**Anime Character Generation**

Team Member: Yuming Hu, Qi Le

**Solutions Attempted:**

We choose WGAN-div as our basic model. WGAN-div uses Wasserstein divergence as its loss function. Comparing the 64\*64 resolution pictures generated from WGAN-div [Appendix A, Fig 1] to the 128\*128 resolution pictures generated from WGAN-div[Appendix A, Fig 2], we could find a little improvement in the imaging quality. But in general, the imaging quality is not very good.

**Challenges Faced:**

At first, our team wanted to implement the StyleGAN2 by ourselves after learning the source code posted by NVIDIA. Then we realized that it was not applicable for two reasons. The first reason is that the project itself is very complicated and we do not have enough optimization knowledge for reproducing the project. The second reason is that the Anime Character Generation might not need the high-resolution picture to show the details of the faces. Thus, we decided to implement a simpler model first and then add some modules that we learned from the StyleGAN2 project to our model to increase the imaging quality.

We selected WGAN-div as our basic model from DCGAN, WGAN, WGAN-GP, LSGAN, SNGAN. Compared to DCGAN, WGAN-div has a better loss function – Wasserstein divergence, which is a smooth function. But the Wasserstein divergence is hard to understand. [列公式，讲讲好处难处之类的。]

**Next Steps:**

The goal of our project is to improve the quality of the generated images. By now, there are two ways in front of us. The first way is to improve the resolution of the input and generated pictures. With the increasing resolution of the pictures, more and more details would show up on the generated pictures. The second way is the main focus of our group. We are trying to add some modules learned from the StyleGAN2 to our basic WGAN-div models. StyleGAN2 has 4 main changes in the new style-based generator:

1. Removing traditional input. [Appendix B. Fig 1]

2. Mapping Network. [Appendix B. Fig 2]

3. Style modules (AdaIN). [Appendix B. Fig 3]

4. Stochastic variation (Stochastic variation, generates random details for the generator by adding noise). [Appendix B. Fig 4]

Our group wants to add the first three parts to the WGAN-div model in the next steps and see how the generated pictures would change.

**Expected Outcome:**

We expect our generated pictures to have more details and clearer parts on the faces.



*Figure 1. Pictures generated from StyleGAN2 model*

We hope that our generated pictures could be close to the pictured generated from the StyleGAN2 model above.

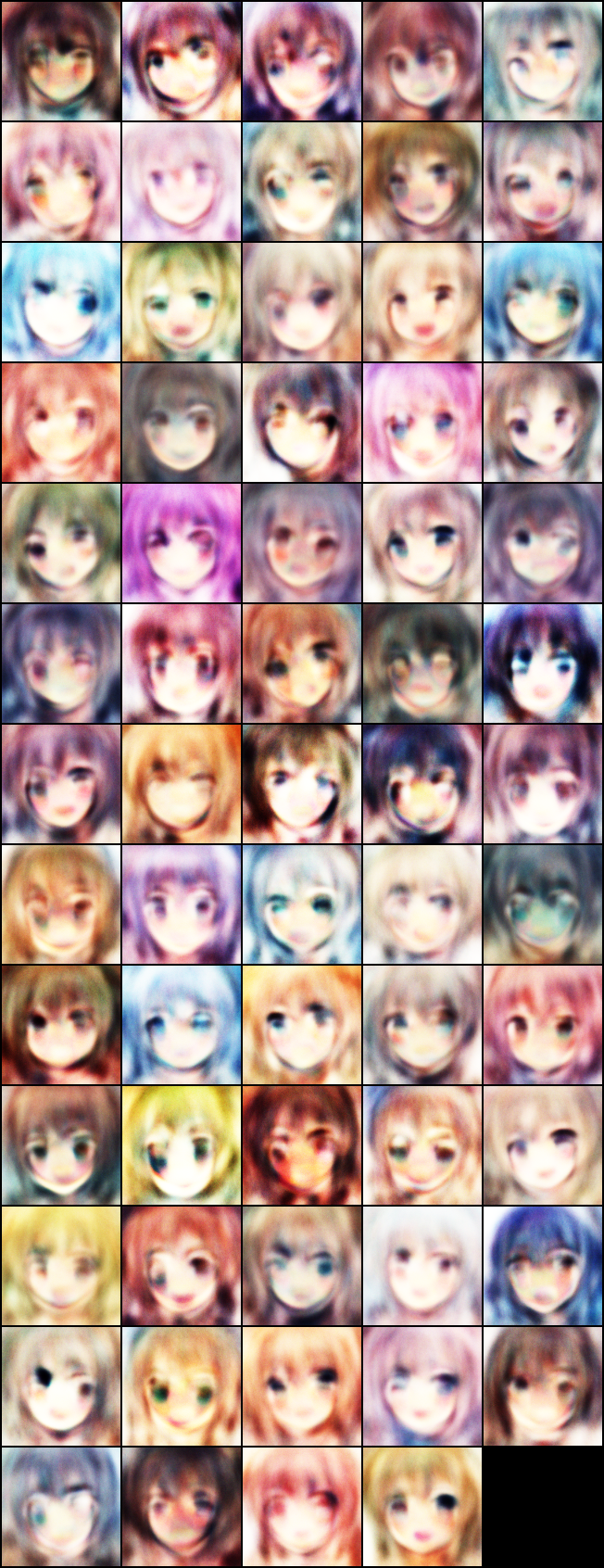
**Reference**

1. Rani Horev, *Style-based GANs – Generating and Tuning Realistic Artificial Faces,* <https://www.lyrn.ai/2018/12/26/a-style-based-generator-architecture-for-generative-adversarial-networks/>, 2018
2. Tero Karras, Samuli Laine, Timo Aila, *A Style-Based Generator Architecture for Generative Adversarial Networks*, <https://arxiv.org/abs/1812.04948>, CVPR 2019

**Appendix A**

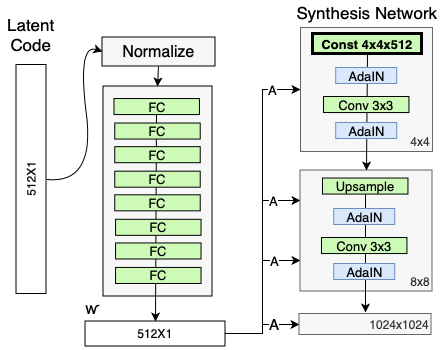


*Figure 1. 64\*64 resolution pictures generated from the WGAN-div*

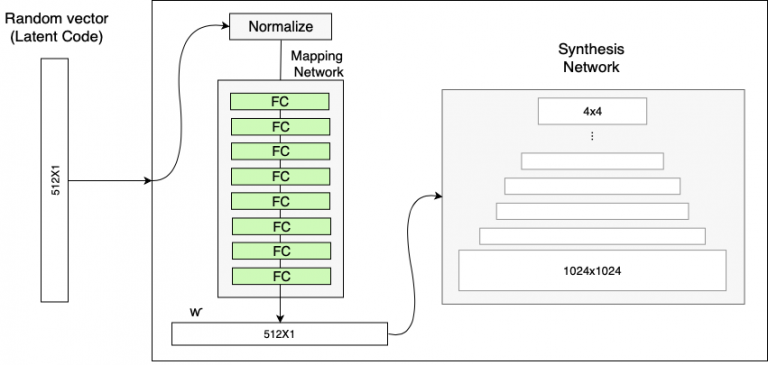


*Figure 1. 128\*128 resolution pictures generated from the WGAN-div*

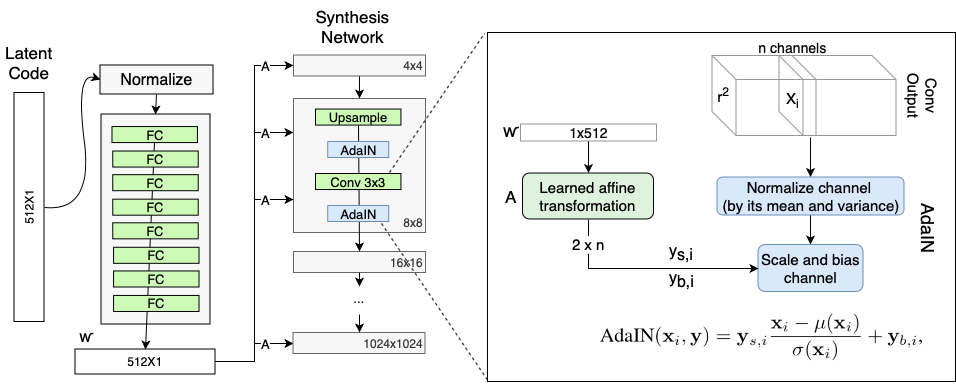
**Appendix B**



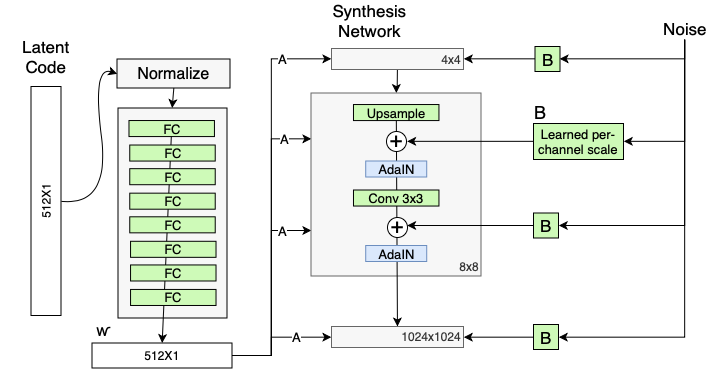
*Figure 1. The Synthesis Network input is replaced with a constant input*[1]



*Figure 2. The generator with the Mapping Network (in addition to the ProGAN synthesis network)* [1]



*Figure 3. The generator’s Adaptive Instance Normalization (AdaIN)* [1]



*Figure 4. Adding scaled noise to each resolution level of the synthesis network*[1]