Generation of Realistic Facemasked Faces With GANs

Samuel Mumford (smumfor2@stanford.edu)

March 18, 2021

Abstract

Facial recognition and identification are key aspects of modern technology and are impeded by the advent of facemasks to combat to Covid-19. Large datasets of masked faces are needed in order to improve facial identification with masks. Such datasets may be produces using generative adversarial networks. In this work, realistic pictures of faces with facemasks are generated using a CycleGAN model. A less computationally intensive SimGAN model is also used as a point of comparison. The generated pictures of faces with masks can be used for facial identification and mapped back to the original face with a .2% error rate, demonstrating the usefulness of GAN models in face recognition.

1 Introduction

Facial identification is a benchmark problem in convolutional neural networks[1][2]. Correspondingly, there are many established datasets of faces used to train models[3][4]. However, the recent adoption of facemasks to slow the spread of Covid-19 creates a facial recognition challenge without large datasets of masked faces. Masked facial identification thus necessitates taking large quantities of new data[5]. Alternatively, pictures of unmasked faces can be converted to pictures of masked faces with generative adversarial networks (GANs). In this work, I demonstrate the performance of two GANs in converting pictures of faces into masked faces and the corresponding consequences on facial recognition.

2 Datasets

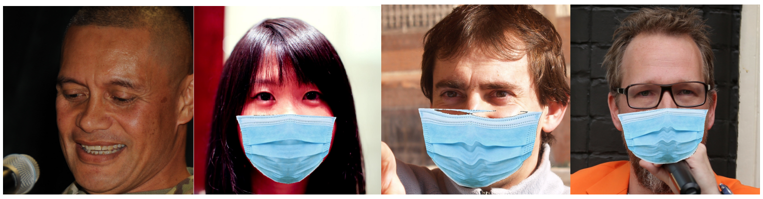

Figure 1: Example face pictures to be used for training and evaluation[6][7]. Note that the artificially added masks are uniformly blue masks and exhibit rare problems such as discontinuous features and mishandling of real-world objects.

A dataset of 67,193 masked faces[7], in turn formatted to work alongside a dataset of 70,000 faces[6] was used. The dataset of masked faces contains artificial facemasks superimposed onto faces as shown in Fig. 1 instead of true facemask pictures[7]. In order to transfer files and train more quickly, a subset of 2000 train and 1000 test images was used from each of these larger datasets. There was no evidence of model overfitting or a train/test set mismatch in model performance, and so the reduced dataset was sufficient.

3 Model 1: CycleGAN

CycleGAN models have been used for tasks such as converting pictures of horses to zebras with better performance than similarly structured GAN models[8][9]. A CycleGAN approach correspondingly was used to add masks to face pictures. CycleGAN loss primarily consists of GAN loss with a two trained discriminators D_X and D_Y and generators G and F. The GAN loss used on one generator-discriminator pair evaluated on X and Y with n_x and n_y examples is,

$$\mathscr{L}_{GAN}(G, D_Y, X, Y) = \frac{\mathbb{1}[y \text{ from } Y]}{n_y} \ln\left[D_Y(y)\right] + \frac{\mathbb{1}[x \text{ from } X]}{n_x} \ln\left[1 - D_Y(x)\right]. \qquad [3.1]$$

Additionally, a cycle loss term penalizes unnecessary image changes. The cycle loss[8]

$$\mathscr{L}_{cyc}(G, F, X, Y) = \frac{\mathbb{1}[x \text{ from } X]}{n_x} \|F(G(x)) - x\|_1 + \frac{\mathbb{1}[y \text{ from } Y]}{n_y} \|G(F(y)) - y\|_1 \qquad [3.2]$$

evaluates the distance between the original image and the result of attempted mapping of that image into the complimentary dataset and then back into the original dataset. The full loss with cycling parameter λ is $\mathscr{L} = \mathscr{L}_{GAN}(G, D_Y, X, Y) + \mathscr{L}_{GAN}(F, D_X, Y, X) + \lambda \mathscr{L}_{cyc}(G, F, X, Y)$ and was used for training in an Adam optimizer[10].

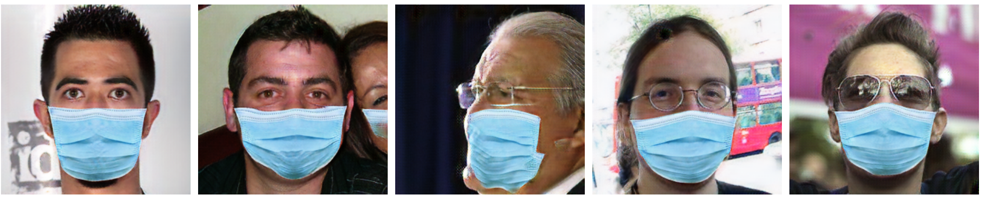

Figure 2: Results of the final CycleGAN model applied to the first five initially maskless pictures. Masks remain slightly transparent, but transparency has been significantly reduced by reducing λ.

A CycleGAN model was trained to add masks to unmasked faces[8][11]. As described in[8], generator networks were formed from three initial convolutions, nine 64-channel convolutional ResNET blocks, two fractionally strided convolutions, and a final convolution to reduce output to three channels. The discriminator CNN is evaluated on overlapping 70x70 image patches[8][12]. Three convolutional layers with a stride of two are applied to each image patch to increase the activation depth and reduce activation width. The reshaped intermediate layer is flattened and fed into a fully connected single neuron sigmoid output[8][12].

The CycleGAN model was trained to produce realistic facemasked images. No changes were needed in the base implementation other than formatting pictures into the required folder structure and tuning hyperparameters[11]. Model performance and training effectiveness were evaluated by Fréchet inception distance (FID)[13]. Examples of the best performing model are displayed in Fig. 2, and earlier generations of training are provided in Appendix A. Model performance improved dramatically with adequate monitoring and training time as shown in Table 1. The largest improvement to performance came from increasing training time under the default parameters. This is to be expected as the default parameters were established on similar style transfer problems. However, performance deteriorated with increased training time after 35 epochs, suggesting that the learning rate was too high. Reducing the learning rate to slowly converge on optimal performance and reducing λ to make the masks less transparent yielded generated images which are difficult to distinguish from the base dataset images and train/test FID scores of 28.43/28.48, demonstrating the effectiveness of the CycleGAN approach.

Training Procedure	Additional Epochs Trained	Train FID	Test FID
Initial ($\alpha = 2E - 4$, $\lambda = .5$)	10	78.1	78.3
Reduced $\lambda = .005$	7	149.1	150.0
Continued Training	25	42.45	42.02
Continued Training	10	48.56	48.73
Reduced $\alpha = 2.5E - 5$	20	31.14	30.06
Reduced $\lambda = .1$	30	28.43	28.48

Table 1: Outline of the training procedure used to produce a final model with results shown in Fig. 2. Note that α is used to denote the learning rate. The model displays low variance and improved performance after prolonged and tuned training.

4 Model 2: SimGAN

Although the CycleGAN approach produces high-quality masked faces, it requires high computational power and time to train. A faster-training model using only one generator and discriminator could be useful in mobile applications. A SimGAN model penalizes changes to the original image x, giving an additional loss term $\mathcal{L}_{img}(G, X) = ||G(x) - x||_1$ and total loss with penalty factor β of $\mathcal{L} = \mathcal{L}_{GAN}(G, D, X, Y) + \beta\mathcal{L}_{img}$. This simpler loss function does not require a second generator-discriminator pair and can be evaluated using fewer than half as many forward passes as the CycleGAN loss.

Alternate loss and model structures were considered. A CoGAN model is also based on coupled generators and maps inputs to a feature vector z for every test example, which increases computational cost and creates blurry images when generalizing[14]. Such feature mapping is also an issue for BiGAN/ALI models[15][16]. A SimGAN model therefore was chosen as a lower-cost alternative among the commonly used image generation models[8].

SimGAN was implemented convolutionally for both the discriminator and generator models. Significant changes were made to the base implementation to accommodate new images, account for updates to TensorFlow, and improve performance[17][18]. First, the dataset structure and batch loading were rebuilt to accommodate multi-channel images of variable size and a new data structure. Second, a flattening and fully-connected layer were added to the discriminator model to accommodate images with variable sizes. Third, batch normalization layers were added to each ResNet block[19] in the generator and before the fully connected layer of the discriminator network to avoid exploding gradients. Fourth, Adam optimization was implemented[10]. Finally, the discriminator was frozen during generator training. The best SimGAN generator network consisted of an initial channel-altering convolution, eight 64-channel ResNET blocks, and a final convolution to change output to three channels. The best discriminator model consisted of five convolutional layers, two pooling layers, flattening, and a final fully connected layer to outputs.

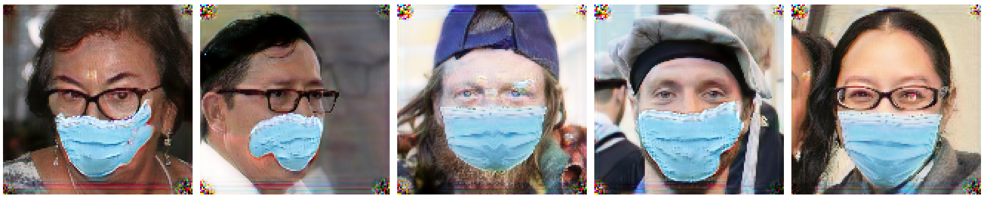

Figure 3: Results of the best SimGAN model trained, yielding Train/Test FID scores of 169.14/136.55. Masks were added to the correct locations on faces, but there are unrealistic changes made to the edges of pictures and other facial features.

Training Procedure	Additional Epochs Trained	Train FID	Test FID
Initial ($\alpha = 1E - 4$, $\beta = 2E - 4$)	20	307.5	217.8
Reduced $\beta = 8E - 4$	20	370.8	270.1
8 ResNET Blocks	8	319.4	233.7
Reduced $\alpha = 1E - 5$	30	199.39	163.58
Reduced $\alpha = 3E - 6$	50	169.1	136.6
Doubled D Update Rate	20	173.7	137.3
Increased D Layers	30	217.6	139.2

Table 2: Outline of the training procedure used to produce a final model with results shown in Fig. 3. Training in the initial 4 ResNET block model occurred at a rate of 8 epochs per hour, while the 8 ResNET block trained at around half that rate. The initial 8 ResNET block model was trained for more than 8 epochs, but produced cyclical results and was stopped early, giving 8 effective epochs of training.

A variety of training parameters and architectures were explored, with results of earlier models shown in Appendix B and example outputs of the best SimGAN model shown in Fig. 3. The most significant improvements to performance were caused by changing β, α, and the generator model size as seen in Table 2. The penalty β was reduced to account for larger 3-channel RGB images and allow the generator to add blue features as seen in Fig. 11. However, the discriminator loss was consistently low and generated images were distorted but not mask-like, and so the generator model was doubled to eight ResNet blocks. This incurred the largest improvement in performance, as seen in the difference betwen Fig. 11 and 12. Training with a base $\alpha = .0001$ became cyclical as generator results alternated between blue masked faces or bare faces with a period of ≈ 20 epochs after increasing generator power. This was remedied by early stopping and then reducing α by a factor of 10. However, performance remained poor when compared to the CycleGAN and exhibited high discriminator loss after hyperparameter tuning. Changing the ratio of the discriminator to generator update steps, altering the SimGAN penalty, reducing learning rate, and increasing the number of layers in the discriminator model did not significantly improve performance. Performance likely could be improved by training larger discriminator and generator models, but such additional power would defeat the purpose of the SimGAN as an alternative to the functional CycleGAN model.

5 Application: Toy Model of Facial Recognition

Facial recognition and verification programs must be altered in order to recognize faces with masks reliably. A first approach to facial recognition with masks was to remove encoding projections along the mask feature axis. Given a face picture $p^{(i)}_{face}$, an analogous mask picture $p^{(i)}_{mask}$ was produced by the CycleGAN. Picture encoding vectors $e^{(i)}_{face}$ and $e^{(i)}_{mask}$ were then produced by feature-extracting neural networks such as the Inception network[13][20] or a VGG19 network[21]. The face-mask bias vector f defined over m picture pairs was therefore

$$f = \frac{1}{m}\sum_{i}^{m}(e^{(i)}_{mask} - e^{(i)}_{face}) \longrightarrow \hat{f} = \frac{f}{||f||_2}. \qquad [5.1]$$

Finally, encodings $e'^{(i)}_{face}$ and $e'^{(i)}_{mask}$ were produced by removing the encoding projection onto $\hat{f}$.

The difference between feature encodings were evaluated by cosine similarity, denoted as $\cos(e'^{(a)}_{face}, e'^{(b)}_{mask})$ for base images a and b in Table 3. The cosine similarities of pictures of the same face and different faces must be distinct to perform facial recognition. As seen in Table 3, appropriate encoding network choice and projected mask removal improved facial recognition performance. The distributions of pictures of the same face and different faces were separated by 2.677 σ after debiasing, giving a 2% error rate.

Approach	Encoding	$\cos(e'^{(i)}_{face}, e'^{(i)}_{mask})$	$\cos(e'^{(i)}_{face}, e'^{(j)}_{mask})$	Difference in σ	Error Rate
Baseline	Incep.	.963 ± .021	.904 ± .037	1.367	.149
Bias Projection	Incep.	.970 ± .019	.906 ± .041	1.416	.147
CycleGAN Strip	Incep.	.989 ± .009	.901 ± .043	2.003	.047
Baseline	VGG19	.768 ± .064	.569 ± .079	1.957	.082
Bias Projection	VGG19	.890 ± .036	.660 ± .078	2.677	.022
CycleGAN Strip	VGG19	.945 ± .024	.654 ± .080	3.352	.002

Table 3: Performance of facial recognition using CycleGAN-produced mask pictures. The error rate is defined assuming a Gaussian distribution in cosine similarities and placing the facial recognition threshold equidistant from the centers of each distribution in standard deviations.

The de-masking generator of the CycleGAN was also used to remove facemasks with results shown in Table 3. The CycleGAN generator outperforms the bias projection method, yielding a .2% error rate. Such improvement in performance can be explained by the flexibility of a CycleGAN model over bias projection. The CycleGAN can remove masks regardless of facial angle or underlying facial expression while bias projection only accounts for the average features of masks. However, using a trained CycleGAN model to remove masks is more computationally expensive and a larger change to standard facial recognition algorithms.

6 Conclusions

A CycleGAN model may be trained to produce realistic images of faces with facemasks much has been done in other style transfer problems. It is also possible to make flawed but recognizable facemasked pictures with a SimGAN model with one tenth of the number of parameters of the CycleGAN model and half the number of generators and discriminators. A corpus of artificially masked pictures can also improve the performance of facial recognition on new masked pictures. Such generative and facial recognition techniques must be generalized to pictures of real faces with a variety of masks in order to be practically applied.

References

[1] S. Lawrence, C. L. Giles, Ah Chung Tsoi, and A. D. Back. Face recognition: a convolutional neural-network approach. *IEEE Transactions on Neural Networks*, 8(1):98–113, 1997.

[2] Yi Sun, Ding Liang, Xiaogang Wang, and Xiaoou Tang. Deepid3: Face recognition with very deep neural networks, 2015.

[3] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face ...

[4] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang.

[5] Samp;T Public Affairs. News release: Airport screening while wearing masks test, Jan 2021.

[6] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018.

[7] Adnane Cabani, Karim Hammoudi, Halim Benhabiles, and Mahmoud Melkemi. Maskedface-net – a dataset of correctly/incorrectly masked face images in the context of covid-19. *Smart Health*, 2020.

[8] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2020.

[9] Adrian Rosebrock. Covid-19: Face mask detector with opencv, keras/tensorflow, and deep learning, Jun 2020.

[10] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017.

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018.

[13] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. `https://github.com/mseitzer/pytorch-fid`, August 2020. Version 0.1.1.

[14] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks, 2016.

[15] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference, 2017.

[16] A.A. Efros and T.K. Leung. Texture synthesis by non-parametric sampling. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999.

[17] Aymen B Bothmena. Simgan implementation using tensorflow/keras, Apr 2019.

[18] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. Learning from simulated and unsupervised images through adversarial training, 2017.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[20] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

A Previous Generation Outputs of CycleGAN

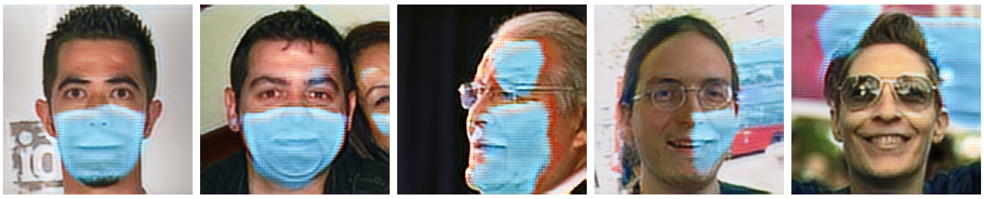

Figure 4: Training on initial parameters. Note that the masks are amorphous, semi-transparent, and often placed in the wrong location.

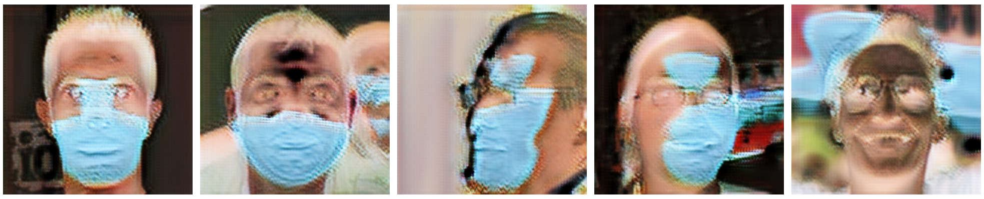

Figure 5: Training with reduced λ, showing worsening cycle performance without improving the masks.

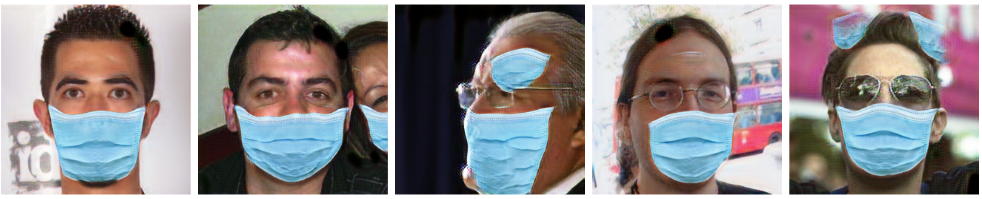

Figure 6: Continued training on initial parameters, showing improvement with longer training. Masks now have the correct shape, but artifacts of masks are outside of the lower face.

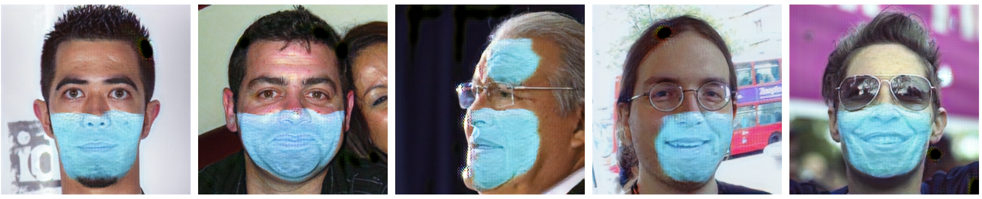

Figure 7: Continued training on the same parameters as Fig. 6, note that performance deteriorated as confirmed by the FID.

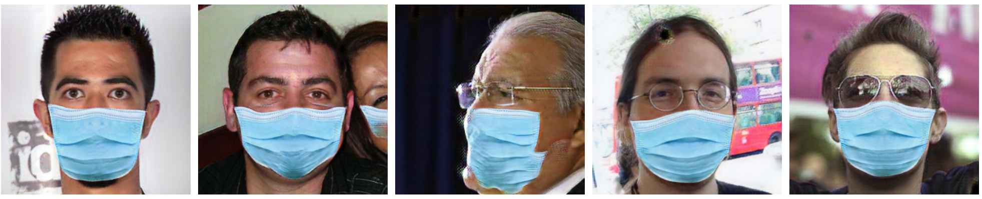

Figure 8: Results after decreasing α, which dramatically increased the plausibility of mask pictures. However, masks were still slightly transparent.

B Previous Generation Outputs of SimGAN

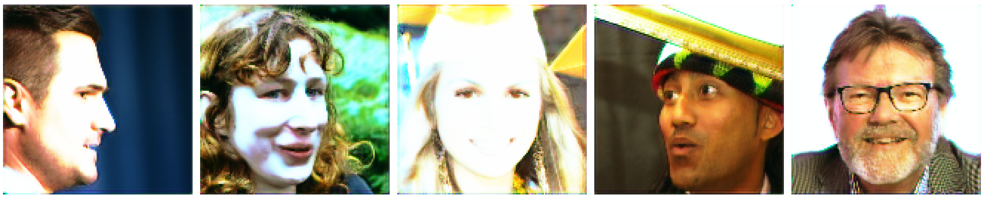

Figure 9: Results of pre-training the SimGAN model to minimize only $\mathscr{L}_{img}$. This serves as a baseline model for further training. Pictures look color-shifted or low definition, but the model now recognizably outputs a similar image to the input, verifying the base generator training procedure.

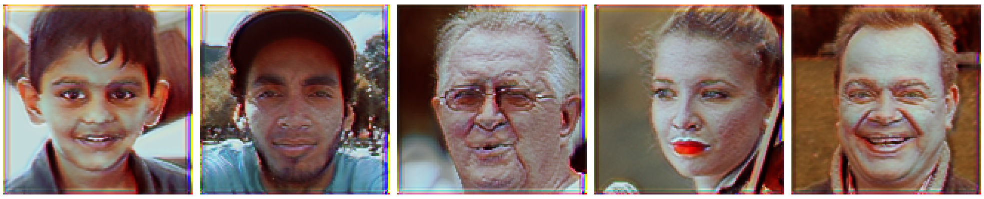

Figure 10: Results of the first generation of adversarial training over 32 epochs. The loss was overly dictated by the $\mathscr{L}_{img}$, and there is little evidence of mask-like features being added by the network. Initial training was worse than the results shown in Fig. 4, which is to be expected as the initial training parameters used for the CycleGAN model were developed for more similar problems and data.

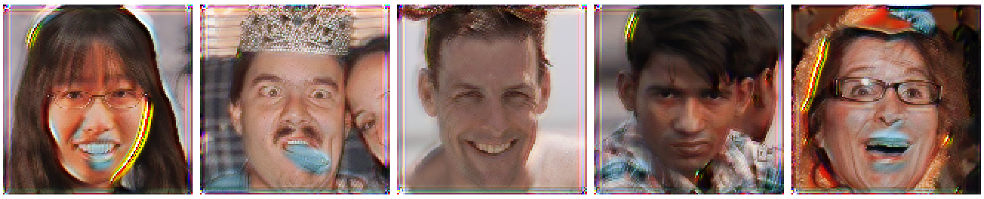

Figure 11: Results of the first productive generation of SimGAN model training. Examples of the pre-trained model and early training are shown in Appendix B. The learning rate, β, and batch sizes were adjusted to make the generator model produce blue faces and mouths. Train/Test FID scores were 370.8/270.1, worse than even the first generations of the CycleGAN. However, the results are not realistic despite many unnecessary changes being made to the images.

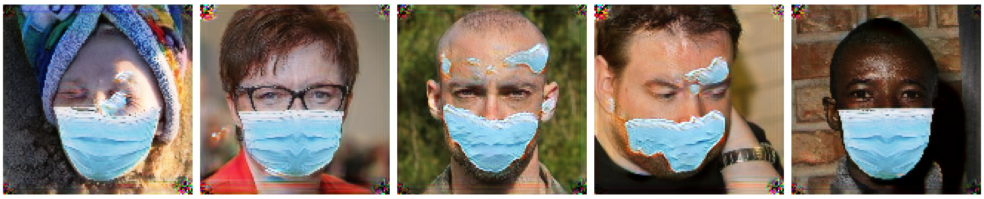

Figure 12: Results from early training on an 8-layer ResNet SimGan model. Train/Test FID scores were 163.6/199.3, and results are qualitatively improved from Fig. 11. Note that the noise at the corners expanded once the generator model became more powerful.

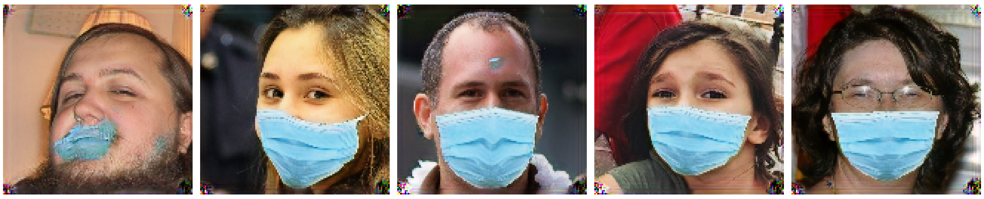

Figure 13: Results from training with two convoluntional layers added to the discriminator model. Training was slower with minimal improvement in performance, evidenced by the similar Train/Test FID scores of 217.6/139.2.