CS230

Gender Classifier and CycleGAN for Altering Facial Images

Srinivas Halembar
halembar@stanford.edu

Mark Cramer
mdcramer@stanford.edu

Harry Jiang
xinxuan@stanford.edu

Abstract

We developed a gender classifier using transfer learning with a pre-trained VGG-16 model that achieved results comparable to state-of-the-art. We also employed a CycleGAN to address the issue of gender conversion. We applied hyperparameter tuning, fine tuning and transfer learning to improve the accuracy of our models. We present the gender classifier performance along with sample images from our gender conversion using a CycleGAN.

1 Introduction

Our initial motivation was to determine what sorts of personality characteristics can be predicted from a facial image (using a CNN classifier) and then, in a second step, manipulate that image by adjusting the levels of the predicted characteristics (using an Invertible Conditional GAN).

People's ability to assess tribal, gender and emotional properties by examining faces has been, for the vast majority of human existence, a biological imperative [1]. Rightly or wrongly, many believe they can additionally identify more subtle personality traits, such as sexuality, spirituality, intellect, political inclination or any of the Myers-Briggs or Big 5 personality traits. While there is significant scientific research demonstrating this might be possible [2], Deep Learning offers an opportunity to test those theories through classification and prediction. Being able to manipulate facial images then offers an entertaining and perhaps instructive example of how someone might look with an altered set of personalities or other characteristics.

After an offer to make available an extensive data set of facial images labeled with personality information was reneged, subsequent to the milestone, we were fortunate enough to be granted a different set of facial images, although labeled with gender and age as opposed to personality. We therefore adjusted our objective to build a gender classifier, which reached near state-of-the-art accuracy [3], before proceeding to build a CycleGAN, trained with the same data, to, in many cases, quite convincingly convert the gender (male to female or vice-versa) of a facial image.

2 Related work

We explored classifiers for determining facial characteristics and GANs to enable altering facial images by changing specific characteristics. We found the VGG-16 architecture [4], a preferred choice for extracting features from images, to be particularly promising. The weight configuration of the VGG-16 is also publicly available and has been optimized for multiple different use cases, including extracting features from facial images.

For the generative model, we focused on a Generative Adversarial Network (GAN) model called CycleGAN (Cycle Consistent GAN) [5]. This model fit our needs well as it is a proven and efficient model to learn a feature mapping between domains, which in our case, is the mapping between males and females. In addition, the CycleGAN can be trained on unpaired data, which was perfect for our purposes since we don't possess a data set where we would have both male and female versions of an individual. Using two pairs of discriminator and generators, a CycleGAN is able to perform unpaired learning.

A common downside of the VGG-16 and CycleGAN is the size of the models; the VGG-16 has approximately 130 million parameters and CycleGAN is even larger. We compensated for this by employing greater computational power (AWS P3 instances) as well as prolonged training.

CS230: Deep Learning, Winter 2019, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

3 Data Set and Features

Two days after the first lecture an Associate Professor at Stanford offered us access to a vast array of facial images labeled with personality information, including 6 million Facebook profile images with Big 5 personality and other personal data [6]. After two in-person meetings, however, the offer was revoked right after the milestone, without much explanation.

While working to gain access to data above, we tested models with the everypolitician.org [7] data set of 77,732 politicians throughout the world. Here we attempted to build a political party classifier, again using facial images, but were unable to get the accuracy even modestly better than random.

Fortunately, in mid-February, we were granted access to the brand new Diversity in Faces (DiF) [8] data set from IBM. While coming under some controversy just days ago [9], it contains nearly 1 million faces (with bounding boxes) labeled for skin color, age and gender. As implied by the name, the motivation for the data set is to enable ML algorithms to work with genders, ages and races that are equally represented [10].

The everypolitician.org data set was challenging, but we were able to download all members of the U.S. Congress from the past 20 years (1845) labeled by political party and split 50.2% Democrat and 49.8% Republican. (The few independents were removed.) We leveraged a pre-trained facial recognition neural network [11] to uniformly crop the faces. The resulting square head-shots, color and gray-scale, were used to train a CNN to predict political affiliation.

The DiF data was easier to download and manipulate. The images were fetched from URLs one at a time, but they were accompanied by bounding boxes to crop the faces and each was labeled for age and gender. We initially downloaded only 10 thousand to get started, split evenly across gender, age and race, and fed those into our CNN to train it to predict gender. While working on different models we continued to download and process the roughly 950k more images, which took ~80 hours spread across a week.

Once downloaded, we noticed several thousand of the images were blank, so they were removed. We also removed ~111k baby pictures (under 4 years old) and any images with 'unknown' genders. Finally, with slightly over 8600 images in the development set, we scanned them visually and removed any examples that were clearly mislabeled. The final data set consisted of 855,397 examples, split evenly between male and female with 1% then reserved for the test set.

For the CycleGAN we experimented with different data set sizes and found that a very large data set didn't yield good results. A moderate data set (20,000 to 50,000 examples for each class) performed much better. We filtered out children, low resolution and invalid images. All images were re-sized to 256x256 resolution during training.

4 Methods

For the classifier our initial supposition was that a huge CNN trained on thousands of different classes, such as VGG-16 or Inception, wouldn't provide the accuracy of something specifically trained on facial images. Substantial research, however, led us to realize that even popular gender and racial classifiers are built on VGG-16 or Inception while CNNs trained specifically on facial images are rare and not particularly robust. As such, we found a repository using transfer learning to build a classifier on top of a VGG-16 model [12] and then reworked that code to train on both the everypolitician.org and DiF data sets.

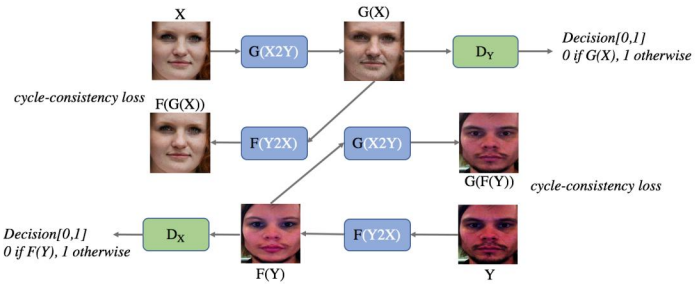

Figure 1: CycleGAN Model

The CycleGAN [5] aims at finding mappings between source and target domains for a given image without any pairing information, such as grey-scale to color, image to semantic labels, edge-map to photograph, horse to zebra, or so on. The CycleGAN model (Figure 1) consists of two generators G(X), F(X) and two discriminators D_X, D_Y. Half of the model (G and D_Y) is trained with inputs from domain X while the other half (F and D_X) is trained with inputs from domain Y. The "cycle" part involves reconverting the newly generated G(X) image, that is now in domain Y, back to an image in domain X. The same process applies to the F(Y) image. Ensuring the generated "cyclic" image is close enough to the original input image guarantees a meaningful mapping is defined, without the need of a paired data set.

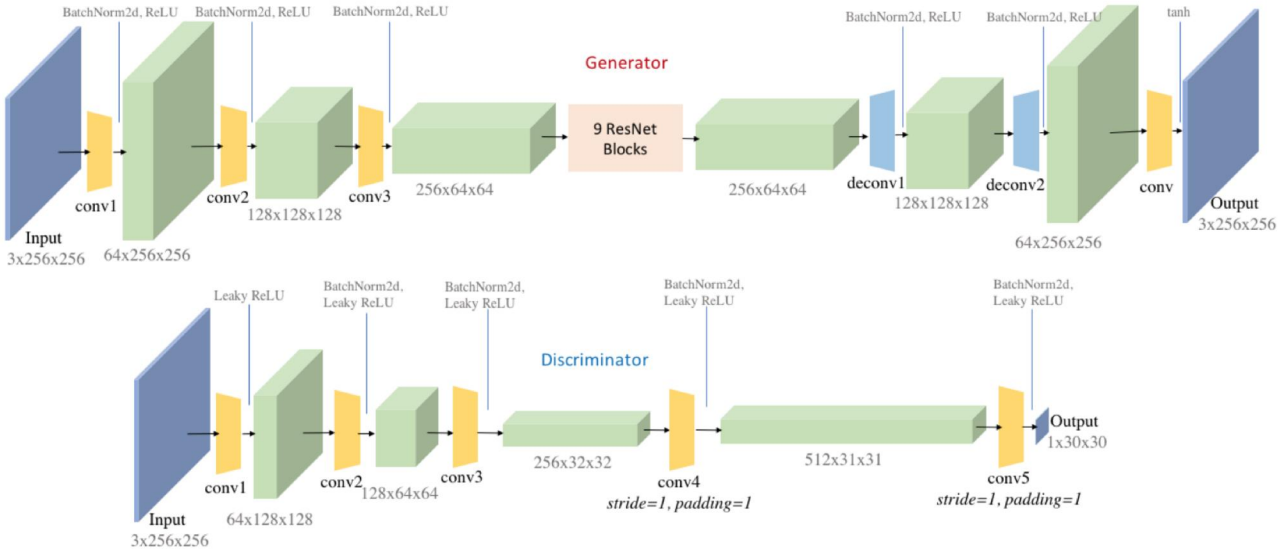

Figure 2: Generator and Discriminator Networks

We used the implementation provided by the original CycleGAN paper (Figure 2). The generator network has an Encoder (several convolutional layers), followed by a Transformer (9 ResNet Blocks) and finally a Decoder (several convolutional layers). The discriminator is simply a convolutional network consisting of 5 down sampling layers.

The CycleGAN model has two loss functions:

• **Adversarial loss:** This matches the generated image's distribution to the target domain distribution.

$$\mathcal{L}_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{data}(y)}\Big[logD_Y(y)\Big] + \mathbb{E}_{x \sim p_{data}(x)}\Big[log\big(1 - D_Y(G(x))\big)\Big]$$

• **Cycle consistency loss:** This prevents the learned mappings G and F from contradicting each other.

$$\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{x \sim p_{data}(x)}\Big[\Big\|F\big(G\left(x\right)\big) - x\Big\|_1\Big] + \mathbb{E}_{y \sim p_{data}(y)}\Big[\Big\|F\big(G\left(y\right)\big) - y\Big\|_1\Big]$$

The full CycleGAN objective function is given by $G^*, F^* = \arg\min_{F,G} \max_{D_x, D_Y} \mathcal{L}(G, F, D_X, D_Y)$ where $\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, Y, X) + \lambda \mathcal{L}_{cyc}(G, F)$.

5 Experiments, Results and Discussion

We anticipated possible difficulties identifying correlations between independent (i.e. facial images) and dependent (i.e. personality traits) variables, which we quickly encountered with the everypolitician.org head shots. We tried multiple classifiers on top of the VGG-16, using 1 and 2 FC layers, with and without dropout, and categorical cross-entropy for loss and accuracy as the performance metric, but were not able to improve prediction accuracy beyond 57.0%, which is not much better than random (Table 1). Attempts to train the deepest layers of the VGG-16 did not help.

Accuracy	FC layers	Regularization	Epochs	Batch Size	VGG-16
56.8%	1 FC layer with 1024 nodes	50% dropout	20 epochs	batch size 20	none
53.6%	2 FC layers with 1024 nodes	50% dropout	20 epochs	batch size 20	none
56.6%	2 FC layers with 1024 nodes	50% dropout	40 epochs	batch size 20	none
57.0%	2 FC layers with 1024 nodes	no dropout	40 epochs	batch size 8	none
51.8%	1 FC layer with 2048 nodes	no dropout	40 epochs	batch size 8	none
56.6%	1 FC layer with 1024 nodes	50% dropout	20 epochs	batch size 20	blocks 4, 5

Table 1: Experimentation with everypolitician.org data set

Rather than persist trying to improve political party predictions by experimenting with the model, gathering more data by expanding to other countries or performing more pre-processing on the data, such as converting everything to grayscale, we really only wanted to validate our model and approach. As such, we switched to the DiF data set and attempted to train the CNN to predict gender, which we assumed, based on our human perception, to be a significantly easier task. We were correct.

With only 1000 images, roughly evenly split between males and females and then split 70/30 between training and test, we were able to achieve 68.5% accuracy with a single 1024-node FC layer and 50% dropout (Table 2). While discerning differences in facial characteristics between politicians from the two major political parties was difficult, this first step with gender classification was promising. Continuing with our gender classifier, we tried again with 10k images split 90/10 and were able to achieve 75.8% accuracy with the same model (Figure 3).

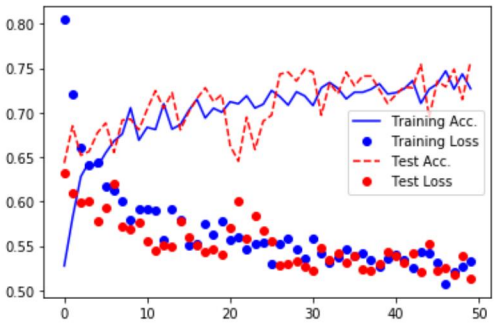

Figure 3: 10k examples initial testing

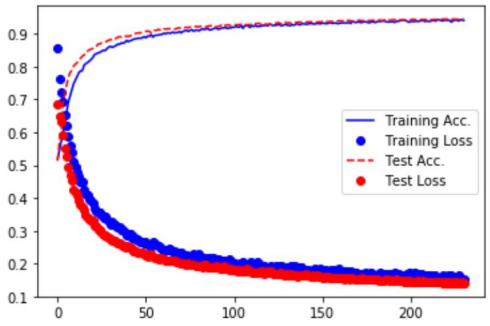

Figure 4: 855k examples & full VGG-16 fine tuning

To squeeze out more performance while we continued to download and process the DiF data set, we tried *fine tuning* the last block of the pre-trained VGG-16, consisting of 3 convolutional layers followed by pooling. After training for more than twice as long we were only able to achieve 64.0% accuracy. The trajectory looked solidly in the right direction, however, so to test that theory we decided to train the same model again for 250 epochs, which took $4\frac{1}{2}$ hours on a local GPU. Regardless, we were only able to reach 69.5% accuracy.

We surmise from this experience, and the course material, that there might be two reasons why we saw a degradation in performance when fine tuning the VGG-16: 1) training even the deepest layer would require significantly more data and compute time; 2) our own FC layer had not been sufficiently trained to back-propagate the gradients to the earlier layers of the VGG-16.

Accuracy	FC layers	Regularization	Batch Size	# Batches	Images	VGG-16
68.5%	1x1024 nodes	50% dropout	20	2k batches	1000	none
75.8%	1x1024 nodes	50% dropout	20	5k batches	10k	none
64.0%	1x1024 nodes	50% dropout	20	10k batches	10k	block #5
69.5%	1x1024 nodes	50% dropout	20	25k batches	10k	block #5
73.6%	2x512 nodes	50% dropout	64	5k batches	10k	none
71.0%	3x512 nodes	50% dropout	32	5k batches	10k	none
89.1%	1x1024 nodes	50% dropout	10	5k batches	855k	none
88.8%	1x2048 nodes	50% dropout	10	40k batches	855k	none
87.5%	2x1024 nodes	50% dropout	20	20k batches	855k	none
86.8%	3x1024 nodes	80% dropout	20	40k batches	855k	none
89.2%	3x1024 nodes	20% dropout	20	40k batches	855k	none
94.9%	2x1024 nodes	50% dropout	20	26k batches	855k	blocks #4, 5
95.2%	2x1024 nodes	50% dropout	20	28k batches	855k	blocks #3, 4, 5
95.1%	2x1024 nodes	50% dropout	10	658k batches	855k	all blocks[1]

Table 2: Experimentation with Diversity in Faces data set

After downloading and processing the full 960k images from the DiF data set, we were able to experiment with a number of different models and hyperparameters, eventually achieving a remarkable 95.2% accuracy (Table 2). An experiment training the full VGG-16, using a recommended learning rate (Figure 4), did not product the best result. By examining examples of classification mistakes (Figure 5), we were able to identify that many of the mistakes were a result of misclassification, difficult lighting and young children. A 'cleaner' data set could have potentially produced even better results.

For the CycleGAN model, we experimented with different setups (Table 3), adjusting the number of images in the training set, freezing discriminators and adding skip connection between input and output in the generator. We manually examined the quality of the generated images and chose model #4. Sample images from that model after epoch 30 are shown below (Figure 6).

[1]Learning rate dropped from 10^{-5} to 10^{-7}. Training took over $5\frac{1}{2}$ days on a local GPU.

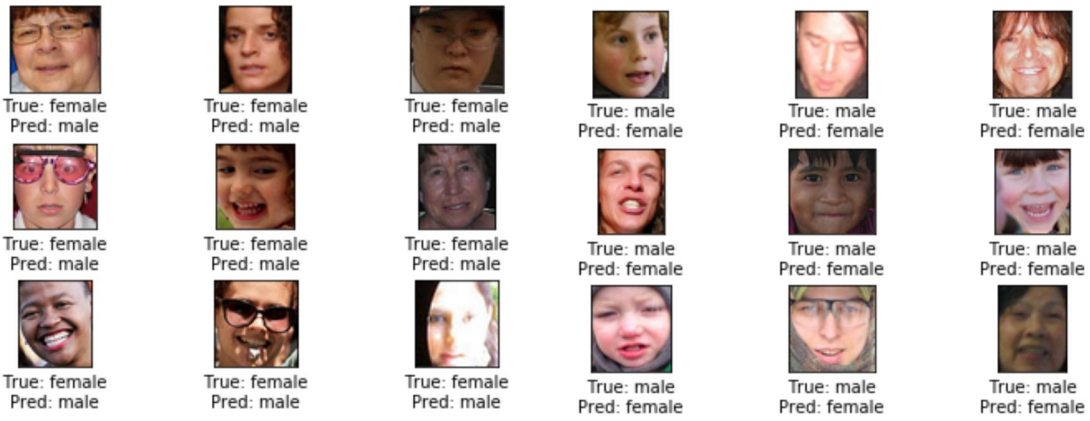

Figure 5: Example gender classification mistakes

#	Setup	Images	Skip	Epochs	Results
1	Tensorflow CycleGAN	155k	False	1	Aborted. Images had multiple eyes, noses, etc.
2	Tensorflow CycleGAN	2k	False	200	Faster training. Images were mediocre quality.
3	Tensorflow CycleGAN	20k	False	100	Acceptable training times. Images were good after 80 epochs.
4	Tensorflow CycleGAN	50k	True	30	Faster convergence. Images were good after 20 epochs.
5	Keras CycleGAN[2]	100k	N/A	100	Faster training. Gender conversion was not noticeable.

Table 3: Experimentation with CycleGAN

Figure 6: Sample results of the gender conversion. Top row is the original and the bottom row is the "fake".

6 Conclusion and Future Work

The gender classifier demonstrated excellent performance, so improving from here would most likely involve using the classifier to identify and correct mislabeled items in the training and test sets. We could conceivably also continue to train the full VGG-16 with a 10^{-7} learning rate, or perhaps even smaller, although marginal improvements are vanishingly small. The same model could also be easily re-purposed, with the DiF data set, to produce an age classifier.

The CycleGAN demonstrated encouraging signs of producing interesting results from our initial training, but required significantly more time to train. Further experimentation could be done by changing hyperparameters as well as the use of transfer learning to speed-up model convergence and improve model performance. In addition, we would have liked to investigate the use of Inception Score (IS) and Frechet Inception Distance (FID) for evaluating GAN performance rather than rely on manual examination, which is qualitative.

While we were never able to get the facial images labeled with personality data, our work here demonstrates relatively straightforward paths to using such data to produce personality classifiers as well as a CycleGAN to potentially modify facial images by altering personality inputs. Eventual access to the aforementioned data sets could, through multi-task learning, build a classifier for multiple predictions per image.

We could simultaneously look to build an invertible conditional GAN. In our proposal we identified one for image editing [13], available on GitHub [14], that could potentially be re-purposed. We also found a model described in “Face Aging with Conditional GANs” [15] which looked promising.

[2]With discriminators frozen

7 Contributions

Mark Cramer: Acquired data sets (discussions with Stanford Assoc. Professor regarding Facebook data and submitted application to IBM for DiF data). Wrote code to fetch and process everypolitican.org data and ran political party classifier on local GPU. Wrote code to fetch and process DiF data. Wrote code to optimize VGG-16 gender classifier, including saving and reloading models, and ran all training on local GPU. Spun up p3.2xlarge on AWS and trained Keras version of CycleGAN.

Srinivas Halembar: Researched the available pre-trained models that can be leveraged for the Gender Classification model using Transfer Learning. Explored the available GAN models for Gender conversion. Trained the Cycle GAN models with different input sizes and training parameters, baselined and consolidated the results.

Harry Jiang: Wrote code to leverage a face detection model to download, process and crop face images from the everypolitician.org data set. Handled setting up Amazon AWS instances/account groups. Worked on the project proposal, project milestone as well as final report/poster-related production.

8 Repository

https://github.com/mdcramer/CS230-Deep-Learning/

9 Acknowledgement

We would like to thank Hojat Ghorbanidehno for guiding us during the project. We would also like to thank the CS230 teaching staff for their assistance on Piazza. Finally, Mark would like to thank his wife for putting up with all the late nights in front of the computer and the constant whirring from the GPU fan, and Srinivas would like to thank his wife for putting up with all the late nights in the office working on this project.

References

[1] R W Squier and J R Mew. The relationship between facial structure and personality characteristics., Sep 1981.

[2] Jason M Gold, Patrick J Mundy, and Bosco S Tjan. The perception of a face is no more than the sum of its parts, Mar 2012.

[3] Joseph Lemley, Sami Abdul-Wahid, Dipayan Banik, and Razvan Andonie. Comparison of recent machine learning techniques for gender recognition from facial images. 2016.

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

[5] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.

[6] Mypersonality project. `https://sites.google.com/michalkosinski.com/mypersonality/home`. Accessed: 2019.

[7] Every politician. `https://everypolitician.org/`. Accessed: 2019.

[8] Michele Merler, Nalini K. Ratha, Rogério Schmidt Feris, and John R. Smith. Diversity in faces. *CoRR*, abs/1901.10436, 2019.

[9] Olivia Solon. Facial recognition's 'dirty little secret': Millions of online photos scraped without consent. *NBC News*, 2019.

[10] Jed Kim. Before facial recognition tech can be fair, it needs to be diverse. *Marketplace Tech Blogs*, 2019.

[11] Adam Geitgey. Face recognition: Facial recognition api for python and the command line. `https://github.com/ageitgey/face_recognition`.

[12] Hvass-Labs. Tensorflow-tutorials. `https://github.com/Hvass-Labs/TensorFlow-Tutorials`.

[13] Guim Perarnau, Joost van de Weijer, Bogdan Raducanu, and Jose M. Álvarez. Invertible Conditional GANs for image editing. In *NIPS Workshop on Adversarial Training*, 2016.

[14] Guim. Invertible conditional gans for image editing. `https://github.com/Guim3/IcGAN`, 2017.

[15] Grigory Antipov, Moez Baccouche, and Jean-Luc Dugelay. Face aging with conditional generative adversarial networks. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 2089–2093. IEEE, 2017.