G tries to minimize this function where an adversary tries to maximize it, creating the min-max optimization problem: $\min_G \max_{D_y} L_{GAN}$.

Another related work that builds on top of the traditional GAN is called the CycleGAN [1]. This work is more related to our project as it aligns with our goals of removing specific facial features. The CycleGAN presents a method of mapping an image X to an image Y (G: X→Y) without the requirement of paired samples. Furthermore, CycleGAN also learns an inverse mapping (F: Y→X) such that F(G(X)) ≈ X. This is done by adding in an "identity loss" function to the training process to preserve such identity between input and output. This type of architecture will be very useful towards our project since we would like to preserve the identity of the person while removing the facial features.

III. DATASET

A. Data Sources

Finding an appropriate dataset is one of the most important task for this work. Unfortunately, due to privacy concerns and inherent difficulties in obtaining ground truth associated with human faces [8], only the dataset obtained by Wang and Kumar [5] and the popular *CelebA* [6] dataset created by Liu, Wang and Tang are suitable for the task of detailed facial feature manipulation.

This project initially used Wang and Kumar's dataset because it contains nicely aligned and cropped facial images pre-processed into gray scale along with multiple annotations. This dataset consists of 2460 images of 410 different celebrities. All facial images are collected directly from the publicly available websites which are clearly cited in the database.This dataset provides the following ground truth attributes corresponding to human inspection of each of the images in the database:

File Name	File Size	Gender
Skin Color	Hat	Ethnicity
Hair Style	Glasses	Beard

TABLE I. Tags for HK Polytechnic dataset

In addition to the dataset provided by HK Polytechnic, we added the famous *CelebA* dataset as we became more confident with capability of our network. Like Wang and Kumar's dataset, *CelebA* also contains celebrity images "in-the-wild". This dataset contains over $200K$ images with various tags. We selected approximately $10K$ images that contain suitable tags (beard and glasses) for the project. Tags used for this project are included in TABLE II. Select images from both datasets are presented in Fig 1.

Male	Eyeglasses	Goatee
No_Beard	Mustache	

TABLE II. Useful tags for CelebA dataset

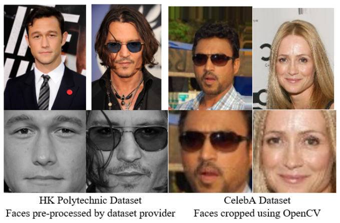
HK Polytechnic Dataset
Faces pre-processed by dataset provider
CelebA Dataset
Faces cropped using OpenCV

FIG. 1. Sample images from the two datasets

Out of the 10K images selected for our project, we divided them into training and testing datasets using a $8-2$ ratio. Since inception score should only be used as a reference for quality of image, there is little point in designating a validation set. We tuned parameters mostly based on manua inspection of generated images. Given the limited tool-sets in evaluating image quality, manual evaluate is the most appropriate for our purpose.

B. Pre-processing

We expect neural networks to produce better results when faces are intelligently selected from the images. Cropping out faces help the neural network select "area of interest" and reduces input size. With a reduced input size, the network can spend resources on applying feature transformations and on identifying features. For this project, we used image sizes of 64×64 to decrease demand on GPU memory.

Our dataset from Hong Kong Polytechnic comes with cropped images. CelebA, in contrast, contains too much background, for the dataset to be generic enough for a variety of tasks. We made use of OpenCV's Haar Cascades [9] to detect and crop out faces recognized from images. These crops, after rescaling, look almost identical to those provided by HK Polytechnic. We did not manually filter out poor quality crops because it is too time consuming. Low percentage of poorly cropped images should have little effect on training.

IV. METHODS

A. Vanilla GAN Structure

Neural network used for our purpose are much more sophisticated than typical generative adversarial networks that deals with MNIST datasets. Multiple network structures have been attempted and their differences will be presented in the *Results* section. Given the theoretical background of Generative Adversarial Neural Network, as discussed in Section II, a vanila GAN can be roughly represented by Fig 2.

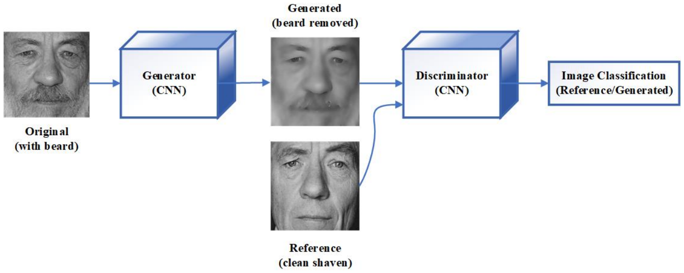

FIG. 2. Vanila GAN Model for beard removal

a. Perceptron-Based A multi-layer perceptrons network (shown in Table III) is built to test whether the infrastructure is reliable.

Generator	**Discriminator**
Input 128 × 128 Gray Scale	Input 128 × 128 Gray Scale
3×1024 Neuron Leaky ReLu	3× 512 Neuron Leaky ReLu
512 Neuron Leaky ReLu	
Output 128 × 128 Gray Scale	Output $[y_1, y_2]$ One-hot encoding

TABLE III. Structure of multilayer perceptron

b. Simple Convolutional Neural Networks Simple multilayer convolutional networks are built for generator and discriminator. For example, one such model that we have built is shown in Table IV.

Generator	**Discriminator**
Input 128 × 128 Gray Scale	Input 128 × 128 Gray Scale
5×CNN with 3 × 3, 128 features	CNN with 3 × 3, 256 features
CNN with 3 × 3, 64 features	CNN with 3 × 3, 256 features
CNN with 3 × 3, 1 feature	CNN with 3 × 3, 128 features
	256 Neuron Leaky ReLu
Output 128 × 128 Gray Scale	Output $[y_1, y_2]$ One-hot encoding

TABLE IV. Structure of simple CNN

c. Residual Convolutional Neural Networks Residual networks [10] was first introduced by Szegedy. Neural networks with residual structure do much better in retaining original images. Wang [11] has implemented a similar structure for generating high resolution images. Our ResNet structure is presented in TABEL V.

Generator
Input 128 × 128 Gray Scale
CNN with 3 × 3, 128 features
CNN with 3 × 3, 128 features
Residual 2-Layer CNN with 3 × 3, 128 features
Residual 2-Layer CNN with 3 × 3, 128 features
Residual 2-Layer CNN with 3 × 3, 128 features
CNN with 1 × 1, 1 feature
Output 128 × 128 Gray Scale

Discriminator
Input 128 × 128 Gray Scale
CNN with 3 × 3, 256 features
CNN with 3 × 3, 256 features
CNN with 3 × 3, 128 features
256 Neuron Leaky ReLu
Output $[y_1, y_2]$ One-hot encoding

TABLE V. Structure of Residual-CNN

B. CycleGAN

The *Project Milestone* has demonstrated that vanila GANs have limited capability in terms of both beard removal and facial feature reconstruction. This, as described in Section II, can be tackled by introducing coupling, implemented as a CycleGAN-like structure shown in Fig 3. Note that the figure presented is only half of CycleGAN. This half demonstrates how the desired "forward" image is generated. "Reconstruction" is achieved by training the backward generator. Clearly, the other half of this network helps the forward generator maintain image fidelity.

The forward generator G maps disguised faces to original faces, whereas the backward generator F maps original faces back to disguised faces. We apply adversarial loss functions to both GAN's:

$$\begin{aligned} L_{GAN}(G, D_Y, X, Y) = &\mathop{\mathbb{E}}_{y\sim p_{data}(y)}[\log D_y(y)] \\ &+ \mathop{\mathbb{E}}_{x\sim p_{data}(x)}[\log(1 - D_y(G(x)))] \end{aligned}$$

In addition to the adversarial loss functions, we have an additional cycle-consistency loss to preserve the individual identities through the generation process:

$$\begin{aligned} L_{cyc}(G, F) = &\mathop{\mathbb{E}}_{y\sim p_{data}(x)}[||F(G(x)) - x||_2] \\ &+ \mathop{\mathbb{E}}_{y\sim p_{data}(y)}[||F(G(y)) - y||_2] \end{aligned}$$

Such that our full objective would be:

$$\begin{aligned} L(G, F, D_X, D_Y) = &L_{GAN}(G, D_Y, X, Y) \\ &+ L_{GAN}(F, D_X, Y, X) \\ &+ L_{cyc}(G, F) \end{aligned}$$

where λ is a hyperparameter that controls the relative importance of the two objectiv losses.

We tested CycleGAN using relatively simple network structure. However, the CycleGAN structure has two sets of generator-discriminator pairing, effectively doubling the size of the network. Structure of both pairs are the same, as presented in Table VI. As expected, this complex networks takes considerable amount of time to train, but it certainly does excel in preserving irrelevant facial features.

2 × Generator	**2 × Discriminator**
Input 64 × 64 Gray/Colored	Input 64 × 64 Gray/Colored
CNN with 3 × 3, 128 features	CNN with 3 × 3, 256 features
CNN with 3 × 3, 64 features	CNN with 3 × 3, 256 features
	Flattening
Output 128 × 128 Gray/Color	Output $[y_1, y_2]$ One-hot encoding

TABLE VI. Structure of CycleGAN with "simple" CNN layers

C. Inception Score

As an evaluation metric, we calculate the inception scores based on the inception model derived in [2] by Salimans. Every generated image has a conditional label distribution $p(y|x)$ based on the inception model. Images that contain meaningful objects should have $p(y|x)$ with

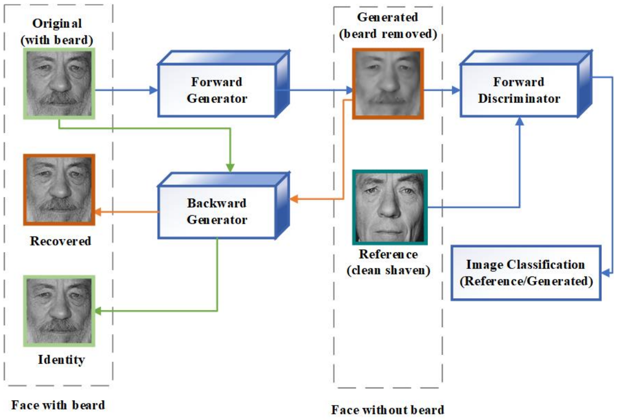

FIG. 3. Cycle GAN Model for beard removal

low entropy. Moreover, we expect the model to generate varied images, so the marginal $\int p(y|x = G(z))dz$ should have high entropy. Since these two criteria are related to the same entity, the inception score is defined as the exponential of the KL distance between their respective distributions:

$$exp(\underset{x}{\mathbb{E}}[KL(p(y|x)||p(y))])$$

Taking the exponential makes it easier for us to compare the values.

V. EXPERIMENTS

A. Experiment Environment

Our experiments are conducted on Google Cloud VM instances with NVIDIA K80 GPUs. This setup significantly speeds up the training process compared to running on CPU-only machines, decreasing discriminator training time from over an hour to less than a minute and generator training from 10 to 15 minutes per batch on a simple CNN structure to a few seconds.

We built the training infrastructure using *Keras*. In addition, we have developed a generic infrastructure that is capable of handling difference generators and discriminators in a plug-and-go fashion. This modular infrastructure has significantly lowered overhead associated with experimenting with a wide range of network structures. Our custom code referenced vanilla GAN implementation from [12], CycleGAN implementation from [12] and [13], and inception score from [14]. Source files use for this project are available on Github.

B. Results and Discussion

We present generated images from different networks that we have experimented with.

a. Multilayer Perceptron As shown in FIG 4, generated images have rather poor quality. This is because multilayer perceptrons cannot capture spatial relationships. Nevertheless, this demonstrates that the generator loss function should be correct, as it is producing images that look like human faces.

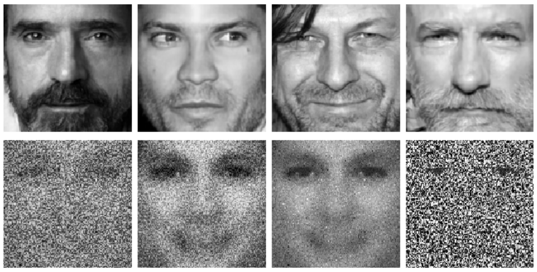

FIG. 4. Generated images from multilayer perceptron

b. Simple Convolutional Neural Network As shown in FIG 5, generated images are a lot smoother than that from multilayer perceptron. There is also “‘traces” of beard/mustache region being modified by the generator network. Also the generator seems to be brightening columns near nose, where mustache typically appears.

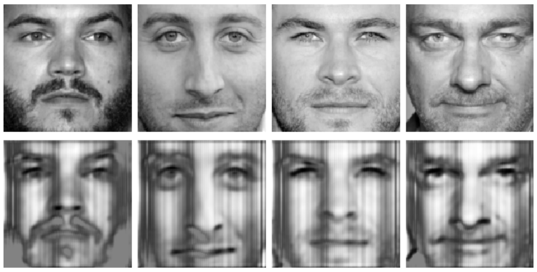

FIG. 5. Generated images from simple CNN

c. Residual Convolutional Neural Network Residual networks are supposed to be better in retaining characteristics of the original image. Since this network also contains more convolutional layers, the result, shown in FIG 6, has slightly higher quality than images generated using simple CNNs. These images have far less bright/dark “bars”.

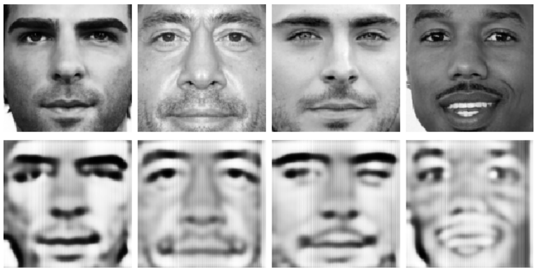

FIG. 6. Generated images from residual CNN

d. CycleGAN A relatively simple CycleGAN structure is implemented for this work. This is because CycleGAN consumes more than twice the memory compared to its vanilla counterparts. Expanding our network to support colored images also significantly limits complexity of the network. Nevertheless, CycleGAN produces high quality images, as shown in FIG 7.

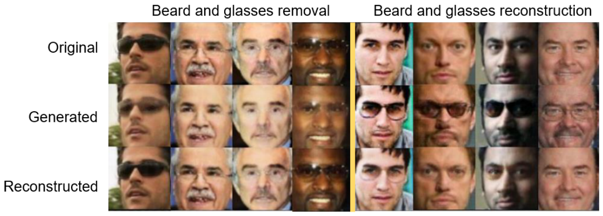

FIG. 7. Generated test images from CycleGAN

Clearly, with the introduction of reconstruction and identity loss, the generated images are of much higher quality. Not only that irrelevant features are modified, our reconstructed images look almost identical to the original, verifying that the reconstruction losses are highly effective.

Plot of losses for CycleGAN running the beard and glasses modification task is presented in FIG 8. It is conceivable that the generator losses plateau after a few hundred iterations, while overall network loss continue to decline. The overall network loss is weighted, accounting for discriminator accuracy and quality of generated images.

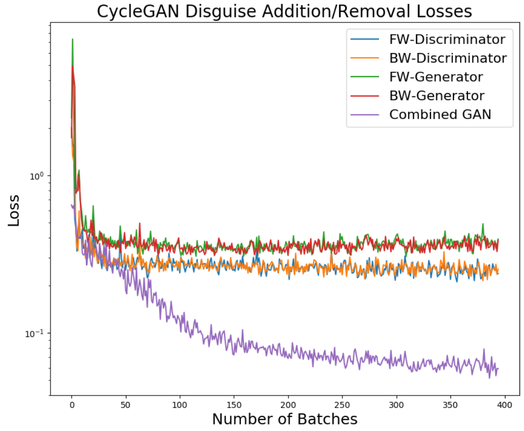

FIG. 8. Model losses of CycleGAN

All images presented here are faces of male. This is because training the network with female faces introduces makeup to modified faces. For example, removing beard adds lipstick regardless of gender. Similarly, removing sunglasses frequently adds eyeshadow or eyeline. Another interesting phenomenon we notices is that old celebrities tend to get clear glasses whereas younger celebrities tend to get sunglasses.

	CycleGAN	ResCNN	CNN	Perceptron
Mean	2.25	1.38	1.02	1.21
Variance	0.20	0.23	0.025	0.29

TABLE VII. Inception score of various models

Though the network handles most images reasonably well, we have noticed that it is still struggling with removing opaque sunglasses. This difficulty is expected because image with opaque sunglasses provides little information about wearers' eyes. The algorithm has nothing to construct the eyes from. Instead, it puts a "generic" eye in place of sunglasses, which often look out of place. This effect is observed among images in which glasses hide significant portion of eye brows. Reconstructed eye brows in those cases are of dubious quality.

Since this project is generative in nature, there is no accuracy to evaluate. Inception score is perhaps the more appropriate numerical metric to include for the experiment. Inception score of all tested networks are presented in TABLE VII. Since inception score for CIFAR-10 images [15] are only around 2.15 [2], images generated by our CycleGAN are in fact, of decent quality. The other three networks, as expected, have much lower inception score. Score of these three are not exactly the same as how human would rank their image qualities, which in a way verifies that inception score should not be the only method to quantify image quality.

VI. CONCLUSION AND FUTURE WORKS

This project successfully identified a neural network structure to perform the task of modifying facial features. Although results of this work focuses exclusively on beard and glasses, the same infrastructure can certainly be used for other features.

In the future, we would like to build a generic infrastructure that is capable of handling any facial feature. It would also be helpful to make the training process semi-supervised. This will allow us to include other datasets that do not have relevant tags. Mirza's [16] work on conditional GAN is high relevant if we were to moev toward this direction. We can also experiment with other bi-directioned GANs or certain autoencoder models like those created by Makhzani [17]. These models have been shown to perform reasonably well for similar tasks.

VII. DISTRIBUTION OF WORK

Our team has divided work evenly based on each team member's technical background and course load. To be more specific, Jingbo worked on pre-processing and testing neural network models, Boning worked on building various neural network models, and Meixian focused on plotting and writing reports/poster.

[1] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *arXiv preprint*, 2017.

[2] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, pp. 2234–2242, 2016.

[3] H. Chang, J. Lu, F. Yu, and A. Finkelstein, "Pairedcyclegan: Asymmetric style transfer for applying and removing makeup," in *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[4] H. Dong, P. Neekhara, C. Wu, and Y. Guo, "Unsupervised image-to-image translation with generative adversarial networks," *arXiv preprint arXiv:1701.02676*, 2017.

[5] T. Y. Wang and A. Kumar, "Recognizing human faces under disguise and makeup," in *Identity, Security and Behavior Analysis (ISBA), 2016 IEEE International Conference on*, pp. 1–7, IEEE, 2016.

[6] 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.

[7] 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.

[8] T. Mitchell, "Cmu face images data set." https://archive.ics.uci.edu/ml/datasets/cmu+face+images, 1997. [Online; accessed December-2018].

[9] A. Reimondo, "Haar cascades," *OpenCV Swiki*, 2008.

[10] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning.," in *AAAI*, vol. 4, p. 12, 2017.

[11] M. Wang, H. Li, and F. Li, "Generative adversarial network based on resnet for conditional image restoration," *arXiv preprint arXiv:1707.04881*, 2017.

[12] eriklindernoren, "Keras implementations of generative adversarial networks." https://github.com/eriklindernoren/Keras-GAN. [Online; accessed November-2018].

[13] tjwei, "wgan, wgan2(improved, gp), infogan, and dcgan implementation in lasagne, keras, pytorch." https://github.com/tjwei/GANotebooks/. [Online; accessed December-2018].

[14] nnUyi, "Inception-score." https://github.com/nnUyi/Inception-Score, 2017. [Online; accessed December-2018].

[15] A. Krizhevsky, V. Nair, and G. Hinton, "The cifar-10 dataset," *online: http://www. cs. toronto. edu/kriz/cifar. html*, 2014.

[16] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[17] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.