Reproducing Star-GAN v2: Diverse Image Synthesis for Multiple Domains

Mohammad Bilal Arif, Madhur Sabherwal, Zimeng Wang, Yaqin Xiao

November 6, 2020

Abstract: This report chooses the paper, Star-GAN v2: Diverse Image Synthesis for Multiple Domains, to reproduce because of its high quality and code availability. The brief introduction and replication work will be presented first. Moreover, the report utilizes the pretrained model to test on new generated and processed dataset. Furthermore, the performance, evaluated by FID and LPIPS, will be discussed.

1 Source Paper Description

The criteria for choosing a research paper was whether it could be reproduced and applied to new dataset or not. Moreover, at the same time, the quality of the paper is also an important factor for choosing a reproduced paper. Based on above reasons, the paper, Star-GAN v2: Diverse Image Synthesis for Multiple Domains[1], has been chosen.

At present, the existing image-to-image translation model can only have limited diversity or multiple models for all fields. Thus, the chosen paper aims to utilize Star-GAN v2 to solve both problems, diversity of generated images and scalability over multiple domains, in one model[1]. To be more exactly, the paper performs experiments on CelebAHQ and a new animal faces dataset (AFHQ). Moreover, it has trained the model to generate different reaction pictures according to multiple fields and multiple styles like hairstyles, breeds and fur patterns for animals and so on. Furthermore, results have been compared with other three models (MUNT, DIR, MSGAN) and show the significantly improvement (Note that the results of other three models are cited by paper's authors from other papers).

1.1 Evaluation Framework

Two measurements, Frechet inception distance (FID) and Learned Perceptual Image Patch Similarity (LPIPS) have been presented to evaluate the performance of Star-GAN v2. To be more precise, the first one is to compute the distance between two distributions of real and generated images while the second one is to indicate the diversity of generated images[2]. Moreover, lower Fid and higher LPIPS mean the generated image is more closely to the real image and has higher diversity[3].

[1]Paper: https://arxiv.org/abs/1912.01865v2

Based on above two measurements, the performance has been compared with other three models (MUNT, DIR, MSGAN). To be more exactly, the FID score of Star-GAN v2 is 23.9 and 19.7 on above two mentioned datasets respectively, which are the smallest compared with the previous leading method[1]. In addition, the LPIPS score of Star-GAN v2 is 0.388 and 0.432 on above two mentioned datasets respectively, which are the highest compared with the previous three methods[1]. Therefore, Star-GAN v2 significantly improves the performance over the baselines.

1.2 Justification

The quality of the paper can be verified from verious rankings for example: CORE rankings or Google Scholar discipline rankings. For the justification of the chosen paper, it has been published by the Conference on Computer Vision and Pattern Recognition (CVPR) which is an annual major computer vision event such as main conferences, short courses and so on. Moreover, the most important is that CVPR is regarded as one of the most important conferences in its field and provides high-quality papers. In addition, the paper has been published in JANUARY 2020 but already has more than 56 citations in Google Scholar (see in Figure 1). Furthermore, the code for the paper is readily available provided by the researchers themselves on Github[2]

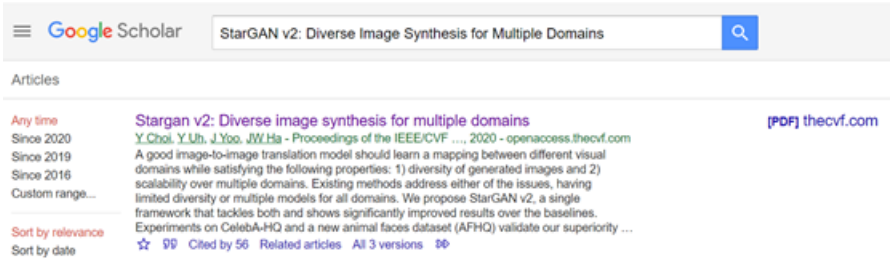

Figure 1: StarGAN V2 in Google Scholars

2 Description of original dataset

CelebAHQ[3] has up to 200k+ high quality pictures and has been separated into two domains (male and female) with different style like hairstyles, makeup and so on (in Figure2). Moreover, Animal Faces-HQ data set (AFHQ[4]) has 50000 high-resolution images in each of three areas (cat, dog, and wildlife) with various breeds. However,for the data set available and the memory of Google Colab, this report will focus on the first dataset which is free online. Moreover, the dataset is splited into test data set and train data set split based on 33% and 67% respectively. For a fair comparison, all images are resized to 256 × 256 resolution for training.

[2]Code Github: https://github.com/clovaai/stargan-v2
[3]CelebAHQ: https://drive.google.com/drive/folders/0B4qLcYyJmiz0TXY1NG02bzZVRGs
[4]AFHQ: https://www.reddit.com/r/datasets/comments/g8v3vo/animal$_f aceshq_d ataset_a fhq$/

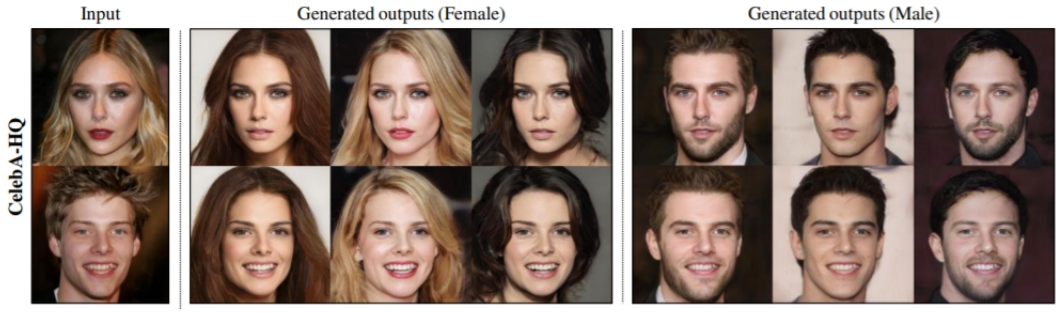

Figure 2: CelebAHQ dataset

3 Replication of original work

Exploration being replicable is firmly identified with the possibility of observational speculation. The instruments are made accessible to permit us copy the consequences of the research, possibly broaden it. Prior to doing as such in any case, there should be a decent arrangement that can redress and notice fundamental changes before hand. Our objective is to reproduce the StarGAN-V2 model and ensure it runs as said with the given data. The paper centers around contrasting the proposed model and standard models, our generation anyway doesn't do as such, no gauge models are repeated. The independent model is subjectively assessed dependent on human constancy of what high visual quality interpretations of picture looks like while keeping the source research paper as a standard measure. The evaluation scores generated from replication will be compared with the scores of the author thereby an accuracy percentage will be calculated to determine how accurate is our replication.

3.1 Issues of Original Work

The source code referenced in the paper works similarly as referenced in the repository read-me file. The code is written in python with dependencies mentioned below:

- PyTorch, torchvision, cudatoolkit
- ffmpeg
- opencv-python, scikit-image
- pillow, scipy, tqdm and munch

Command line contentions are expected to run the python documents with a rundown of characteristics that describes the details like typ of mode (train,sample,eval,align), image size, directory of the image (input) areas, directory of the output image and so on. Making a Jupyter Notebook on Google Colaboratory seemed fairly easier since

the commands were python codes and Unix commands. The accompanying issues faced are tended to as follows:

- Initially, we started with using Unix Command line via AWS. But we faced crashing issues since we needed to download large sized dataset and running interpolation commands which took a lot of time thereby crashing the command prompt.
- The versions of dependencies needed to install before executing the code had compatibility issues with the python version of Unix. Therefore, we switched to Google Colaboratory.
- The model took too long to run in generating the interpolation videos and images for verification. Considering the high quality images and that too in large number, it took us atleast an hour to run the interpolation command.
- In addition to the time issues faced, we faced memory allocation issues as well. Google Notebook also faced the crashing issue due to memory halt.

3.2 Resolving Issues

- The main driving factor was the dependencies/libraries installation in order to successfully execute the work. The versions quoted in the repository were compatible in the Google Colaboratory, all we needed to do was changing the python version to 3.6.7.
- The timing issues faced in the Google Colaboratory were resolved by changing the processor to Graphics Processing Unit (GPU) and increasing the memory from 12.5 gbs to 25 gbs
- To make the commands, written by the authors, executable, we had to install MINICONDA on our Google Colab notebook to run CONDA scripted commands.
- Making the data available and commands quoted by the author required a shell command to run that would download the dataset and execute the commands which we easily resolved.

3.3 Generating Interpolation Images and Videos

After downloading the pre-trained networks, output images are synthesized, reflecting diverse styles (e.g., hairstyle) of reference images. In the Figure[5], the first row

[5]Output image: https://ibb.co/rtdNL6G

in the picture is taken as the source images. The first column on the extreme left is take as reference images. The remaining images are generated by StarGAN-V2 in such as way that it has taken the facial structure of the source image (top row) and constructed the hair styles and makeup styles taken from the references images (first column) and integrated on the source image. Each intersection point of the matrix above depicts a different output of image, generated by StarGAN-V2 depending on the facial structure of the row number and makeup/hair style of the column number.

3.4 Evaluation Matrix

To evaluate StarGAN v2 using Fréchet Inception Distance (FID) and Learned Perceptual Image Patch Similarity (LPIPS), we ran the commands exactly as the author's. It should be mentioned here that the evaluation metrics are calculated using random latent vectors or reference images, both of which are selected by the seed number. In the paper, the average of values from 10 measurements using different seed numbers, are being reported.

Evaluation Criteria	Author's Score	Replicated Score	Accuracy Match (%)
FID(latent)	13.73	13.7605	99.78
LPIPS(latent)	0.4515	0.4512	99.93
FID(reference)	23.84	23.8791	99.87
FID(reference)	0.3880	0.3875	99.87

3.5 Male to Female or Female to Male Generated Images:

The below figure show that Model works perfectly fine in some image transformations generated by StarGAN-V2 but at some places it has not transformed the images well enough.

Figure 3: Image to Image Interpolation

4 Construction of new data

In order to verify the reproducibility of the paper, we constructed a new data set to test the model.

4.1 Data collection

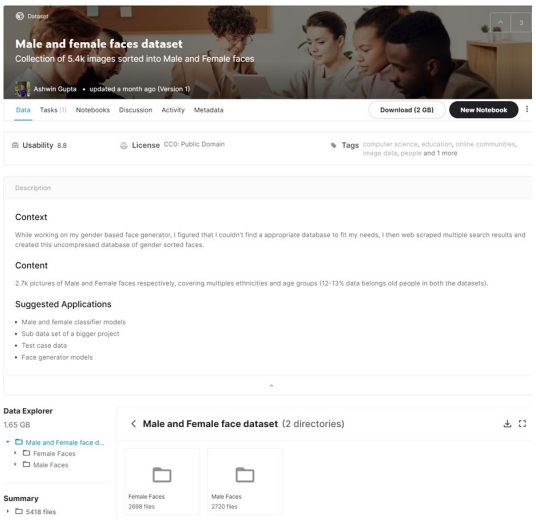

Figure 4: Some female faces with the problems of duplicate images and incomplete faces

The faces data set [6] is sorted into males and females, covering multiple ages and ethnicities. The author scraped multiple search results from the Internet and created this uncompressed database of gender sorted faces. There are 2698 images of female faces with a size of 0.97 GB; and 2720 images of male faces with a size of 0.692GB.

4.2 Data processing[7]

This is a raw data set and the number of images is satisfactory (we tried to capture 'female face' and 'male face' with google image API, but it was difficult to capture a sufficient number of images). However, there are still some problems compared with the data required by the model, such as duplicate images and incomplete faces.

4.2.1 Remove duplicate images

We use the Message-Digest Algorithm 5 to check whether two images are same. The Message-Digest Algorithm 5 is used to ensure complete and consistent information transmission. That is, if the MD5 hash value of two files are the same, it can be considered that the two files are same.

[6]The faces data set: https://www.kaggle.com/ashwingupta3012/male-and-female-faces-dataset
[7]https://github.com/xyq1996/COMP8240-GroupI

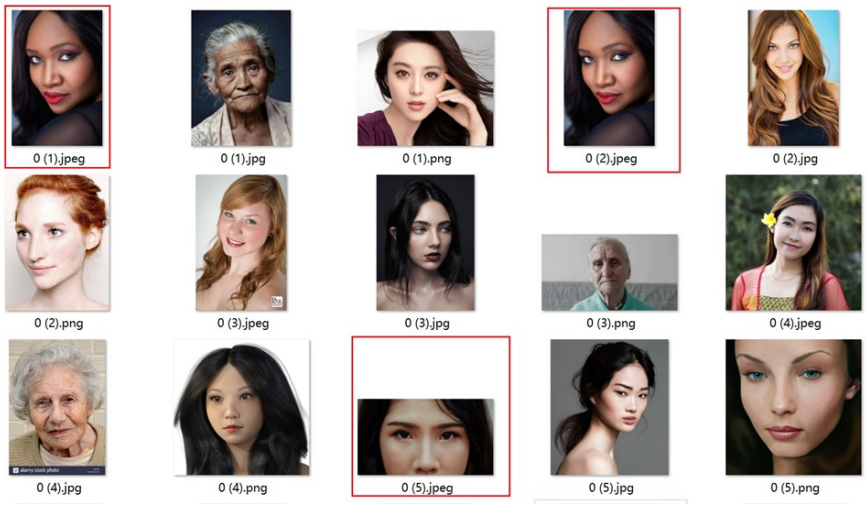

Figure 5: Some female faces with the problems of duplicate images and incomplete faces

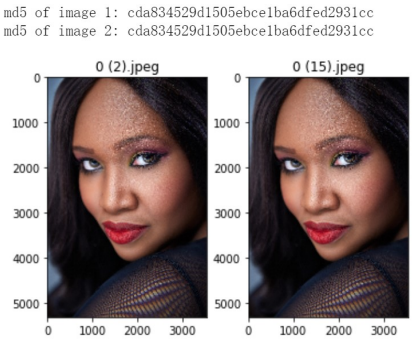

Figure 6: An example of the duplicate images in data set.

The 'hashlib'[8] module implements a common interface to many different secure hash and message digest algorithms, including the MD5. First, we generate a dictionary with the file name as the key and MD5 hash value as the value. Then, we check whether the MD5 hash value of the new file is in the value of the dictionary. If it is, it means this is a duplicate file so we delete it; if it is not, it means it is not a duplicate file so we keep this file and add the file name and the MD5 value to dictionary.

4.2.2 Remove similar images

By checking the data set processed in the previous step, we found that there are still some similar images. Because their md5 hash values are different, they are close, but not the same. The Hamming distance is the number of bit positions in which the two bits are different.

[8]Hashlib: https://docs.python.org/3/library/hashlib.html

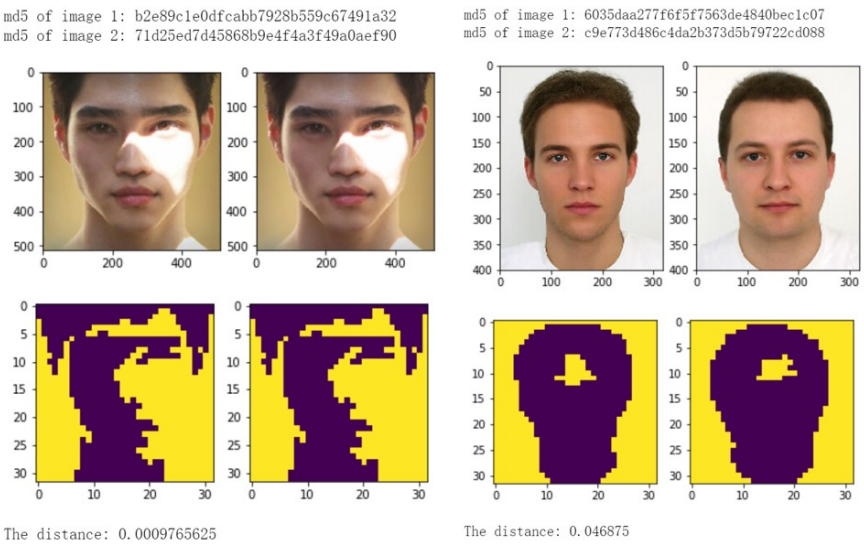

Figure 7: Two examples of similar images

To delete similar images, we do the following steps:

step1: Resize and grey scale image into 32*32

step2: Compute mean grey value, create an empty list to store hamming code

step3: Traversal every pixel and compare it with mean grey value

step4: Calculate the hamming distance between two lists.

For the step1, the image size is very important. If it is too large, it will take longer and more memory; if it is too small, some dissimilar images may be wrongly judged as similar. After doing some tests with similar images and dissimilar images, we chose to resize the image into 32*32, so that the generated list is 1024 bits, which can effectively compare the similarity of the two images in a relatively short time.

After calculating the distance of two images, it is important to determine the similarity of the two images. That is, when do we think the two pictures are similar? After testing some similar images, we choose the distances of 0.005 which means that if the Hamming distance is less than 0.005, we will consider the two images are similar.

4.2.3 Crop images

After removing duplicate images, our new data set still has some problems. For example, there is more than one face or only one incomplete face in an image. For these problems, we need to remove the image with an incomplete face and crop the image so that the proportion of face occupied in the whole is similar to that of CelebA-HQ.

Open-CV provides face orientation models[9]. Through cv2.CascadeClassifier.detectMultiScale()

[9]Open-CV: https://github.com/opencv/opencv/blob/master/data/haarcascades

function, we can identify the position of the face in the image. If there are multiple faces, multiple locations will be recognized; if it is an incomplete face or a side face, it may not be recognized (also what we want).

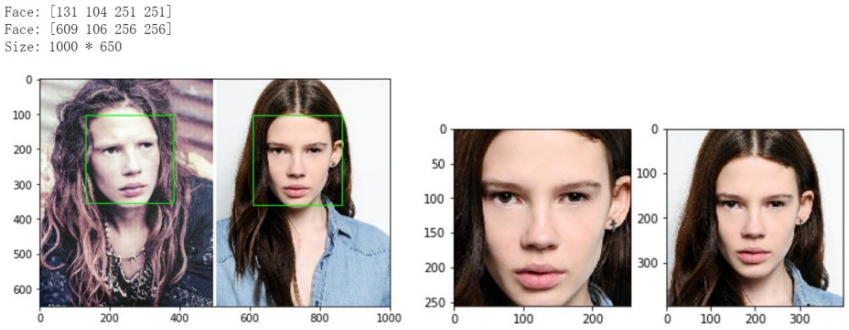

Figure 8: An example of an image with more than one face.

After testing some images in the original data set of the paper, the face occupies approximately 43% of the whole image. Therefore, we hope that the face in the new dataset will also occupy approximately 43% of the image.

After data processing, some images are 'non-face' images, which may be due to Face recognition error. Because the number of these 'non-face' images is small (only about five), we deleted them manually.

4.3 Summary of data processing

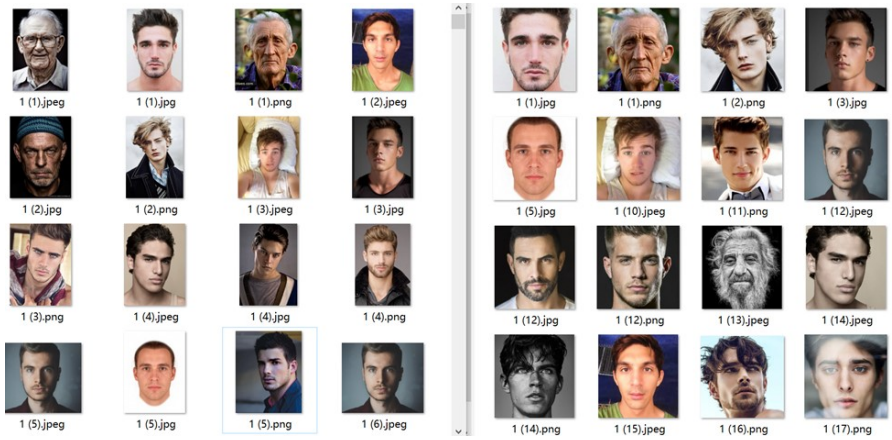

Figure 9: Some unprocessed images (left) and some processed images (right) in male faces data set.

Step1: Remove duplicate images. After step 1, the number of images in male faces data set is

reduced from 2720 to 966 and the number of images in female faces data set is reduced from 2698 to 735.

Step2: Remove similar images. After step 2, the number of images in male faces data set is reduced from 966 to 685 and the number of images in female faces data set is reduced from 735 to 625.

Step3: Crop images. After step 3, the number of images in male faces data set is reduced from 685 to 574 and the number of images in female faces data set is reduced from 625 to 562.

5 Results on new data

The new data set we proposed to implement with the STARGANv2, is crafted using the techniques mentioned above,Once the desired images were created we tends to test the model that if the model is working correctly as claimed by the model authors. we first did the batch prepossessing by adding a small images into the model and received the output, Although the results were in the range of which author proved however there were certain images which were not up to the mark and were quite noisy.

Annotation is one of the basic key point in the feature selection part for the model , as mentioned in the paper there were 40 attributes which were used to train the model. Each feature of the face was considered a attribute such as each hair color had its own attribute value while gender had binary values .Pre-annotating our new data set however, is one of our huge limitations in terms of time and resources.

Here is an example

Figure 10: male 2 female noise

5.1 Problems faced

We were having limitations while working on the hardware requirement of the the model.We found the there are some memory issues in Google colab. Since we have to increase the memory from 12gb to 25gb to run the whole code.

Having a deep understanding of our limitations, we notice that although the paper had some flaws,

the paper was reproducible at certain extent. Having noticed our limitations, we have Concluding on our empirical research, we do notice that replicating the work was possible . We initiated the project ahead and pre-processed 1700+ images. We have uploaded the new data set for running the pretrained model to test if the generated outcome images are correctly generated or not.

We kept the source images (the top row) same as with the original images in order to generate some similarity, only the reference images have been changed.The result

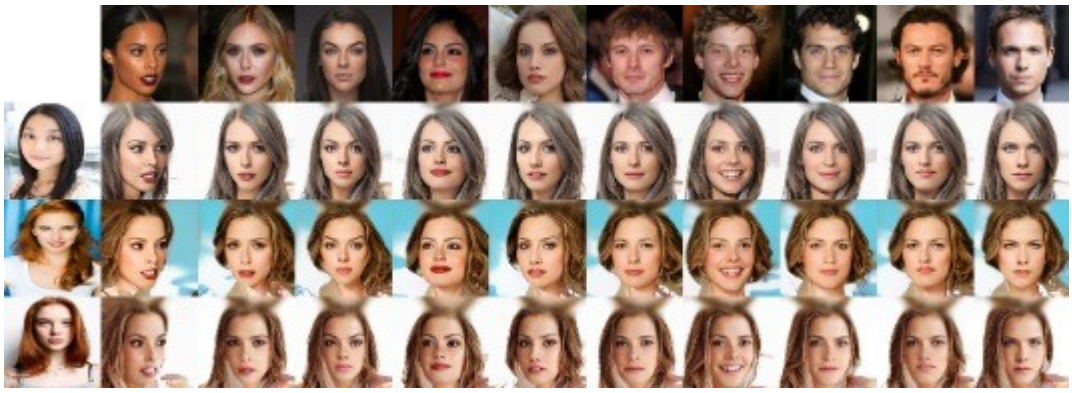

Figure 11: working image

We have put the whole image output on an image hosting website [10], We also have all the output stored in a gdrive [11], Our results have been put online on youtube link[12]

We evaluate both the visual quality and the diversity of generated images using Frechét inception distance (FID) and learned perceptual image patch similarity (LPIPS) . We compute FID and LPIPS for every pair of image domains within a dataset and report their average value

Table 1: Your first table.

Parameters	**Values**
FID Latent:	13.785027906581332
FID Reference :	23.850672602382335
LPIPS Latent:	0.45175284396111964
LPIPS Reference:	0.3884918097108603

5.2 Conlusions

We tested StarGAN v2, which have two major challenges in image to image translations in image-to-image translation; translating an image of one domain to diverse images of a target domain, and

[10]image output: https://ibb.co/XF9DN8d

[11]Output: https://drive.google.com/file/d/1Mp-yGx0E7IjR9K$_DPK5JbqoaB8wLOYiQ/view$

[12]Youtube: https://www.youtube.com/watch?v=4WfOw$_YI - nwfeature = youtu.be$

supporting multiple target domains. The experimental results showed that our model can generate images with rich styles across multiple domains,

5.3 Acknowledgements

We thank our A/Prof Mark Dras for letting us work on this project and guiding us throughout the project to work on the concepts. And our group Github is `https://github.com/xyq1996/COMP8240-GroupI`

References

[1] Y. Choi, Y. Uh, and J.-W. Yoo, Jaejunand Ha, "Stargan v2: Diverse image synthesis for multiple domains," pp. 8188–8197, 2020.

[2] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," Jan. 2018.

[3] P. Zhang, Richardand Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," Jan. 2018.