Steering HoloGAN

Kaustubh Olpadkar (SBU ID: 112584724)
Graduate Student, Computer Science, Stony Brook University, New York, United States

***Abstract*—Recent advancements in GANs[1] have shown ability to produce promising results in the 3-D domain and to control different aspects of generated images by manipulations in the latent space. The recent papers on HoloGAN[2] and the GAN steerability show the approach to control the pose as well as learn the walks in latent space to control various parameters of the generated image in a fully self-supervised manner. The main idea of this project is to apply the steerability[3] on the HoloGAN and to edit the human facial expressions using it. We train the HoloGAN to generate human faces with different poses. We examine the pose invariance in HoloGAN and utilize the steerability for editing the facial expressions. We use the action unit regressor as well as the pre-trained facial embedding in this part. We design the novel loss function which can measure the difference between the desired expressions as well as preserve the identity. This facial expression editing technique can also be applied to any other GAN.**

***Index Terms*—Generative Adversarial Networks, Facial Expression Editing**

I. Introduction

Traditionally, GANs have been used to generate 2-D images. Further research in GANs allowed controlling the class of the generated image. The GANs are also widely used in the image-to-image translation tasks. Recent advancements in GANs have allowed the GANs to produce promising results in the 3-D domain. And, a lot of works have shown to control different aspects of generated images by manipulations in the latent space of GAN. The majority of the previous works in these domains were supervised and needed huge amount of annotated data. But, the papers, 'HoloGAN' and 'GAN Steerability', which we will use for the project, control the pose as well as learn the walks in latent space to control different parameters of the generated image in a fully self-supervised manner, which is a notable improvement over the previous works. In this project, we apply the steerability on the HoloGAN to edit the human facial expressions using the steerability. We train the HoloGAN to generate human faces and examine the pose invariance. We apply the steerability for editing the facial expressions. We design the novel loss function which measures the difference between the desired expressions using the action units as well as preserve the identity using facial embedding. For this, we trained the action unit regressor and used the pre-trained differential facial embedding.

II. Method

A. HoloGAN

First, we implemented the HoloGAN from scratch in Pytorch[4] framework. The original code open-sourced by the authors is in Tensorflow[5] framework, which uses the older deprecated version of the framework. For the compatibility and flexibility we decided to use the Pytorch framework which comes with optimized GPU support and it is widely used amongst the deep learning researchers. We trained our HoloGAN on 1 GPU NVIDIA for 50 epochs which took around 52 hours for training with batch size of 32 and learning rate of 0.0005 for generator and discriminator both as per the guidelines in the HoloGAN paper. We trained the HoloGAN on the CelebA[6] dataset, which contains 200K celebrity images, each with 40 attribute annotations. While training, we allowed the pose to vary in range of -45 to +45 degrees in all 3 directions.

B. Action Unit Regressor

We trained the Action Unit Regressor[7] on the EmotioNet[8] dataset, which is a huge facial dataset with 950,000 images available with variety of expressions and annotated action units. We fine-tuned the Res-Net[9] architecture by adding one fully connected layer to the end to predict the intensities of the Action Units for 6 epochs with learning rate of 0.001 until the convergence with mean squared error as loss function and stochastic gradient descent optimizer. We use 17 Action Units and train a regressor to predict the intensities. The action units have intensities in range of 0 to 5, which we rescale during the training. We use this regression model to help predict the similarity in the generated face and edited face while steering the GAN to edit facial expression.

C. Facial Embedding

We use the pre-trained facial identity embedding from CMU Openface[10], which is differential network architecture. We use this in our loss function to measure the similarities between the similarity between the original and edited face images. Novel Loss Function:

D. Novel Loss Function

In the original GAN steerability paper examines the basic transformations like zooming, shearing and brightness change. The above transformations can be achieved in a straightforward way by basic image editing library. While training, they directly edit the generated image to apply the desired transformation. But, in our case, where we want to edit the facial expressions, it is not directly possible to edit generated face. Thus, we create our own loss function which can help us edit the facial expression.

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{emb} + \mathcal{L}_{aur} \\ \mathcal{L}_{emb} &= L2(EMB(G(z)),\ EMB(G(z + \alpha \cdot w))) \\ \mathcal{L}_{aur} &= L2(\alpha,\ AUR(G(z + \alpha \cdot w))) \end{aligned} \tag{1}$$

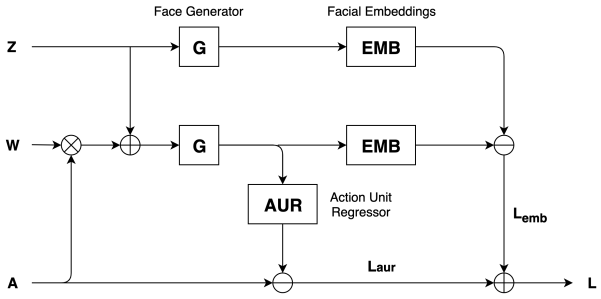

Fig. 1. Loss Function for editing Facial Expression

We depict the design of our loss function in the Figure1. We utilize the Action Unit regressor to measure the intensities of the action units for the generated image and compare that with the set of action unit intensities for the desired facial expression. We can reduce the difference between both using mean squared error as a loss function. But a problem that can arise during this process is that the generator may change the whole identity while changing the expression. To avoid that, we use the pre-trained facial identity embedding from the CMU Openface. We compute the difference between the facial embedding of the image to be edited and the image generated after applying a walk. Minimizing this error can help us to be sure that the identity is unchanged. Thus, we use the sum of both the embedding error and the action unit regressor error to define final loss function to edit the facial expressions. Minimizing this loss function will let us find a walk in latent space to edit the desired facial expression.

III. Experiments

We trained the HoloGAN on the CelebA dataset. We also train the Action Unit Regressor on EmotioNet dataset. We also used Progressive GAN trained on CelebA to show effectiveness of our approach for the facial expression editing on the GANs other than HoloGAN. We learn a latent space walks by leveraging the trained action unit regressor and pre-trained facial embedding, and further quantify how these walks affect various faces while editing the expression. We focus on linear walks in latent space for this project. We used Google Colab environment for learning different walks for editing expressions, which comes with specification of 1xTesla K80 GPU, having 2496 CUDA cores, compute 3.7, and 12GB GDDR5 VRAM. The training took around 1 hour to learn a single walk with batch size of 8 and learning rate of 0.001 for Adam optimizer with total 80000 samples during training.

First, We examined the pose invariance in the images generated with the HoloGAN. We fixed the identity and generated the images for different poses. From the resulting images shown in appendix we can see that we are able to preserve the identity when we change the 3D pose. Thus, it supports the correctness of our HoloGAN implementation. But, the quality of the faces generated by our HoloGAN is not very good. The generated images tend to have some noise in those. The results of HoloGAN are shown in the appendix. Thus, for expression editing we used the pretrained Progressive GAN model on CelebA dataset.

While experimenting with the Facial Expression editing, we experimented with different approaches. For some experiments we kept the one of the action units variable and others at neutral intensity. This helped us understand the effect of only one action unit while editing the image. We saw that it is not able to edit some action units which are having very low intensities in the images of training data. The large portion of training data is concentrated with few action units with high intensities.

We also tried to control a fixed emotion rather than individual action unit. For example, we experimented with the sad expression and happy expressions which have specific action units associated with them. So, we tried to achieve high intensities for those expressions while training. We can see that we are able to achieve the smile expressions in most of the images and sad expression in several images. This can be result of having less number of sad images in the training data.

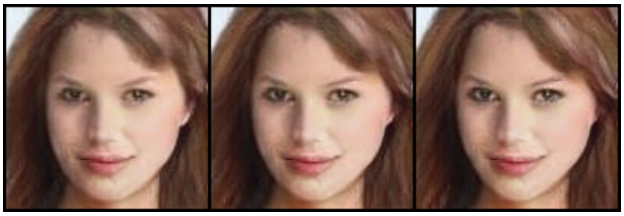

Fig. 2. Steering Smile Expression in range 0 to 1

Fig. 3. Steering Smile Expression in range 0 to 1

Fig. 4. Very less change in Expression in range 0 to 1

Fig. 5. Very less change in Expression in range 0 to 1

One interesting result we found was that after training a walk, we can change the controlling probe even out of the range of 0 to 1. We found that we can increase the controlling intensity alpha up to the value 4 in most of the cases. We also achieve interesting findings in the negative ranges of alpha. We could also found the range of expression editing in range of -10 to -5 intensities of alpha. Changing the range more from -10 to +10 results in changing the identity for most cases as expected. From these results, we conclude that the amount of expression editing we can achieve is related to the dataset variability. We find that we are able to achieve editing the expression of smile effectively, but expressions like sadness and anger are not achieved till the extent at it becomes noticeable. We get such results because the dataset we use does not contain huge number of images which relate to the expressions like anger and sadness, however it has lots of smiling and laughing faces. We conclude that this bias in the data limits us from achieving the ability to edit certain type of expressions. This finding is consistent with the finding in GAN steerability paper that 'The extent of each transformation is limited, and we quantify a relationship between dataset variability and how much we can shift the model distribution'.

IV. Conclusion

HoloGAN is a powerful generative model which can learn the 3D representation of objects and using the concept of Steerability we can achieve control over certain aspects of the generated images. Can we use the steerability to edit the facial expressions and apply steerability on the the GANs to achieve that? We investigate this question in this project by designing the novel loss function to edit the expressions. We use the Action-Unit regressor and pre-trained differential facial embedding which help us to minimize the loss function. We see that we are able to edit certain expressions like smile accurately but other few expressions we are not able to edit with high intensity. We find that, we can edit a few expressions to some degree but cannot extrapolate entirely outside the range of the training data. Our experiments illustrate that we can edit certain set of expressions using steerability by the loss function we designed and estimate that incorporating the training data with wide variety of expressions can help to edit more facial expressions.

Acknowledgment

We would like to thank Mr. Shahrukh Athar for the supervision and productive discussions during the project. We acknowledge the guidance and support from Dr. Dimitris Samaras throughout the project. We appreciate the support from the Department of Computer Science, Stony Brook University for providing the hardware for computation purposes.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[2] T. Nguyen-Phuoc, C. Li, L. Theis, C. Richardt, and Y.-L. Yang, "Hologan: Unsupervised learning of 3d representations from natural images," *arXiv preprint arXiv:1904.01326*, 2019.

[3] A. Jahanian, L. Chai, and P. Isola, "On the"steerability" of generative adversarial networks," *arXiv preprint arXiv:1907.07171*, 2019.

[4] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[5] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[6] Z. Liu, P. Luo, X. Wang, and X. Tang, "Large-scale celebfaces attributes (celeba) dataset," *Retrieved August*, vol. 15, p. 2018, 2018.

[7] J. J. Lien, T. Kanade, J. F. Cohn, and C.-C. Li, "Automated facial expression recognition based on facs action units," in *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, IEEE, 1998, pp. 390–395.

[8] C. F. Benitez-Quiroz, R. Srinivasan, Q. Feng, Y. Wang, and A. M. Martinez, "Emotionet challenge: Recognition of facial expressions of emotion in the wild," *arXiv preprint arXiv:1703.01210*, 2017.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[10] B. Amos, B. Ludwiczuk, M. Satyanarayanan, *et al.*, "Openface: A general-purpose face recognition library with mobile applications," *CMU School of Computer Science*, vol. 6, 2016.

Appendix

Additional experimental results are shown below.

Fig. 6. Change in Expressions in range -5 to 5 of alpha

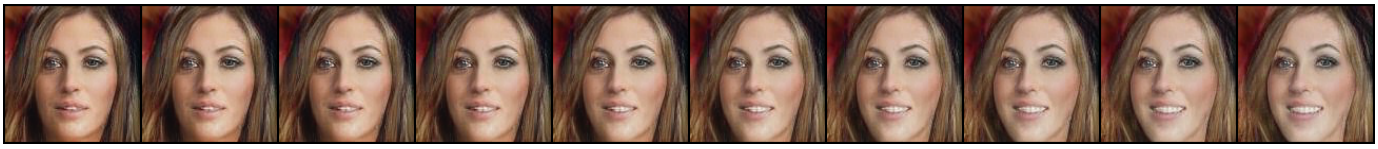

Fig. 7. Change in Expressions in range -5 to 5 of alpha

Fig. 8. Change in Expressions in range -5 to 1 of alpha

Fig. 9. Change in Expressions in range -10 to 2 of alpha

Fig. 10. Change in Expressions in range -10 to 2 of alpha

Fig. 11. Change in Expressions in range -10 to 2 of alpha

Fig. 12. Change in Expressions in range -10 to 5 of alpha

Fig. 13. Examining the Pose Invariance for fixed identity in HoloGAN

Fig. 14. Examining the Pose Invariance for fixed identity in HoloGAN

Fig. 15. Batch of faces generated by HoloGAN

Fig. 16. Batch of faces generated by HoloGAN