White Noise Reduction using Style-Based Generative Adversarial Networks

Denzell Ford[1], Alexander Billiot[1], Rahul Shanmugham[1], Lydia Wharton[1], Swapnil Shaurya[1], Jesse Pisel[1,2], Michael Pyrcz[3,4]

[1]. College of Natural Sciences, The University of Texas at Austin, 120 Inner Campus Drive, Stop G2500 Austin, Texas 78712

[2]. Texas Institute for Discovery Education in Science, The University of Texas at Austin, 120 Inner Campus Drive Stop G2500 Austin, Texas 78712

[3].Cockrell School of Engineering, The University of Texas at Austin, 200 East Dean Keeton Street, Stop C0300 Austin, Texas 78712

[4].Jackson School of Geosciences, The University of Texas at Austin, 2305 Speedway, Stop C1160Austin, Texas 7871

Abstract

In this paper, we demonstrate the efficacy of using a style-based generative adversarial network (StyleGAN) that implements the pixel2Style2pixel (pSp) framework to achieve white noise reduction on images. We do this by emulating the target as opposed to attempting to match the target exactly. With our implementation, we gain a substantial stylistic improvement over the input at the tradeoff of accuracy. While our system will most likely not be usable in cases where the accuracy of the denoised data is paramount, our model is very effective in denoising at high noise levels where other models struggle.

Executive Summary

White noise is random variation in any part of data that causes the data to be skewed. Removing this white noise from datasets increases data usability. Much of the data that has been acquired for research purposes cannot be used due to the random flaws found within them; however, our research would remove these flaws, thus data can be used much more widely than it is now. Our project seeks to find patterns within image data, and then use these patterns to

remove any white noise from the images. Using a machine-learning model called a generative adversarial network (GAN), we find a way to extract a style latent vector from each image and use these latent vectors to effectively remove the white noise. A simple real-world use case for our project includes removing static and white noise found within any images or videos and removing blurriness found in CCTV camera footage. Considering future work and greater applications for our project, it would be an interesting question whether our multi-faceted approach to image-to-image translation can be generalized to data types outside of images such as audio.

Introduction

Noised data is any type of data that is corrupted, distorted, or contains meaningless information (Bethge 2020). This can lead the false data of the noise being interpreted as meaningful when attempting to work with it (Bethge 2020). Therefore, denoising data to ensure it is clean or noise-free is very important before analyzing or running any type of model on it (Bethge 2020). Noise can be anything from blurs and dots on an image to distortion in audio to even deviation from a set pattern within data.

The topic of noise reduction is important as any physical, or real-world, data that is obtained will be subject to some amount of random variation. One clear example of this would be video feeds taken from closed-circuit television (CCTV) cameras. These video feeds are one type of real-world data that contain random amounts of static and white noise that make viewing the video quite difficult. White noise in particular is hard to eliminate, as it is completely random, and thus challenging to model. Machine learning models of white noise reduction are difficult to create, as the noise itself cannot be predicted.

We address this problem by using the underlying structured nature of the data to lower the white noise as opposed to predicting the noise within a model. Furthermore, we propose a noise reduction generative adversarial network (GAN) that will remove noise from a facial image as long as the structure of the image is strong enough to be determined (Guan 2020, Radford 2016).

GANs fall under the category of unsupervised learning and are nested in the field of machine learning known as deep generative learning, a subfield of artificial intelligence in which models learn to synthesize unique samples based on a given data distribution (Gautam 2020). Unsupervised learning differs from supervised learning in that the model is not given any human input besides the dataset and the model searches for patterns (Radford 2016). Although there are many types, GANs have two main networks: a generator and a discriminator. The job of the generator is to create an image that successfully fools the discriminator. The two compete against each other and try to get an error of zero, a type of zero-sum game approach (Gautam 2020). The generator produces an image tensor from a latent Z dimensional vector that will have a different output each time the generator and discriminator is trained. Basically, the latent vector works as a form of memory so the generator can map specific images to different points in the latent vector. Then, these different points in the latent vector are randomly selected and queried to create new images (Karras 2019, 2020). For this project specifically, these new images are images that have had the white noise filtered out due to the process that the generator and discriminators go through.

We use a StyleGAN encoder network implementing a pixel2style2pixel (pSp) framework for removing white noise from the images (Richardson 2020). This approach has seen tremendous results with StyleGAN inversions when pretrained on a specific set of images (Richardson 2020). StyleGANs are a specific type of generative adversarial network that allows us to customize different features of the generated image easily and enables intuitive, scale-specific control of the synthesis of our original images (Gal 2021, Nguyen 2020). The StyleGAN produces styles that control the layers of the synthesis network via adaptive instance normalization (AdaIN). AdalN is a normalization algorithm that "normalizes" or aligns the mean and variance of the content features with the style features of a network (Karras 2019, 2020). We exploit this by mapping our noised image into the latent vector for StyleGAN to produce an image. This means the encoder can create images from the produced latent codes more effectively and precisely (Tov 2021, Bethge 2020).

We analyze the output images based on their Fréchet inception distance (FID) in relation to the original image. This technique estimates a human determination of the similarity between two images. This is especially valuable for GAN architectures that are attempting to trick human perception.

Methods

In this section, we introduce the process for reducing noise in images. The goal is to produce an image clean of noise from a duplicated image with randomized white noise overlayed. First, we will discuss how we acquired our dataset. Then, we will overview which type of generative adversarial network we chose and why we chose it.

We use an aligned and cropped subset of the CelebA dataset (Liu 2015). This dataset is large enough to train without overfitting. To create a dataset of white noise that we could train our encoder network on, we chose to add noise to the images. It is important to note the noise level is controlled and adjusted, so we can create images with varying degrees of random white noise. This is important because we want our model to be able to generate clean images no matter the noise level of the input image. As seen in Figure 1, we have an image on the right from the CelebA dataset. The image on the left is the same image with structured noise at a level of 0.70 added onto the image, with noise being bounded between 0 and 1 with 0 being no noise and 1.0 being only noise.

Our pSp model is pretrained on a specific set of images. Due to this, we are able to focus on a very specific subset of formatted images, leading to a greater accuracy when removing noise from those same images. We perform qualitative experiments utilizing both the Flckr-Faces-HQ dataset (ffhq) as well as the CelebA dataset (Liu 2015, Karras 2019~~;~~). First, we will discuss our encoder network and the loss functions we use to calculate the cumulative loss of the model. Then, an overview of the various qualitative experiments we performed with our datasets.

The model's encoder network (GradualStyleBlock) receives the noised image input of batch-size 3x256x256 and converts this into a latent style vector used for styleGAN. This encodes the image into a vector of batch-size x 512 that the styleGAN interprets and produces a facial image. An interesting quirk of this is that we are not sure exactly which model "removes" the white noise. StyleGAN could be removing extra noise given to it from its style vector, or it could be the encoder network.

We use a loss function that is a combination of learned perceptual image patch similarity (LPIPS) with a weight of 0.80, mean squared error (MSE) loss with a weight of 1.0, weighted norm (wnorm) loss with a weight of 0.005, and identity loss with a weight of 0.90. We then sum up all our losses to get our cumulative loss. We train two models: one with a noise level of 0.55 for 49,500 iterations, and another with a noise level of 0.70 for 39,000 iterations. The number of iterations is a result of the number of iterations performed in two hours of training on the V100 nodes in the Maverick2 system at the Texas Advanced Computing Center. Both began training using the pretrained weights from the pSp model ~~(Richardson 2020)~~ trained on the ffhq dataset (Richardson 2020, Karras 2019). We trained both models with a batch size of 1 and the Adam optimizer with a learning rate of 0.0001.

Results

Regarding the qualitative experiments, the model performs poorly when there is little noise, and generally produces the best results at its targeted noise level (Figure 2). One interesting aspect to notice about the models when augmenting the original image is that we can see how the image is converted from one to the other. The model trained at 0.70 seems to darken the image more on average compared to the model at 0.55, which adds more greens to the image. The images on the left in Figure 2 were trained at a noise level of 0.55 for 49,500 iterations, while the images on the right were trained at a noise level of 0.70 for 39,000 iterations.

We logged our loss throughout training and noticed a spiking loss between both our models trained on 0.55 noise (Figure 3) and 0.70 noise (Figure 4). We used the Fréchet inception distance (FID) score of the reconstructed image with the original images (using a sample size of 500 images) to evaluate our model as seen in Figure 5 (Heusel, 2018). FID was chosen as the metric here because it imitates human perception of similarity.

Discussion

As shown in Figures 3, 4 and 5, the differences in loss between 0.55 and 0.70 noise are minimal. This indicates that the model is not affected significantly by the changes to the noise level in this instance. Furthermore, Figure 2 shows that the 0.70, and to a lesser extent the 0.55, noise models are able to generalize well to a noise level of 0.80. The model does not seem to struggle at any noise level presented and there is a minimal change in loss between the different noise levels. This leads us to the conclusion that this model could achieve good results at noise levels reaching into the 0.80 or potentially even 0.90 noise range.

Although increasing the noise levels leads to very good results, the model struggles when presented with images of a different noise level than it was trained on. This is likely because only the encoder was trained and that the encoder expects a specific level of noise for it to offset. Unfortunately, because the model does not generalize well to multiple noise levels, it is unlikely that we achieved the main goal of extracting the underlying structure of the data. If a model can be trained on a set range of noises, for instance from .40 to .70, and still work on other noise levels it would be more general. For now, it seems our network is able to determine the weight of the noise on the image and extract the result as opposed to detecting the style of the image itself.

Another feature we found was that the dataset we trained on was large enough that we never got through our dataset and therefore never completed an epoch. This might lead to the loss being higher and more erratic than if the model trained on a smaller set of data repeatedly. A further problem with the dataset is the lack of good standardization. While all the images were cropped and aligned, many were not front facing, included stray lines at the top, and had articles of clothing such as hats or glasses that might throw off the model.

Additionally, one major issue that needs to be resolved is that at higher noise levels there is a loss of information in areas where there are few pixels such as the eyes. This leads to results resembling mode collapse as the model does not have enough information to make a prediction on these areas. One potential solution to this would be to use multiple potential outputs similar to the super resolution model from the pSp framework (Richardson 2020).

One point of negligence on our part was training, testing, and validating on the dataset used in the pretraining of the pSp model. Specifically, the testing and validation datasets were tainted by the pre-trained weights. Meaning we cannot assume without doubt that this model will generalize outside of the ffhq dataset. The decision to use the same dataset from the training was made to minimize the amount of time required to train our model. However, in the future, a different dataset from the pretraining should be used for the testing and validation.

There are many potential sources of bias revolving around the generation of images our StyleGAN produces. One such bias is related to the brightness of the original image. Our current method of noise generation averages the RGB values of a completely random layer of generated noise with the target image to get the input. One source of bias that this creates is that on average, colors with values nearer to the middle of the 255 value representation of the pixel color (such as a pixel with a value of 128, 128, 128) will be on average affected less than those pixels nearer a value of zero or 255. This means there might be a bias in the effect of the noise on images with lighter and darker tones. This means the model might perform better on the neutral tones. Another potential source of bias could be coming from the dataset used to train the model. As the dataset is sourced from images of celebrities, there could be a lack of diversity in terms of facial features. This creates a model skewed towards a certain set of features, presenting less accurate results when presented with derivations from the norm in facial shape, age, skin and hair color, and facial symmetry.

Due to the nature of GANs, in addition to the complex style vector we use to generate the image with our pixel2style2pixel model, this problem will be quite complex to tackle. It will require more research into how to change various aspects of how the GAN generates its images in relation to the given style vectors. Furthermore, there could be some potential outliers in our results because there are subgroups of images with distinct facial features, such as children, which might skew error metrics. This is possible because as the facial features of children are much more scaled down in relation to those of adults, we may see generated images that struggle to differentiate between adults and children and thus be stuck as to how to deal with the "styled" latent vector in that case.

Conclusion

Our work has demonstrated the efficacy of using style GANs with the pixel2style2pixel implementation to achieve white noise reduction at high noise levels through emulating the target as opposed to attempting to match the target exactly (Karras 2019). While such systems will most likely not be usable in cases where the accuracy of denoised data is paramount, our system and similarly designed systems can be very useful in scenarios where data accuracy is less important. For example, photos or data that rely on curves as opposed to distinct points such as instrumentation spectra would benefit from our pixel2style2pixel implementation of denoising (Nguyen 2020).

Furthermore, while the scope of our specific implementation is limited, the style matching technique could yield networks in which the scope could be much broader due to the universal nature of pattern in data that could be detected by a style matching system. We hope our work will lead to a wider adoption of GANs for problems resembling multi-step image-to-image translation. It would be fruitful to see if such a network could be used for more differing styles such as a more diverse data set of photos.

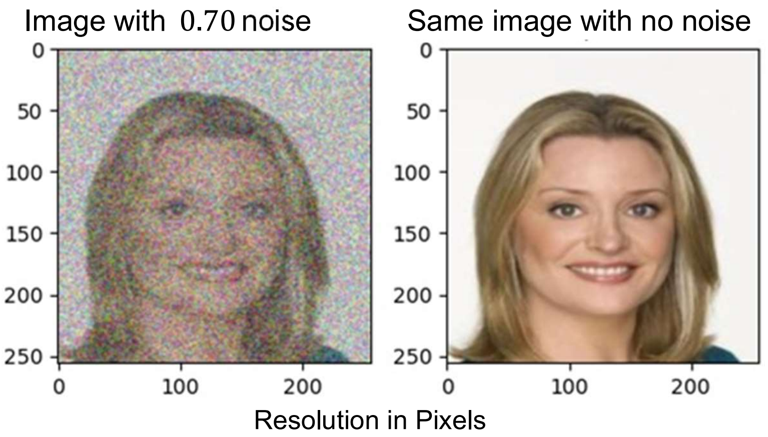


Figure 1, image after and before noise is added

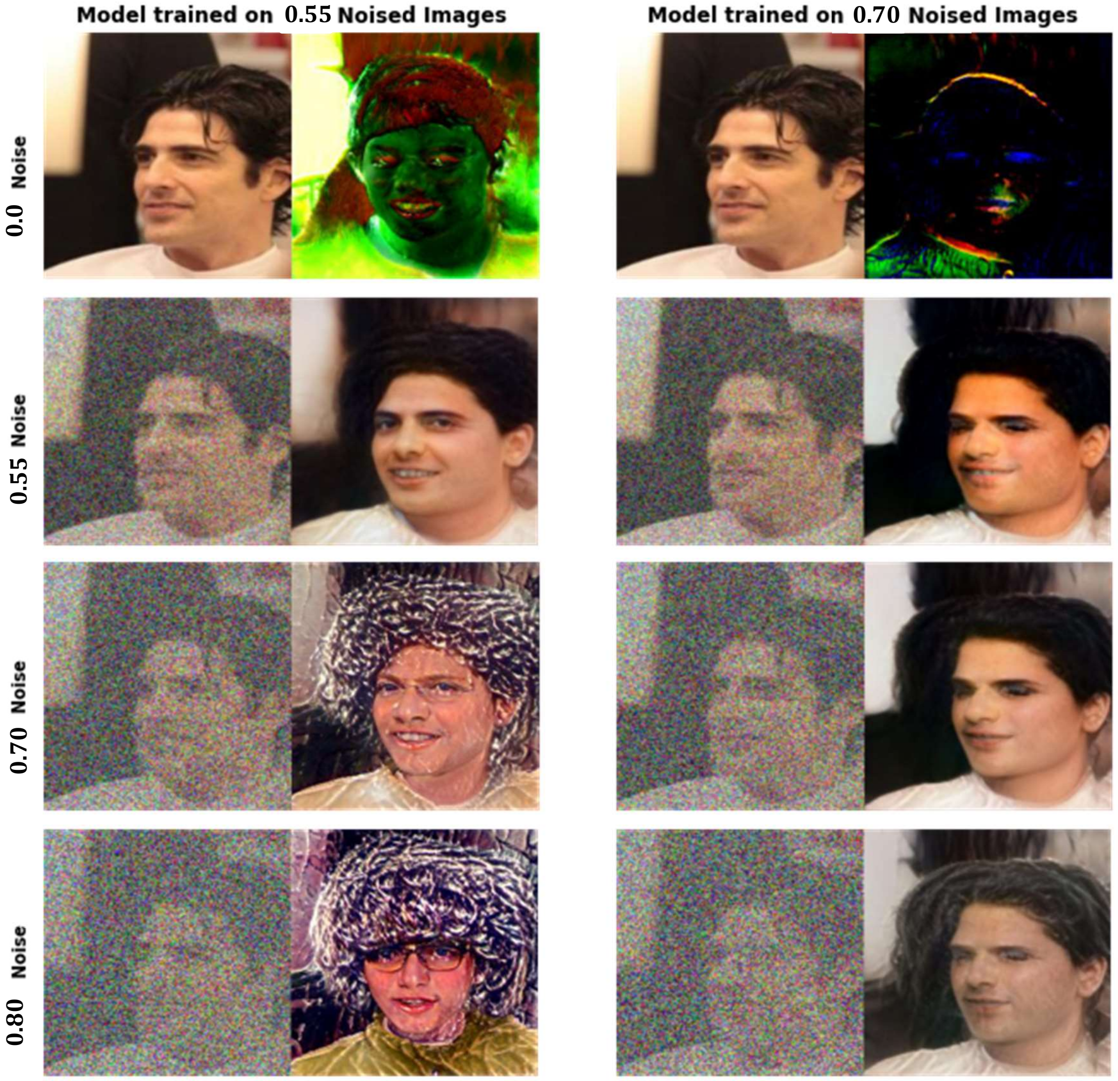

Model trained on 0.55 Noised Images
Model trained on 0.70 Noised Images
0.0 Noise
0.55 Noise
0.70 Noise
0.80 Noise

Figure 2, images at different noise levels ran through the 0.55 and 0.70 models

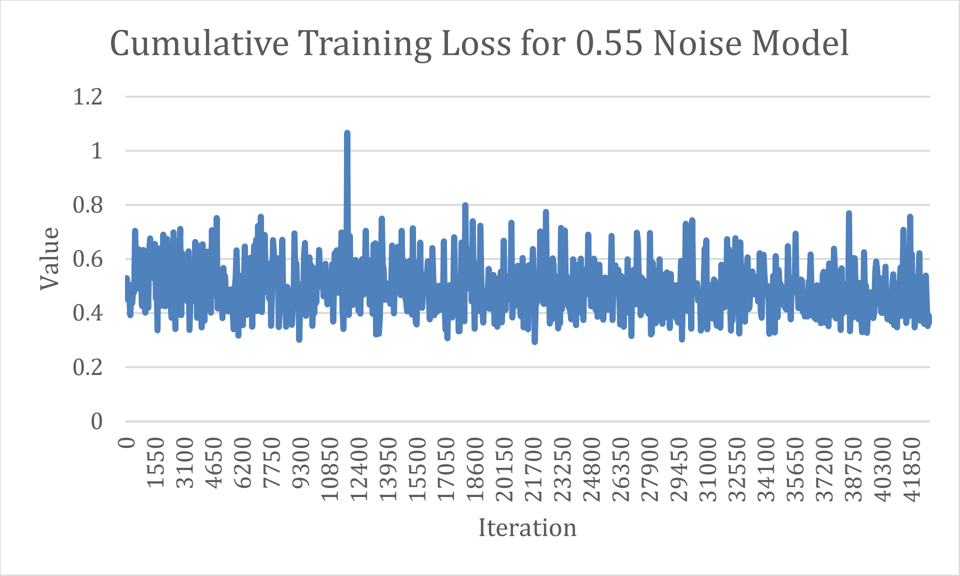


Figure 3, loss over iterations for the 0.55 model

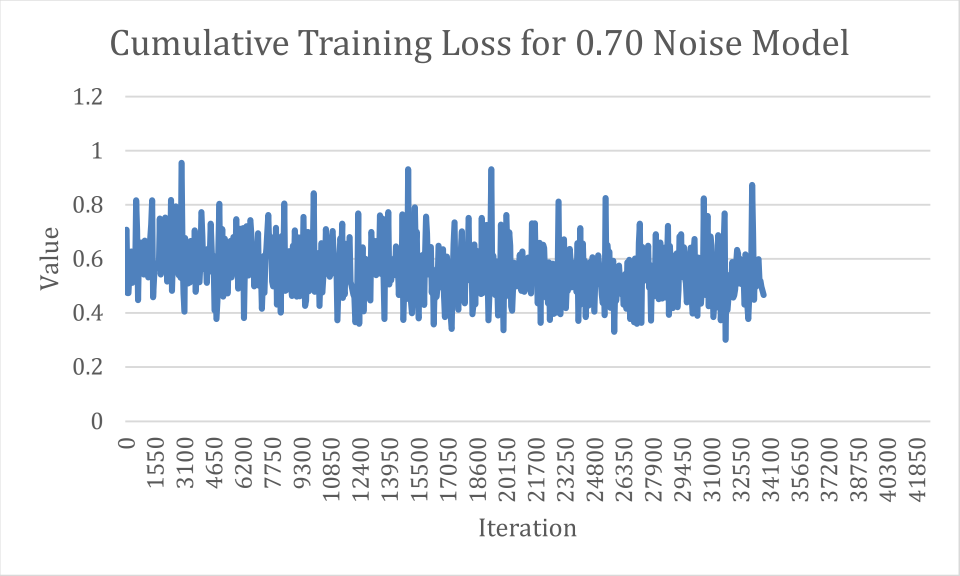
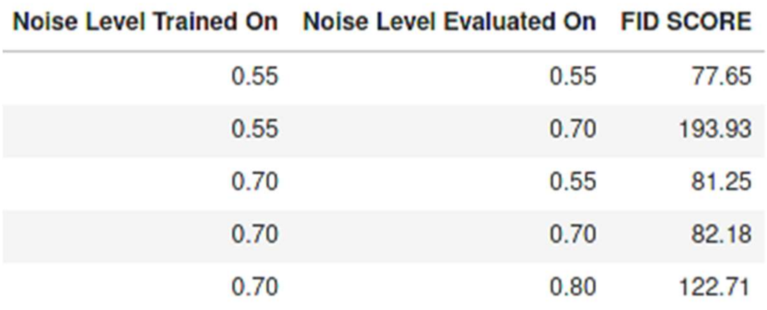


Figure 4, loss over iterations for the 0.70 model



Noise Level Trained On	Noise Level Evaluated On	FID SCORE
0.55	0.55	77.65
0.55	0.70	193.93
0.70	0.55	81.25
0.70	0.70	82.18
0.70	0.80	122.71

Figure 5, Fréchet Inception Distance for different modes evaluated at different noise levels

References

Bethge, J., Bartz, C., Yang, H., et al. (2020). One Model to Reconstruct them All: A Novel Way to Use the Stochastic Noise in StyleGAN. *arXiv.* https://arxiv.org/pdf/2010.11113v1.pdf

Gal, R., Cohen, D., Bermano, A., et al. (2021). SWAGAN: A Style-based Wavelet-driven Generative Model. *arXiv.* https://arxiv.org/pdf/2102.06108v1.pdf

Gautam, A., Sit, M., & Demir, I. (2020). Realistic River Image Synthesis using Deep Generative Adversarial Networks. *arXiv*. https://arxiv.org/abs/2003.00826

Guan, S., Tai, Y., Ni, B., et al. (2020). Collaborative Learning for Faster StyleGAN Embedding. *arXiv*. https://arxiv.org/pdf/2007.01758v1.pdf

Heusel M., Ramsauer H.., Unterthiner T.., et al. (2018). GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. *arXiv.* https://arxiv.org/pdf/1706.08500.pdf

Karras, T., Laine, S., Aittala, M., et al. (2019). A Style-Based Generator Architecture for Generative Adversarial Networks. *arXiv*. https://arxiv.org/pdf/1812.04948.pdf

Karras, T., Laine, S., Aittala, M., et al. (2020). Analyzing and Improving the Image Quality of StyleGAN. *arXiv*. https://arxiv.org/pdf/1912.04958.pdf

Liu, Ziwei., Luo, Ping., Wang, Xiaogang., Tang, Xiaoou., et al. (2015). Deep Learning Face Attributes in the Wild. *Proceedings of International Conference on Computer Vision (ICCV).* http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

Nguyen, H., Yamagishi, J., Echizen, I., et al. (2020). Generating Master Faces for Use in Performing Wolf Attacks on Face Recognition Systems. *arXiv.* https://arxiv.org/pdf/2006.08376v1.pdf

Radford, A., Metz, L., Chintala, S. (2016). Unsupervised Representation Learning With Deep Convolutional Generative Adversarial Networks. *arXiv*. https://arxiv.org/pdf/1511.06434.pdf

Richardson, E., Alaluf, Y., Patashnik, O., et al. (2020). Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation. *arXiv.* https://arxiv.org/pdf/2008.00951v1.pdf

Tov, O., Alaluf, Y., Nitzan, Y., et al. (2021). Designing an Encoder for StyleGAN Image Manipulation. *arXiv.* https://arxiv.org/pdf/2102.02766v1.pdf