FacePaint: An Exploration of Localized Transfer on Facial Expressions

Sasha Harrison
Stanford University
aharris6@cs.stanford.edu

Frits van Paasschen
Stanford University
fritsvp@cs.stanford.edu

December 12, 2019

1 Abstract

Is it possible to change a person's facial expression? In this paper, we apply Neural Style Transfer, Image Segmentation, and Generative Adversarial Networks (GANs) to a new application; namely, changing the expression on a human face. Because the human eye is particularly sensitive to distortions of human facial features, accomplishing this goal will require precise and detailed results. We present qualitative results from various architectures, and present the ones that show the most promise with respect to this supervised task. In our experiments, we found that Cycle-GANs show the most promise in this application area. Overall, we present an end-to-end neural framework for realistic expression modification on human faces.

2 Introduction

Neural style transfer can be used to transfer the texture of one image onto the content of another, often yielding psychedelic, otherworldly results [4]. Moreover, the concept of Deepfakes [17], or the generation of synthetic images using neural networks, has also recently grown in popularity. In 2018, a group of NVIDIA researchers used an architecture they called a StyleGAN [10] to produce a high resolution human face that is visually indistinguishable from a photograph.

We seek to experiment with applying neural techniques to a new domain of expression transfer. Our first experimental approach leverages Neural Style Transfer (NST) [4] in concert with image segmentation in the form of Mask R-CNN [6]. The input to this model are two images. The first, (a) contains the subject, and (b) contains the target expression. We use segmentation to localize the faces in both images, and then use NST to superimpose facial 'style' from (b) onto image (a).

Following the sub-optimal results of our first experiment, we improve our methodology using two different types of Generative Adversarial Networks [5] to manipulate facial expression. With several varieties of the GAN architecture [20] [7], we attempt to learn a transformation between different facial expression domains.

3 Related Work

Our approach is related to many deep learning pipelines previously published by computer vision researchers.

Over the past several years, Generative Adversarial Networks (GANs), originally suggested by Goodfellow et al, have become the state of the art method for image manipulation tasks [5]. Under this framework, a mini-max game between a discriminator D and a generator G models a data distribution by minimizing the Jensen-Shannon distance between real and fake data. In practice, however, the difficulty of training a vanilla GAN has led to improvements in optimization and architectural design which, in turn, improve the stability and performance of this type of model. Thus, as a first attempt, we incorporate these improvements by implementing the Wasserstein distance metric [1] for a Self-Attention GAN [20]. In essence, these two architectural choices address common GAN issues such as mode collapse and receptive field size in convolutional Generators.

Another promising GAN-based approach to our chosen task is the Cycle-GAN [21] [7], an architecture that leverages cycle-consistency to learn cyclical transformations between data domains. In this paper, authors Zhu et al. showed that GANs can yield high quality results on image-to-image translation. Specifically, given an image dataset broken into discrete categories, it is possible to use GANs to translate an image from Domain X to Domain Y by transforming the distribution of $G(X)$ to approximate the distribution of Y. One large benefit of the Cycle-GAN is that it avoids mode

collapse by introducing additional constraints on the objective function of the network. The vanilla GAN objective yields a distribution $\hat{y}$ that models the empirical distribution $p_{data}(y)$. But there are infinitely many mappings G that produce such a $\hat{y}$. The cycle-GAN objective exploits the property that translation should be "cycle consistent"; If we have two mappings $G : X \rightarrow Y$, and $F : Y \rightarrow X$ to be a This problem formulation is highly relevant to the application we explore in this paper because it is natural to view emotions as representing different categories of images. Thus, a large advantage of this approach is that it can be applied to our subject area without any major modifications.

Lastly, a recent publication titled *A Style-Based Generator Architecture for Generative Adversarial Networks* [10] introduces the idea of a style-based generator. The main idea is that by borrowing from Style Transfer literature, this Generator frameowrk is able to separate of high-level attributes like pose and identity from stochastic variation in the generated images (e.g., freckles, hair). Thus, it enables intuitive, scale-specific control of the synthesis which in turn yields incredibly detailed and realistic results. While this methodology falls outside the scope of our work, it is important to note that the StyleGAN architecture shows the natural relationship between style transfer and deepfakes, and is responsible for the state-of-the-art results for synthetically generated faces [10].

4 Methods

Dataset

We used a variety of datasets for different components of our project. For Neural Style Transfer, we used a pretrained VGG19 CNN [16] trained on the ImageNet Dataset [2]. For the Mask R-CNN segmentation model, we trained a model on the WIDER-Face dataset [19]. Finally, for inference with the segmentation and NST pipeline, as well as for training the SA-GAN model, we used the JAFFE (Japanese Female Facial Expressions) dataset [12]. The JAFFE dataset consists of 213 images of 10 distinct Japanese women[12]. Each subject in the JAFFE dataset makes six facial expressions (anger, disgust, fear, happiness, sadness, and surprise), which in the early 20th century, were determined to be recognized the same way across cultures [11]. Examples of images that are part of JAFFE can be seen in the Figure 1.

Neural Style Transfer and Image Segmentation

The goal of NST is to combine the content and style of two arbitrary images using neural models. The key finding of Gatys et. al is that style and content representations in a given image are somewhat distinct. Given a content image, we can capture a *content representation* of objects and their placement within the image [4] using the higher level layers of a CNN trained for object detection, such as a pre-trained VGG19. Next, given a style image, we can obtain a *style representation* using the gram matrix, which computes the correlation over different response filters. The intuition behind the gram matrix is that it captures texture information, but not the global arrangement of objects within the style image.

The NST architecture is trained by minimizing both the style loss and the content loss, which are given by:

$$\mathcal{L}_{content}(p, x, l) = \frac{1}{2} \sum_{i,j} (F_{ij}^l - P_{ij}^l)^2$$

Where p is the original image, x is the output image, and P^l and F^l denote their feature representation in layer l.

The style loss for one layer is given by:

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^l - A_{ij}^l)^2$$

Where a and x are the style image and the output image, and A^l and G^l their respective style representations in layer l.

So the total style loss is given by:

$$\mathcal{L}_{style}(a, x) = \sum_{l=0}^{L} w_l E_l$$

where w_l are the weighting factors of the contribution of each layer to the total loss.

$$L_{total}(p, a, x) = \alpha \mathcal{L}_{content}(p, x) + \beta \mathcal{L}_{style}(a, x)$$

By minimizing L_{total}, an NST architecture alters the output image to transfer the style of a onto the content of p.

For the task of image segmentation, we use a Mask R-CNN [6], which performs pixel-level segmentation to localize objects within an image. The Mask R-CNN architecture extends faster R-CNN architecture to simultaneously perform object detection and generates a high-quality segmentation mask. We used this Mask R-CNN, trained on WIDER-Face, to localize human faces as potential inputs for our NST pipeline.

Generative Adversarial Networks

As mentioned previously, GANs (Generative Adversarial Networks) are considered to be among state of the art methodology for image generation. We tried two different varieties of GAN: (1) a Wasserstein Self-Attention GAN and (2) A Cycle-GAN.

For the first experiment, we will use a custom class-conditional GAN architecture that relies on the Wasserstein distance objective and a self-attention mechanism [20] [1]. The self-attention mechanism uses 1x1 convolutions over each mid-network activation to widen the effective receptive field of the Generator and to allow the Generator to pay attention to spatially-dependent activations.

At several points within both D and G, an attention map o_j is computed over the activations. We used a new, modified supervised softmax cross entropy (CE) loss based on one of our previous projects [18]:

$$L_D = -E_{(x,y)\sim p_{\text{data}}}[CE(X)] - E_{(x,y)\sim p_{\text{data}}}[CE(D(G(X)))]$$

$$L_G = -E_{(x,y)\sim p_{\text{data}}}[CE(y, D(G(X)))]$$

In essence, we use the discriminator to predict a multinomial distribution of class labels, and penalize the discriminator for incorrect guesses, and the generator for causing incorrect guesses by the discriminator.

The advantage of a Wasserstein Self-Attention GAN is that, unlike vanilla GANs, it does not require maintaining a careful balance in training of the discriminator and the generator. It also reduces the mode-collapse phenomenon that is typical in GANs producing several classes of images, which applies to our problem formulation as emotion is a multi-class distribution. The architecture of our final model is shown in Figure 2.

The next variety of GAN we tried was a Cycle-GAN [7] [21]. For the mapping function $G : X \rightarrow Y$ and its discriminator D_Y, the objective we used was

$$\mathcal{L}_{GAN}(G, D_Y, X, Y) = E_{y\ p_{data}(y)}[\log D_Y(y)] +$$

$$E_{x\sim p_{data}(x)}[\log(1 - D_Y(G(x)))]$$

Where cycle consistency loss is given by:

$$\mathcal{L}_{cyc}(G, F) = E_{x\sim p_{data}(x)}[||F(G(x)) - x||_1]$$

$$+E_{y\sim p_{data}(y)}[||G(F(y)) - y||_1]$$

5 Experiments and Results

Neural Style Transfer + Segmentation

For the project milestone, we implemented neural style transfer based on Gatys et. al. using the PyTorch [15] library. We drew inspiration from [8], and used a pretrained VGG19 model for initial weights, then performed transfer learning with a custom loss function composed of Content and Style Loss [8]. To perform style transfer, we initialized the input image as a tensor, so that instead of updating the model weights using gradient descent (as is standard in object detection), we would instead alter the image to minimize content and style loss. The results are shown in Figure 3.

Since the two face images originate from the same dataset, they are quite similar with respect to texture or "style." As a result, the output image differs very little from the original content image. The result from Figure 3 shows that NST will not pick up facial expression as "style" with no alterations. As such, we require a different method for transferring facial expression across images that does not solely rely on NST.

As a proof of concept, we trained a Mask R-CNN on WIDER-Face [19] to localize human faces in source images. These masked images could eventually be used as inputs to our final pipeline, but for the scope of this project this aspect of the pipeline was abandoned in favor of modifying faces in a structurally sound way using GANs.

Generative Adversarial Networks

SA-GAN

In our first experiment, we used a Conditional GAN with Self-Attention to learn a transformation between arbitrary expressions. We experimented both with the Wasserstein distance and our custom loss metric defined above for optimization, and found that the custom softmax loss allowed for some initial convergence and expression modification learning. The SA-GAN with the Wasserstein distance did not fully converge and the results were not meaningful. In order to conduct the SA-GAN experiment with the modified cross-entropy loss, we concatenated images with randomly selected class labels before inputting them into the generator. These randomly selected class labels served as the target class for the generated image. We attempted to train the generator to produce the randomly chosen expression, conditioned on the input image. The discriminator was then trained to identify correct classes from both real data and data that was generated. Re-

sults, showing frequent mode collapse but some initial progress, from this experiment are shown in Figure 4.

One limitation of the SA-GAN experiment was that we used a training dataset consisting of only 213 images, which is pretty small for a model with such a large number of parameters. We expect we would achieve better results repeating this experiment on a larger dataset. In addition, the Softmax Cross Entropy Loss does not have properties that ensure it avoids mode collapse. Therefore, we chose to continue experiments with the Cycle-GAN, a more constrained framework.

CycleGAN

The second GAN architecture we implemented was a Cycle-GAN, originally proposed by Berkeley researchers Zhu et. al. in 2017 [21]. Cycle-GANs are formulated so as to use two generative networks to convert images between two classes while enforcing "cycle consistency," meaning that these two generator functions be inverses of one another. As input to this model, we formulated a custom dataset with two classes: Happy (A) and Neutral (B). Thus, the goal of this algorithm is to convert a happy face to a neutral expression and vice versa. For our implementation, we drew inspiration from the GitHub repository published by [21]. This dataset consisted of 446 training images split amongst the two classes fro a mixture of the JAFFE dataset and the FEI dataset. We trained the model for 350 iterations with an Adam Optimizer, and a learning rate of 0.0005 that decreased linearly to zero over the iterations.

The results from this model are qualitatively strong. At about 200 iterations, the model began to localize its changes to the area around the mouth. This is a logical result, since this is the area of maximal variance between the two classes present in the training set. Notably, the reconstruction loss (difference between original image X and F(G(X))) decreases and becomes more stable over training, as shown in Figure 7. Our most successful training experiment yielded an L1 distance of 134.59 between images in a hold-out test set images and their reconstructions passed through the two generators (F(G(X))). Qualitatively, the reconstructed images are visually indistinguishable from the originals, as one can see in Figure 5.

One drawback of these experiments is the lack of quantitative metrics for evaluating the quality of generated images. We hoped to use the inception score as an additional quantitative metric, but because of the limited size of our dataset (446 training images total) the score would have too high of a variance to be meaningful.

6 Conclusion + Future Work

Overall, we performed a variety of experiments with a host of different architectures and techniques to achieve our goal of expression transfer on human faces. Ultimately, the Cycle-GAN architecture showed the most promise, by generating well-defined macro and micro-level features when transforming facial expression between domains. We are quite satisfied with this initial result, and believe it can be easily extended with more functionality to accomplish a variety of tasks revolving around neural facial transformation.

As for future work, we would like to combine our results into a final pipeline that combines image segmentation with expression transfer, for valid transformations of facial expression within a larger scene. In addition, we could revisit the SA-GAN's loss ojective and data set to attempt to get more refined results. We would also like to work with color images of faces, and we would like to make a more general Cycle-GAN for transformation between multiple classes of expression.

7 Contributions

Both partners contributed equally to this project. For the milestone, Sasha ran and implemented the Neural Style Transfer, while Frits implemented segmentation using Mask R-CNN. Frits took the lead on implementing the SA-GAN with custom loss function and ran the associated experiments, which played to his strengths given his previous experience with generative models. Sasha implemented the Cycle-GAN and ran the associated experiments, which included formulating a custom dataset mixing images from JAFFE and FEI datasets. The two partners collaborated on the poster and final report, as well as troubleshooting Google Cloud Platform, which proved to be a frequent pain point.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[3] f1ashine. Face detection base on mask r-cnn, 2017.

[4] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style, 2015.

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

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[8] Alexis Jacq and Winston Herring. Neural transfer using pytorch.

[9] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2018.

[10] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[11] Shan Li and Weihong Deng. Deep facial expression recognition: A survey. *CoRR*, abs/1804.08348, 2018.

[12] Michael Lyons, Shigeru Akamatsu, Miyuki Kamachi, and Jiro Gyoba. Coding facial expressions with gabor wavelets. *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, April*, 1998:200 – 205, 05 1998.

[13] Michael J Lyons, Shigeru Akamatsu, Miyuki Kamachi, Jiro Gyoba, and Julien Budynek. The japanese female facial expression (jaffe) database. In *Proceedings of third international conference on automatic face and gesture recognition*, pages 14–16, 1998.

[14] Ali Mollahosseini, Behzad Hasani, and Mohammad H. Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 10(1):18–31, Jan 2019.

[15] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[17] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016.

[18] Frits van Paasschen and Yousef Hindy. Self attention generative adversarial networks for high-dimensional scene representations from single 2d images. In *CS231n Class Project*, 2018.

[19] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[20] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

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

(a) Happy (b) Disgust (c) Surprise

Figure 1: Example from JAFFE Dataset

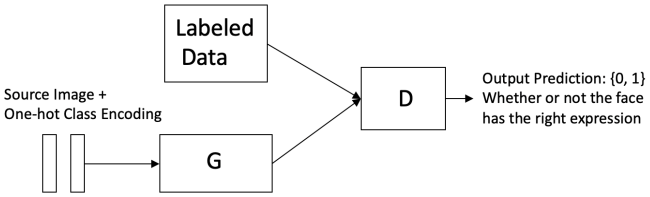


Figure 2: SA-GAN Architecture

Figure 3: Example NST Output (two images)

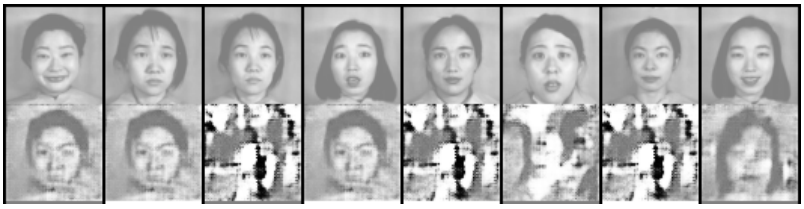
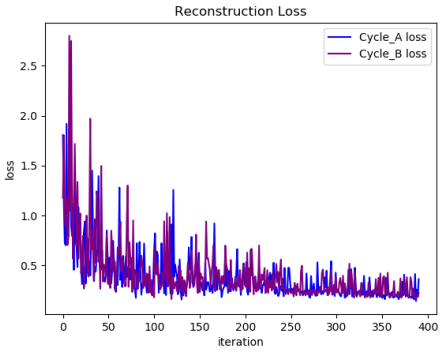


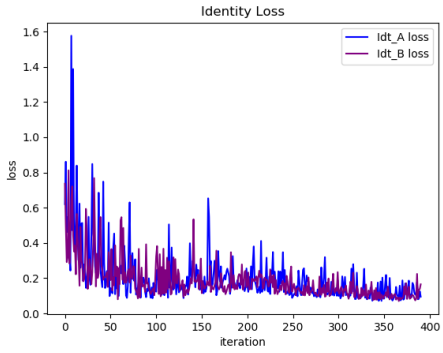
Figure 4: SA-GAN Output after training with CE Loss

(a) Orig. Image (b) Generated Image (c) Reconstructed Image (d) Orig. Image (e) Generated Image

Figure 5: Example output of Cycle-GAN



(a) $|G_a(G_b(A)) - A|$



(b) $|G_a(A) - A|$

Figure 6: Training Metrics for Cycle-GAN