**Aging Simulation using CycleGAN**

**Abstract**

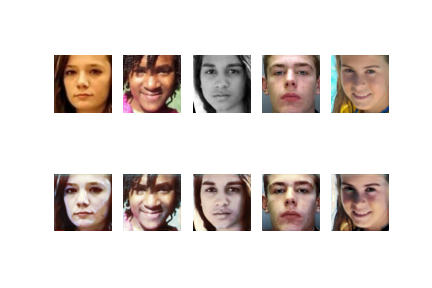
A generative adversarial network (GAN), is a type of machine learning that works by making two neural networks compete against each other. This type of artificial intelligence came from a paper published in 2014 making it a relatively new training method [11]. The neural networks compete in a zero-sum game, and through multiple runs each neural network gets better at their respective job. The two jobs are that of the generative network and the discriminative network. The generative network attempts to trick the discriminative network into making errors by generating data that increases the error rate. The discriminative network attempts to decrease the error rate, and as a result these two networks help each other improve leading to the discriminative network becoming better at doing its task.

In this project a GAN was trained for aging simulation. Aging simulation was selected because nearly every year there are hundreds of thousands of missing children reported in America and around the world. Children not found over long spans of time need have their age simulated. The current method of doing so is an age progression photo software called PhotoSketch. PhotoSketch requires a professional artist to age the photo. This creates a barrier to entry to use this sort of technology because law enforcement would need to have an artist on hand and the funding to maintain up-to-date aged photos for each missing child. It also requires that the artist be specifically trained in how to age people. In the United States the local law enforcement may have access to the resources and artists to do this occasional updates to missing persons images, however in developing countries that may not necessarily be the case. A few others have created GAN based face aging algorithms such as [1], [2], [5], and [6], but their neural networks are more for general face aging of the subject while we hope to have more accuracy with a limited initial age set, and final age target set.

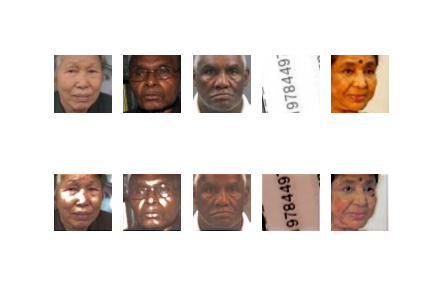
**Methods**

The GAN used in this project is one called CycleGAN, a GAN method for training AI created by Zhu et al. [7] to translate one image into another. In the original paper, the creator’s used it in a few ways to pull off image manipulations such as replacing horses with zebras in images. In this project we used images of young individuals and old individuals from the UTK face data set from the Zhang et al. [6]. We had to clean up the data, but some random images that were in the UTK dataset still made it in. In one of the early training rounds of the AI there will be a few of these random images that get aged, but overall there looks like there may be at most a few dozen random images, and about as many mislabeled images. After that data was cleaned up a bit the data was sorted. The 256x256 pixel images were sorted into young individuals and old individuals. The young individuals were people aged 10-20, and old individuals were people from 61-80. The data was further split in each data set into training and testing data. We used a 70-30 split of training to test data to train AI. After the AI was trained on the data, the ai would randomly age up 5 images in the data set of 10-20. Then there will be 5 images in the dataset of 61-80 that gets aged down. This ended up helping us to see just what the Ai was being trained to do in order to age the person or more generally change the target’s age.

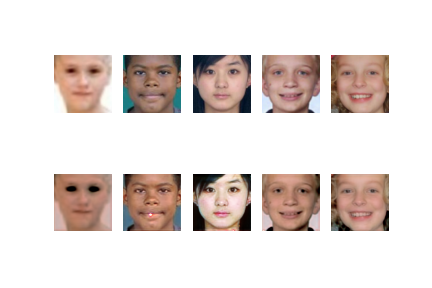
**Results (**early data**)**



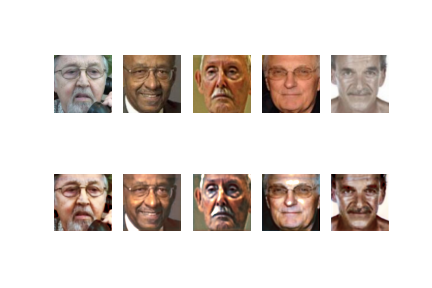
**Figure 1**) First run output: Young to Old



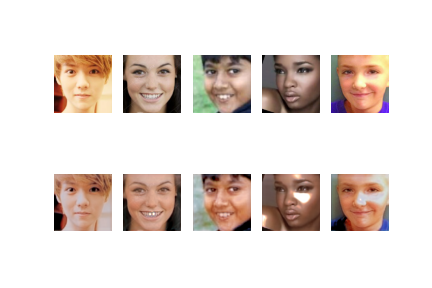
**Figure 2**) First run output: Old to Young



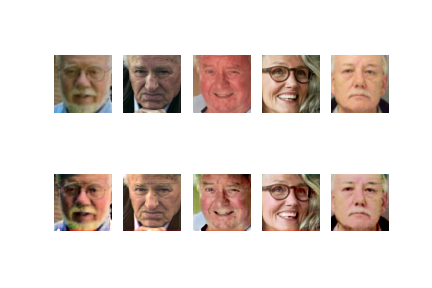
**Figure 3**) Second run output: Young to Old



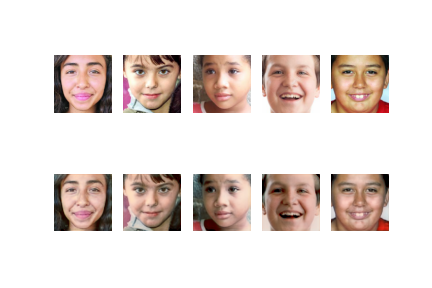
**Figure 4**) Second run output: Old to Young



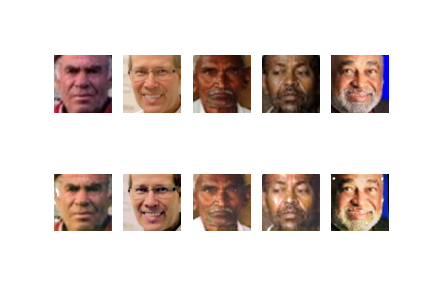
**Figure 5**) Third run output: Young to Old



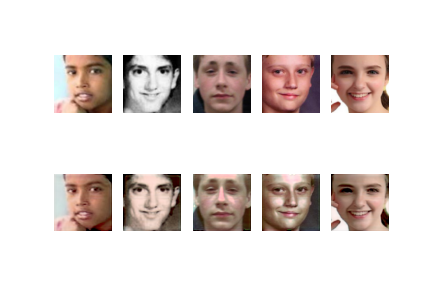
**Figure 6**) Third run output: Old to Young



**Figure 7**) Fourth run output: Young to Old



**Figure 8**) Fourth run output: Old to Young



**Figure 9**) Fifth run output: Young to Old



**Figure 10**) Fifth run output: Old to Young

**Conclusions**

After each run of training the program made an output of the images it selected to demonstrate on. The first output images are seen in fig. 1 and 2, where we see that the GAN seems to have trained initially to grey the people in the photo or make them more pale when they are being aged. It also seems like it’s removed color from the photos it ages up in fig. 1. In fig. 2 on the other hand, it does the opposite and adds more color when aging down. The program also seems to be focused on the shine on the person’s cheeks in the image increasing and decreasing it depending on whether it is aging up or down. This tells us at least in the first run of training the program thinks that paleness, cheek shine, and discoloration is one way to age up/down individuals in a photo. Then in the second run, fig. 3 and 4, the program seems to not stick with the discoloration, and instead begins applying that randomly adding color for some and making the images more pale for others when it is aging up. For fig. 4 the images are still getting their color intensity increased, but to a greater degree as is the case with the shine being applied to the cheeks of the subjects in the photos that are aged down.

In the third run, fig. 5 and 6, the young to old set of images in fig. 5 seems the same as fig. 3 with what looks like an attempt to remove the shine from the faces of the subjects. In fig. 6 however, we see relatively normal looking changes to the photos with nothing major except that the 3rd image looks like color was taken out of the face at random points that seems to make him look younger, and the fourth image looks like they lost a wrinkle on their forehead when aged down. The program also seems to be making changes to the background of the images, such as image 3 in figure 6, which means that in the future we’d need to process the image to possibly remove the background in order to ensure that the only learning being done, and changes being made are in relation to the subject’s face. Run 4 led to less paleness in the images going from young to old, fig. 7, as well as the addition of wrinkles more accurately placed on the faces of the subjects. In fig. 8 one of the subjects, in column 2, gets their hair recolored to a darker color to make them look younger. The rest of the images in fig. 9 and 10 from run 5 don’t change much from the runs before the changes they make.

There were a few issues in the images generated arising from random images being placed in the data set, but overall it only showed up in one run’s example output out of the thousands of pictures in the set. With the Ai taking a few hours to run on Google Colab it limits the amount of times it can be trained, but with more time the results will probably keep improving. However, there is a danger of overfitting in our data because we didn’t use one of the methods used by others like Zhu et al [7], where they use methods like mirroring the images jittering, where the image is expanded then randomly cropped. These methods can help reduce overfitting, but with the time limits on Colab’s runtime, the extra time it would take to process the images would take away from time it could be training off of them. In the future if the program is given more time to train then the overfitting counter measures will likely be necessary, but in our case due to the limited training runs it was not every likely that we’d see overfitting.

In the future it would likely lead to improved results if before the neural networks ran, another Ai went through the images to confirm that all the samples are human, and discarded those that aren’t. It could also be useful to let the GAN run using semi-supervised learning like suggested in [9] and demonstrated with GANs in [10]. Letting our GAN chuck back in newly generated faces into its own training set, that is after the generated samples are deemed good enough that could pass through an face identifier and age estimating neural network. Overall, the biggest gains in terms of accuracy will probably come from better sorted and sanitized data, tweaks in the initial image processing, and finally simply extending training time so that the GAN’s accuracy can improve.

**References**

[1] Grigory Antipov, Moez Baccouche, and Jean-Luc Dugelay. 2017. Face aging with conditional generative adversarial networks. In *2017 IEEE International Conference on Image Processing (ICIP)*, 2089–2093. DOI:<https://doi.org/10.1109/ICIP.2017.8296650>

[2] Xinhua Liu, Chengjuan Xie, Hailan Kuang, and Xiaolin Ma. 2018. Face Aging Simulation with Deep Convolutional Generative Adversarial Networks. In *2018 10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, 220–224. DOI:<https://doi.org/10.1109/ICMTMA.2018.00060>

[3] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. 2017. Learning from Simulated and Unsupervised Images through Adversarial Training. *arXiv:1612.07828 [cs]* (July 2017). Retrieved October 30, 2020 from<http://arxiv.org/abs/1612.07828>

[4] Lipeng Wan, Jun Wan, Yi Jin, Zichang Tan, and Stan Z. Li. 2018. Fine-Grained Multi-Attribute Adversarial Learning for Face Generation of Age, Gender and Ethnicity. In *2018 International Conference on Biometrics (ICB)*, 98–103. DOI:<https://doi.org/10.1109/ICB2018.2018.00025>

[5]Chun Yang and Zhihan Lv. 2020. Gender based face aging with cycle-consistent adversarial networks. *Image and Vision Computing* 100, (August 2020), 103945. DOI:<https://doi.org/10.1016/j.imavis.2020.103945>

[6] Zhifei Zhang, Yang Song, and Hairong Qi. 2017. Age Progression/Regression by Conditional Adversarial Autoencoder. arXiv:1702.08423 [cs] (March 2017). Retrieved October 30, 2020 from http://arxiv.org/abs/1702.08423

[7] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2020. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *arXiv:1703.10593 [cs]* (August 2020). Retrieved October 30, 2020 from<http://arxiv.org/abs/1703.10593>

[8] I. Sutskever, R. Jozefowicz, K. Gregor, D. Rezende, T. Lillicrap, and O. Vinyals, “Towards Principled Unsupervised Learning,” *arXiv:1511.06440 [cs]*, Dec. 2015, Accessed: Nov. 4, 2020. [Online]. Available:<http://arxiv.org/abs/1511.06440>.

[9] J. T. Springenberg, “Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks,” *arXiv:1511.06390 [cs, stat]*, Apr. 2016, Accessed: Nov. 4, 2020. [Online]. Available:<http://arxiv.org/abs/1511.06390>.

[10] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved Techniques for Training GANs,” *arXiv:1606.03498 [cs]*, Jun. 2016, Accessed: Nov. 4, 2020. [Online]. Available:<http://arxiv.org/abs/1606.03498>.

[11] I. Goodfellow and F. Wilczek, “Supervised Learning of Probability Distributions by Neural Networks,” in *Neural Information Processing Systems*, 2014, pp. 52–61, Accessed: Nov. 4, 2020. [Online]. Available:<https://proceedings.neurips.cc/paper/1987/file/eccbc87e4b5ce2fe28308fd9f2a7baf3-Paper.pdf>.