GROUP 32
PAIRED AND UNPAIRED IMAGE TO IMAGE TRANSLATION

Gaurav Kumar, Harpreet Singh*, and Soham Satyadharma**

University of California San Diego, La Jolla

ABSTRACT

Image-to-image translation is an area of active research in the field of Computer Vision that allows us to generate new images with different styles/texture/resolution while preserving the characteristic properties. Recent architectures use Generative Adversarial Networks (GANs) to transform input images from one domain to another. In this report, we worked on paired and unpaired image translation for multiple image domains. We used conditional GAN model for the paired task and trained it with cyclic consistency loss for the unpaired task, and experimented with different types of losses, multiple Patch GAN sizes, and model architectures. New quantitative metrics - precision, recall, and FID score are used for quantitative analysis. A further qualitative study of the results from different experiments is also carried out.

1. INTRODUCTION

Image to Image translation is a class of problems where images are transformed from one domain (style/design) to another, keeping its characteristic features intact. The eventual goal of such a system is to learn a mapping function between the input and the output domain. Recent advances in Generative Adversarial Networks (GANs) [1] have been successfully deployed to solve this problem. GANs can be used to colorize black and white images [2] from the past. People have successfully restored degraded images [3] using a similar technique.

In this project, we approach the problem of image translation using two methods where our model takes an image as input and generates output image from another domain. First, we use an image-level supervised learning approach, called paired image-to-image translation, where paired image samples are used for training. Second, we experiment with a domain-level supervised learning approach where we only have sets of images from two different domains without paired training samples, also called unpaired image-to-image translation. Our contributions in this project can be summarized as follows:

- Combining the training process of paired and unpaired image translation
- Experimenting with various loss metrics such as L1 loss, L2 loss, and their convex combination in the GAN loss to understand their impact. We also experiment with multiple PatchGAN sizes, and model architectures.
- Evaluate the results on quantitative metrics such as precision, recall and Inception FID score. Qualitatively analyze the generated images to corroborate the quantitative results

2. RELATED WORK

For paired image-to-image translation task, Image Analogies [4] used non-parametric models learned using an autoregression algorithm. Unlike this method, our work uses parametric deep learning models, which use feature representations directly from images and produce better quality target domain images. There has been prior work to address specific image translation tasks such as Automatic Image Colorization [5, 6], semantic segmentation [7], edge maps detection [8] which used deep CNNs for learning image feature representations and unstructured losses (per-pixel regression or classification loss) that predicts output pixels independently. In contrast to these, our approach uses structured GAN loss, which generates output image pixel values while preserving the dependency structure resulting in better prediction. Furthermore, our approach can train a deep learning model with the same hyperparameters for multiple domain-transfer tasks.

For unpaired image-to-image translation task, Coupled Generative Network(CoGAN) [9] learns a joint distribution of multi-domain images without using any labeled training image pairs by employing a GAN for each domain and sharing few weight layers among all GANs. However, it is difficult to infer about joint distributions since infinite joint distributions can arise from samples of marginal distributions. Liu et al [10] enforce a shared latent space assumption by using Variational Autoencoders and tying last weight layers of encoder and initial layers of decoders. Our work does not rely on such low-level similarity assumptions hence making our approach flexible for different domains.

3. DATASETS AND PREPROCESSING

We used four open source datasets for this project - CMP facades, maps, cityscapes, horse to zebra. Except the maps dataset, all other datasets had 256x256 RGB images. The maps dataset had 600x600 RGB images, but we randomly cropped them to 256x256 during preprocessing. For the paired task, we used the facades, maps and cityscapes datasets. For the unpaired task, we used the facades, maps and the horse to zebra datasets.

CMP facades dataset [11] is a dataset of manually annotated facades from various sources from various cities around the world having different architectural styles. It contains 606 facade images paired with their corresponding architectural labels.

Maps dataset [12] has images of paired google maps and google earth regions. 2194 images were scraped from google maps from in and around New York city.

Cityscapes dataset [13] includes urban street scenes from 50 cities in Europe, clicked in fair weather across all seasons in the year. We used 3475 images from the dataset.

Horse to zebra dataset [14] was prepared by the authors of the paper from ImageNet [15], using the keywords *wild horse* and *zebra*. It contains 2661 images in total. It is an unpaired dataset, as a horse will not have a corresponding zebra and vice versa.

We did not create any explicit hand engineered features. As the images were passed through a CNN, the CNN learnt the features from the images.

3.1. Preprocessing

We applied the following preprocessing steps on the images.

- *Random flipping*: We generated a random number between 0 and 100, and flipped the image horizontally if the generated number was greater than 50.
- *Random jittering*: We resized the 256x256 images to 286x286 using nearest neighbor extrapolation and randomly cropped them back to 256x256 for random jittering.
- *Normalization*: We normalized pixel values in [-1, 1].

4. MODELS

4.1. Pix2pix Model

In Pix2pix[12], the conditional Generative Adversarial Networks(cGANs) are used. Traditional GANs accept a random noise input to generate an output image belonging to the target distribution, whereas the cGANs accept an input sampled from distribution X and generate the output from another domain Y. Similar to GAN, cGAN also consists of generator and discriminator networks. The goal of the discriminator is to distinguish between the real image that belongs to distribution Y and the image generated by the generator from input x. The goal of the generator is to fool the discriminator into predicting generated image as a real image. The two networks are explained in the following subsections.

4.1.1. Generator

A deep CNN-based architecture is used as a generator. The generator consists of two parts - encoder and decoder. The encoder downsamples the input image while increasing its channel to a latent representation. The encoder comprises blocks of Conv-BatchNorm-Relu. The decoder decompresses this latent representation using transpose convolution layers while reducing the number of channels. The decoder consists of the blocks of ConvTrans-BatchNorm-Relu.

The generator network also contains skip-connections. Similar to U-Net architecture [16], the activations of the encoder network are concatenated to the corresponding inputs to the decoder blocks. Dropout is applied to first three layers of the decoder. The decoder is followed by a CNN, which reduces the number of channels of decoder output to match the input image.

4.1.2. Discriminator

GAN typically uses a deep convolution network as a discriminator, which accepts the actual image sample from the target domain and the generator output to compute a probability score indicating the degree of two inputs belonging to different distributions. In this work, PatchGAN[12], also called markovian discriminator, is used. Instead of computing one probability score for the entire image, PatchGAN aims to classify different image patches of MxM as ground truth or generated, thereby modeling image pixels as Markov random fields. PatchGAN runs a deep convolution network on the concatenated ground truth and generated images (channel-wise concatenation).

4.1.3. Loss Function

Similar to GAN, conditional GAN also uses an adversarial loss function. G($\boldsymbol{x}, \boldsymbol{z}$) is the output from the generator model G where $\boldsymbol{x}, \boldsymbol{z}$ are input and noise, respectively. Similarly let D($\boldsymbol{x}$, G($\boldsymbol{x}, \boldsymbol{z}$)) be the discriminator D output. The motivation behind the adversarial loss is to make the generator better fool the discriminator and discriminator to better differentiate between real and generated image in a minimax fashion. This adversarial loss is given by L_{cGAN} in equation 1. The conditional GAN loss is used with L1(equation 2) loss computed between generated output and ground truth target domain image to make the generator output close to the original ground truth image as possible while promoting better sharpness of edges and reducing blurring. Final loss is given in equation 3 where λ is a regularizer that controls the weightage of L1 loss in the final loss.

$$\begin{aligned} L_{cGAN}(G, D) =& \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}[\log D(\boldsymbol{x}, \boldsymbol{y}) + \\ & \mathbb{E}_{\boldsymbol{x},\boldsymbol{z}}[\log(1 - D(\boldsymbol{x}, \boldsymbol{G}(\boldsymbol{x}, \boldsymbol{z}))] \end{aligned} \tag{1}$$

$$L_{L1}(G) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}}\big[||\boldsymbol{y} - G(\boldsymbol{x}, \boldsymbol{z})||_1\big] \tag{2}$$

$$G^* = arg \min_G \max_D L_{cGAN}(G, D) + \lambda L_{L1}(G) \tag{3}$$

4.2. CycleGAN Model

CycleGAN [14] is used for unpaired image-to-image translation, and it uses the same conditional GAN model from section 4.1. It consists of two conditional GANs - (G_A, D_B) and (G_B, D_A) where G_A is a generator, taking image $\boldsymbol{I_A}$ as input from domain A and generates image $\boldsymbol{I_A^{'}}$ which should belong to the domain B. Similarly G_B is defined. D_B is a discriminator network(section 4.1) distinguishing between real image from domain B and $\boldsymbol{I_A^{'}}$. Similarly D_A is defined.

A **Cycle Consistency Loss** is used to train the cycleGAN model. The generated image $\boldsymbol{I_A^{'}}$ is fed as input to the generator G_B which generates an image which should belong to domain A. To enforce this consistency, L1 loss is taken between $\boldsymbol{I_A}$ and $G_B(G_A(\boldsymbol{I_A}))$. Similarly, for an image $\boldsymbol{I_B}$ belonging to domain B, L1 loss can be enforced between $\boldsymbol{I_B}$ and $G_A(G_B(\boldsymbol{I_B}))$. The net cyclic consistency loss is computed by adding both these L1 losses and are given in equation 4. For the two GANs - (G_A, D_B) and (G_B, D_A), GAN loss is computed and added to the cyclic consistency loss to give full objective in equation 6.

Fig. 1: From left: Input, ground truth and generated images of paired L1 loss experiments on maps and cityscapes datasets, Input and generated images of unpaired L1 cyclic loss experiments on facades and horse2zebra datasets.

$$L_{cyc}(G_A, G_B) = \mathbb{E}_{\boldsymbol{I_A} \sim \boldsymbol{p_{data}}(\boldsymbol{I_A})}\left[||G_B(G_A(\boldsymbol{I_A})) - \boldsymbol{I_A}||_1\right] + \mathbb{E}_{\boldsymbol{I_A} \sim \boldsymbol{p_{data}}(\boldsymbol{I_A})}\left[||G_B(G_A(\boldsymbol{I_A})) - \boldsymbol{I_A}||_1\right] \quad (4)$$

$$L_{cGAN}(G_A, D_B, A, B) = \mathbb{E}_{\boldsymbol{I_B} \sim \boldsymbol{p_{data}}(\boldsymbol{I_B})}\left[\log D_B(\boldsymbol{I_B})\right] + \mathbb{E}_{\boldsymbol{I_A} \sim \boldsymbol{p_{data}}(\boldsymbol{I_A})}\left[\log(1 - D_B(G_A(\boldsymbol{I_A})))\right] \quad (5)$$

$$L(G_A, G_B, D_A, D_B) = L_{GAN}(G_A, D_B, A, B) + L_{GAN}(G_B, D_A, B, A) + \lambda L_{cyc}(G_A, G_B) \quad (6)$$

5. EXPERIMENTS

We experimented by using different loss functions, changing the sizes of the Patch GAN and removing skip connections from the U-Net [16] architecture to see how the results are affected.

5.1. Paired Task Experiments

We trained the conditional GAN model using binary crossentropy loss as the GAN loss for the generator and also for the discriminator. We used the Adam optmizer for both networks with a learning rate of 2×10^{-4} and β_1 as 0.5. One gradient descent step is performed for generator and one step for the discriminator. While performing gradient step for discriminator network, the loss objective is divided by 2 to slow down the rate of discriminator learning relative to generator. This is done to stabilize the GAN training. We tried different values of λ but we got our best results with $\lambda = 100$. We trained all models with a batch size of 16 for 150 epochs. For all experiments except the two Patch GAN experiments, we used a 70x70 Patch GAN.

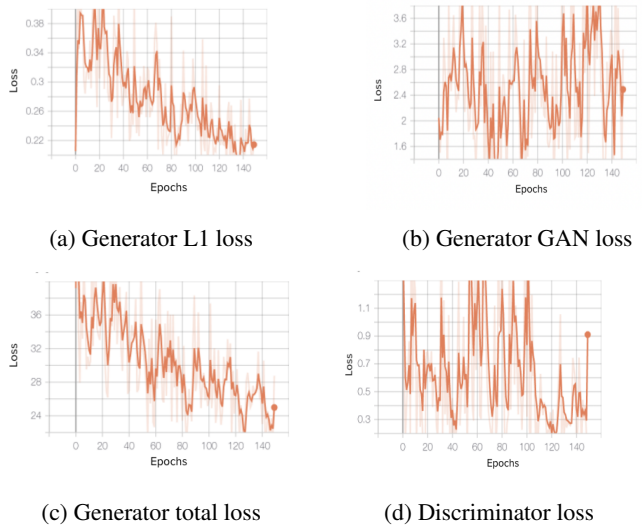
We conducted the experiments below for the paired task.

1. L1 loss: L1 loss between generated and real images in addition to the GAN loss for the generator.
2. L2 loss: L2 loss between generated and real images in addition to the GAN loss for the generator.
3. 0.5× L1 + 0.5× L2: A combination of L1 and L2 loss to see which loss leads to better images.
4. Patch 16: Patch GAN of size 16x16.
5. Patch 286: Patch GAN of size 286x286.
6. Skip: UNet architecture without skip connections.

5.2. Unpaired Task Experiments

Similar training settings as in paired task experiments are used to train unpaired task experiments.

1. L1 loss: L1 distance to compute cyclic loss
2. L2 loss: Mean square error based cyclic loss
3. 0.5× L1 + 0.5× L2: A convex combination of L1 and L2 cyclic loss



(a) Generator L1 loss (b) Generator GAN loss

(c) Generator total loss (d) Discriminator loss

Fig. 2: Loss curves for the L1 loss experiment on the maps dataset with batch size=16 for 150 epochs

We have shown the training loss curves for the L1 loss experiment on the maps dataset in Fig 2. The generator and discriminator losses do not follow a set pattern as the generator tries to minimize the GAN loss, while the discriminator tries to minimize it. The generator L1 loss, which compares the generated and the real images, decreases with the number of epochs as the generator is generating better images. Similarly, the generator total loss, which is a combination of the generator GAN loss and the L1 loss also follows the decreasing trend of the L1 loss.

6. QUANTITATIVE RESULTS

To evaluate our model, we used precision and recall and inception score as the metrics for 256 images on every experiment.

6.1. Precision and Recall for GANs

We followed the method of Kynkäänniemi et al. [17] to calculate the precision and recall for GANs. The InceptionV3 classification network [18] was used to extract high dimensional features from the generated and real images. Let the extracted feature vectors be $\mathbf{\Phi}_g$ and $\mathbf{\Phi}_r$ respectively. For each set of feature vectors $\mathbf{\Phi} \in \{\mathbf{\Phi}_g, \mathbf{\Phi}_r\}$, a hypersphere with radius equalling the distance from

Fig. 3: Images generated by the experiments on the facades dataset. Images from left: Input, Ground Truth, L1 cyclic loss, L2 cyclic loss, L1 loss, L2 loss, Patch 16, Patch 286, Skip.

Experiment	Cityscapes			Maps			Facades		
	Precision	Recall	FID	Precision	Recall	FID	Precision	Recall	FID
L1 Loss	0.52	0.32	100.52	0.43	0.32	152.13	0.76	0.39	110.68
L2 loss	0.59	0.30	115.31	0.35	0.15	195.11	0.64	0.26	118.84
0.5×L1 + 0.5×L2	0.61	0.19	108.70	0.42	0.05	205.14	0.71	0.39	116.89
Patch GAN 16	0.19	0.23	147.51	0.33	0.06	211.69	0.76	0.29	116.50
Patch GAN 286	0.48	0.19	154.22	0.31	0.15	198.18	0.73	0.21	124.90
Skip	0.03	0.01	340.10	0.23	0.01	275.91	0.13	0.01	275.91

Table 1: Results on paired translation task. Experiments were run for batch size-16 and 150 epochs. L1 loss experiment seems to perform the best.

its kth nearest neighbor is defined in the high dimensional space, using Euclidean distance as the distance metric. To determine if a given vector ϕ lies in the hypersphere, we use

$$f(\phi, \Phi) = \begin{cases} 1, \text{ if } \left\|\phi - \phi^{'}\right\|_2 \leq \left\|NN_k(\phi^{'}, \Phi) - \phi^{'}\right\|_2, \forall \phi^{'} \in \Phi \\ 0, \text{ otherwise} \end{cases} \tag{7}$$

Here, $NN_k(\phi^{'}, \Phi)$ denotes the k^{th} nearest neighbor of ϕ in set Φ. If $\phi \in \Phi_g$ lies in the hypersphere of any $\phi^{'}$ in Φ_r, then it is considered a real image. Intuitively, we can say that $f(\phi, \Phi_g)$ is measure of how realistic an image is while $f(\phi, \Phi_r)$ is a measure of if a real image can be generated from a generator. So, precision and recall we use in our experiments can be defined as follows.

$$\begin{aligned} \text{Precision} &= \frac{1}{|\Phi_g|} \sum_{\phi \in \Phi_g} f(\phi, \Phi_r) \\ \text{Recall} &= \frac{1}{|\Phi_r|} \sum_{\phi \in \Phi_r} f(\phi, \Phi_g) \end{aligned} \tag{8}$$

The results are summarized in Table 1.

6.2. Frechet Inception Distance (FID)

FID [19] calculates the quality of generated images by computing its correlation with the real images. The images are passed through the pretrained Inception Network [18], and features are calculated using the second last layer of the model. A lower FID score corresponds to a better generator.

Let $(\boldsymbol{\mu_r}, \boldsymbol{C_r})$ and $(\boldsymbol{\mu_g}, \boldsymbol{C_g})$ correspond to the mean and covariance matrix for the real and generated images. The FID similarity score between the two image sets is calculated using the following equation.

$$\text{FID} = \|\boldsymbol{\mu_r} - \boldsymbol{\mu_g}\|_2^2 + Tr(\boldsymbol{C_r} + \boldsymbol{C_g} - 2(\boldsymbol{C_r}\boldsymbol{C_g})^{1/2}) \tag{9}$$

7. QUALITATIVE RESULTS

This section qualitatively compares the learned pix2pix models in different experiments on image ID 8 of the facade validation dataset. The results are presented in Fig 3. We can observe that the outputs generated by training the pix2pix model with L1 loss or L2 loss are better. These models receive image-level supervision signals, and L1/L2 loss also forces the generator output distribution to match the target domain distribution. We can also observe that the image generated by the Patch GAN 16 experiment is better and has more contrast as compared to the one generated by the Patch GAN 286 experiment. This shows that by increasing the patch size, the discriminator model becomes more relaxed in discriminating between two images based on smoothness level. A smaller patch size enforces the generator to generate images with more contrast and sharp edges, improving generated image quality. Also, we can see that the paired task experiments result in better images than the unpaired task as the latter does not have ground truth images for training.

Experiment	Maps	Facades	Horse2Zebra
L1 Cyclic Loss	235.67	166.12	204.09
L2 Cyclic Loss	245.76	205.95	245.89
0.5*L1 + 0.5*L2	298.45	187.34	211.78

Table 2: Table depicts the inception score for unpaired image translation task. The results with L1 cyclic loss seem to dominate the performance with lower FID score, and is consistent with our findings.

8. CONCLUSION

In this project, we experiment with various generative methods to translate an image from one domain to another. Our results corroborate the efficacy of paired image translation over unpaired task on the multiple dataset using pix2pix GAN, and we also show the improvement in image quality with L1 GAN loss on various datasets. We were also able to incorporate various qualitative metrics such as FID score, precision and recall for images to achieve a similar result.

9. INDIVIDUAL CONTRIBUTIONS

All three members contributed equally in writing the report. We also contributed equally in coding the architecture and training the models. We worked together to analyze the effectiveness of skip connections in the convolutional encoder and decoder networks and experimented with different sizes of PatchGAN and other discriminators and observe their effect on training and generation of images.

Having said that, there were certain tasks that each team member explicitly focused on. We are summarizing it as follows:

- Harpreet Singh
    1. Identified potential datasets for paired image-to-image translation task. Carried out required preprocessing to prepare the datasets for the experiments.
    2. Performed the L1 loss and L2 loss experiments on the paired task.
    3. Performed qualitative analysis of the generated images from both the tasks.
- Gaurav Kumar
    1. Reviewed unpaired image to image translation task training pipeline. Trained variants of the conditional GAN architecture and analyze the results.
    2. Performed the experiments on the unpaired task.
    3. Analyzed literature for FID and used FID to quantitatively analyze the images generated by both the paired and unpaired tasks.
- Soham Satyadharma
    1. Explored datasets for the unpaired image-to-image translational network and idetified the potential challenges while working with the dataset.
    2. Performed the patch GAN and skip experiments on the paired task.
    3. Analyzed literature for precision and recall to quantitatively analyze the images generated by the paired task.

10. REFERENCES

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[2] Kamyar Nazeri, Eric Ng, and Mehran Ebrahimi. Image colorization using generative adversarial networks. *Lecture Notes in Computer Science*, page 85–94, 2018.

[3] Dennis Estrada, Susanne Lee, Fraser Dalgleish, Casey Den Ouden, Madison Young, Caitlin Smith, Joseph Desjardins, and Bing Ouyang. Deblurgan-c: image restoration using gan and a correntropy based loss function in degraded visual environments. In *Big Data II: Learning, Analytics, and Applications*, volume 11395, page 1139507. International Society for Optics and Photonics, 2020.

[4] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340, 2001.

[5] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[6] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European conference on computer vision*, pages 577–593. Springer, 2016.

[7] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[8] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.

[9] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. *arXiv preprint arXiv:1606.07536*, 2016.

[10] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *arXiv preprint arXiv:1703.00848*, 2017.

[11] Radim Tyleček and Radim Šára. Spatial pattern templates for recognition of objects with regular structure. In *Proc. GCPR*, Saarbrucken, Germany, 2013.

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[13] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[16] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[17] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *arXiv preprint arXiv:1904.06991*, 2019.

[18] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[19] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018.