Image-to-Image Translation Using Paired Images using Pix2Pix Step 2

Adithya Shrivastava
as3652
Rutgers University
Department of Computer Science
as3652@scarletmail.rutgers.edu

Nishant Dhargalkar
nsd78
Rutgers University
Department of Computer Science
nsd78@scarletmail.rutgers.edu

Janish Parikh
jrp328
Rutgers University
Department of Computer Science
janish.parikh@rutgers.edu

Jiawei Tang
jt1039
Rutgers University
Department of Computer Science
jiawei.tang@rutgers.edu

Abstract

Generative Adversarial Networks (GANs) *are deep-learning based generative models.They can be used for a wide variety of purposes, one of which is image-to-image translation.* ***Image to image translation*** *is the task of mapping images from one domain to another, the former referred to as the source domain and the latter the target domain, without any loss of the source content representations. Image-to-image translation has various approaches, such as supervised, unsupervised, semi-supervised and few-shot. The supervised approach to image-to-image translation is known as* ***paired image-to-image translation****. This is because in this approach, there are cross-domain aligned pairs of images, i.e. aligned image pairs each having one image in the source domain and one in the target domain. This makes it a supervised approach since we have a mapping from source image to desired target image. The GAN architecture developed for performing paired image-to-image translation is known as the* ***Pix2Pix architecture****. The aim of this task is to implement paired image-to-image translation using the Pix2Pix architecture on the* ***Dayton dataset****. Further, we qualitatively as well as quantitatively evaluate this model using the* ***Frechet Inception Distance (FID)*** *and* ***Inception Score (IS)*** *evaluation metrics.*

1. Introduction

Image-to-Image translation has applications in areas such as converting black and white photos and videos to colour, style transfer, season transfer, etc. This report focuses on paired image-to-image translation, which is a supervised approach. In this technique, each image in the source domain is mapped to the desired image in the target domain. The model is trained to learn this mapping. The architecture used for this technique is called the Pix2Pix architecture [2], which is a **Conditional GAN architecture**. In traditional GAN and DCGAN, we cannot control the class of the image generated by the generator. Conditional GANs overcome this drawback by conditioning the generator and discriminator on specific class labels. The Pix2Pix architecture is an extension of Conditional GANs in which instead of feeding a random noise vector as input to the generator, the image from the source domain is given as input. The output of the generator is the translated image, i.e. the desired image from the target domain. The discriminator, which is a conditional discriminator, is fed a pair of images as input. One image in this pair is the input image, and the other is either the real output image (i.e. the one from the dataset) or the fake output image (the one generated by the generator). The discriminator learns to classify whether the output image is real or fake. In this phase of the project, we perform paired image-to-image translation on the Dayton dataset [5], which is a dataset of street views and overhead views of roads in the US. We also perform a quantitative evaluation of this model using the Frechet Inception Dis-

tance (FID) [1] and Inception Score (IS) [4] evaluation metrics.

The remainder of this report is organized as follows. We introduce the preprocessing techniques and the dataset required for training the model in 2. We present the architecture of the model and the method used for evaluating the model in 3. We evaluate the models qualitatively 4 as well as using the metrics selected in 6.

2. Pre-processing

2.1. Dataset

For the supervised image-to-image translation model using the pix2pix architecture we use the Dayton dataset. The dataset is downloaded from https://www.mediafire.com/folder/f4gga3h86d659/GTCrossView. It contains images of street views and aerial views of roads in 11 cities in the US [1]. The dataset is useful for tasks involving image translation from ground to aerial view and vice versa. The dataset contains a total of 76,048 images. We augmented the dataset and split it into 55,000 pairs of training images and 21,048 pairs of testing images [3].

2.2. Image Pre-processing

The images in the original Dayton dataset [5] are of size 354 x 354 pixels, but we resized them to the size of 256 x 256 pixels. We also use data augmentation methods like adding random jitter and random mirroring of the image. For random jittering we first resize the image to 286 x 286 pixels by using the nearest-neighbour interpolation technique. We then stack the input and the ground truth image and use a random crop operation to resize them to 256 X 256 size. The random crop operation uses a uniform distribution to randomly sample the data. For random mirroring a point is sampled from a uniform distribution in the range [0, 1]. If the sampled point is greater than 0.5 then a mirror image is created of both the input and ground truth label. Both the input and the ground truth labels are normalized to $[-1, 1]$.

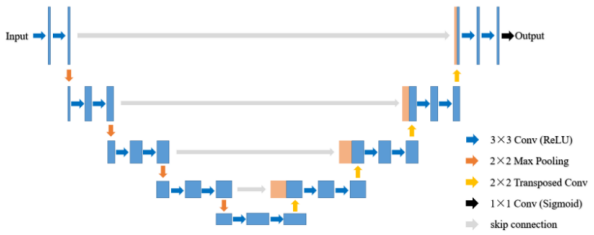

Figure 1. U-Net architecture

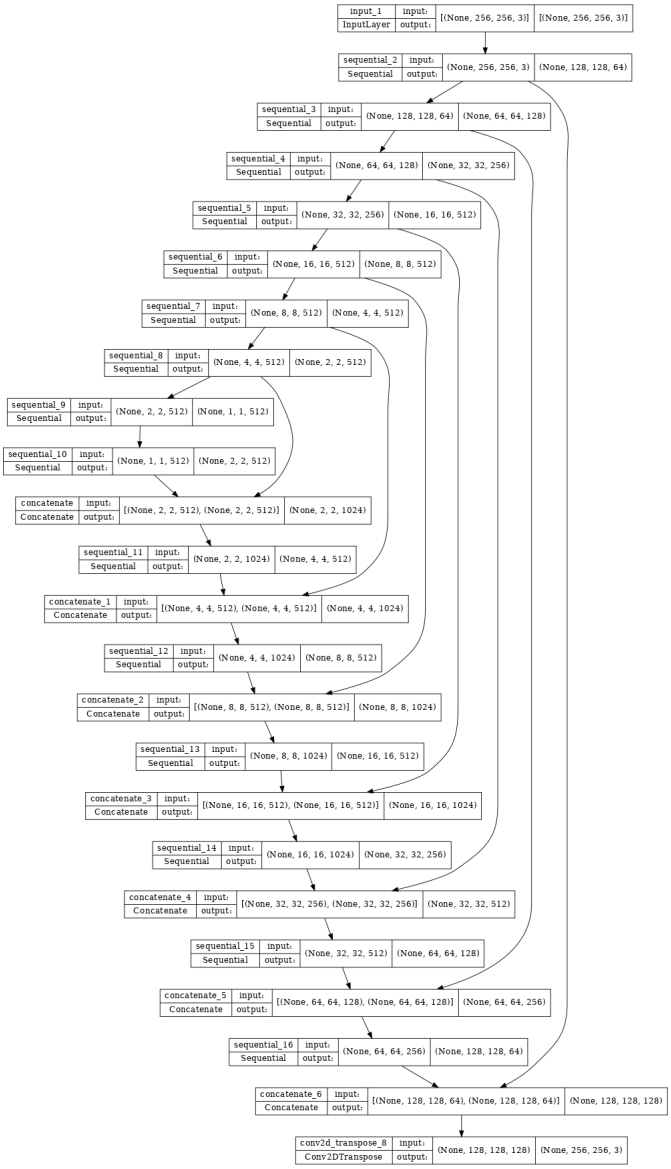

Figure 2. Layers of generator

3. Implementation

3.1. Generator architecture

A generic GAN generator model can be thought as a decoder model whereas the generator in Pix2Pix model [2] can be thought as having an encoder-decoder structure. The Pix2Pix model architecture uses Convolution-BatchNormalization-Activation blocks of layers for the generator called U-Net [2].The U-Net model can be thought of as consisting of two parts. In the first part the convolution layers down samples the data and in the second part the convolution layers upsamples the data. This is reason why it is called 'U-Net' because of its peculiar U-shape. The U-Net architecture is different from the encoder-decoder architecture due to the fact that it also uses skip connections. This allows it to propagate context information from the lower dimensional layers to higher dimensional layers. It also helps with the problem of overfitting of data. The U-Net architecture also does not make use of any max-pooling layers and uses strided convolutions with stride of 2. During the downsampling procedure, each image is downsampled by a factor of 2. All the weights are initialized to a mean of 0 and a standard deviation of 0.2.

3.2. Discriminator architecture

The goal of the discriminator in the GAN architecture is to detect whether the given image is real or fake. In the Pix2Pix model, the PatchGAN architecture is used for the discriminator [2]. A PatchGAN classifies whether patches of a given image are real or fake, rather than the whole input image. The size of the patch is 70×70 [2]. We use L1 loss as a regularizer. The output of the PatchGAN is a 30 x 30 tensor, which represents all patches of size 70 x 70 in the input image. Each value in this 30 x 30 matrix represents the probability of one particular 70 x 70 patch in the input image being real or fake. All the weights of the discriminator are initialized to a mean of 0 and a standard deviation of 0.2.

3.3. Training

The training of the pix2pix model [2] is the same as the conventional GAN model apart from the fact instead of feeding noise into the generator network it is fed an input image. The discriminator trained on the same input image then tries to classify whether a given image is real or fake (generated by generator). Discriminator and Generator are both Neural Networks that are trained together. In pix2pix model, a Discriminator competes with a Generator as two players of zero-sum games.

Refer step 3.3.1 and 3.3.2. The training of pix2pix model can be understood better from 4.

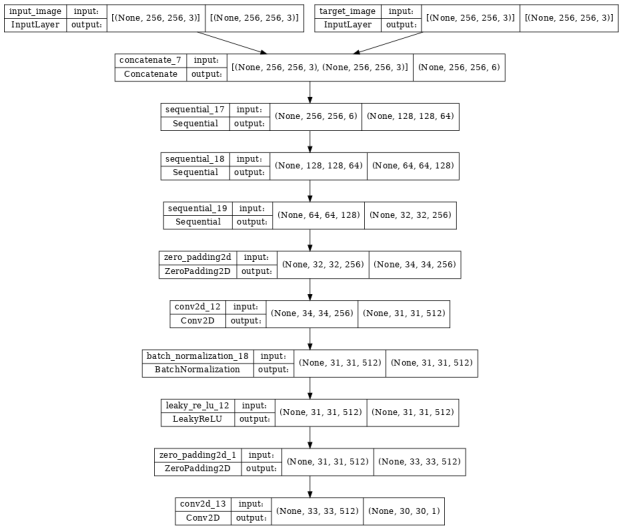

Figure 3. Layers of Discriminator

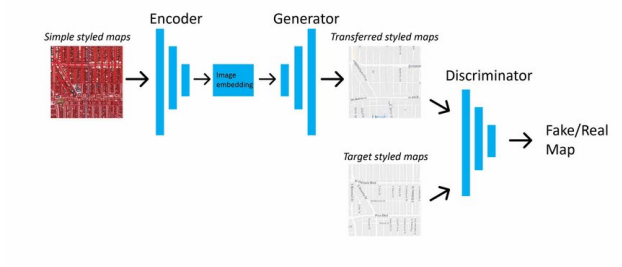

Figure 4. Pix2Pix model Training

3.3.1 Training Discriminator

Get real images from dataset and input them to the discriminator. Label the output as 1. Calculate the loss on the real images. Use those same images and input it to the generator to get $G(z)$, and then input $G(z)$ to the discriminator. Label the output as 0. Then calculate the discriminator loss on the generated images. The real loss is responsible for making the discriminator output a value of 1 for real images while the generated loss is responsible for outputting a value of 0 for images generated by the generator. The loss function used for training the discriminator is Binary Cross Entropy loss. To slow down the rate at which the discriminator is trained relative to generator we divide the total loss of the discriminator by 2 at the time of optimization.

Discriminator loss = (Real loss + Generated loss)/2.

This loss is used for optimizing the discriminator by making use of backpropagation algorithm. The optimizer used for this purpose is Adam.

3.3.2 Training Generator

Get real images from dataset and input it to the generator to get $G(z)$, and then input $G(z)$ to the discriminator. Label

the output as 0. The loss of the discriminator on these generated images is then backpropagated for optimizing the optimizer. The loss function used for training the generator is Binary Cross Entropy (BCE) loss. The BCE loss is the output of the discriminator against the real images. It is used for optimizing generator. In addition to this as mentioned in [5] we use L1 loss that acts as a regularization term. L1 loss is also responsible for creating realistic renderings that correspond to the input labels.

Generator loss = BCE Loss + $\lambda\Sigma$|Generated output - Target|.

Adam is the optimizer used for optimizing the generator by making use of backpropagation algorithm.

3.4. Procedures for Training

We used pyTorch for the implementation of the Pix2Pix architecture and Tensorboard for visualisation.

3.4.1 Training for Batch Size 128 and on 4 GPUs

We first trained the models for aerial to street views for 200 epochs each. The performance of the models improved with increase in the number of epochs. However, in the case of the aerial to street view model, we observed that the model began to overfit after the 160^{th} epoch. Initially each epoch took us around 9 minutes but later due to increased load on the ilabs it went upto 11 minutes per epoch.

3.4.2 Training for Batch Size 48 and on 2 GPUs

We first trained the models for street to arieal views for 200 epochs each. The performance of the models improved with increase in the number of epochs. Initially each epoch took us around 15 minutes but later due to increased load on the ilabs it went upto 18 minutes per epoch.

3.4.3 Training for Batch Size 24 and on 1 GPU

We again trained both the models using a batch size of 24 and trained on only one GPU for 75 epochs. Here, we also calculated the FID and Inception Score evaluation metrics. For this, we created a validation set by randomly selecting 10 images from the test set, after each epoch during training. So, after each epoch, the FID and Inception Score was evaluated on the 10 images selected. We observed improvements in the performance of the models, as well as the values of the evaluation metric scores with increase in the epochs.

3.4.4 Training for Batch Size 1 and on 1 GPUs

We were originally supposed to use a batch size of 1 for training, however, the training was taking 1.5 hours per epoch with this batch size. We started training the aerial to street model with batch size 1 for 50 epochs, and left it to train while simultaneously training both aerial to street and street to aerial models using batch size 128 and batch size 24, as explained above. With batch size 1, the models have completed 32 epochs so far. The FID and Inception Score are also being calculated here using the same method as explained earlier.

4. Qualitative Evaluation of the Models

We evaluated the models on the basis of the quality of the images generated. Figures 5, 7, 8, 9, 10 and 11 represent the images generated by the street to aerial and aerial to street models.

Street to Aerial

Figure 5. Street to Aerial View image generated at batch size 48 at epoch 100

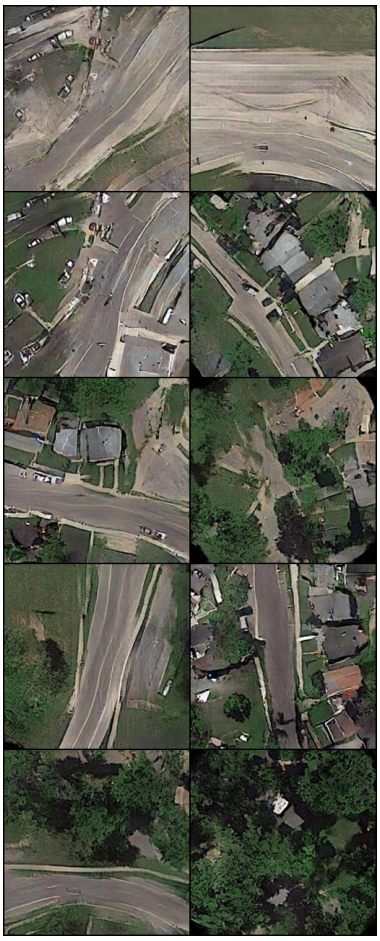

Figure 6. Street to Aerial View image generated at batch size 24 at epoch 150

We observed that we were able to generate better results with a lower batch size. Also we can observe that after 90 epochs for batch size 128, the images get pixalated and blurred. This can be attributed to the L1 regularization in the Pix2Pix loss.
Images with multiple objects or having curves in the aerial were specifically difficult for the the generator to regenerate and the same can be observed from the results above and the images added.
We were able to obtain good results for batch size 24.

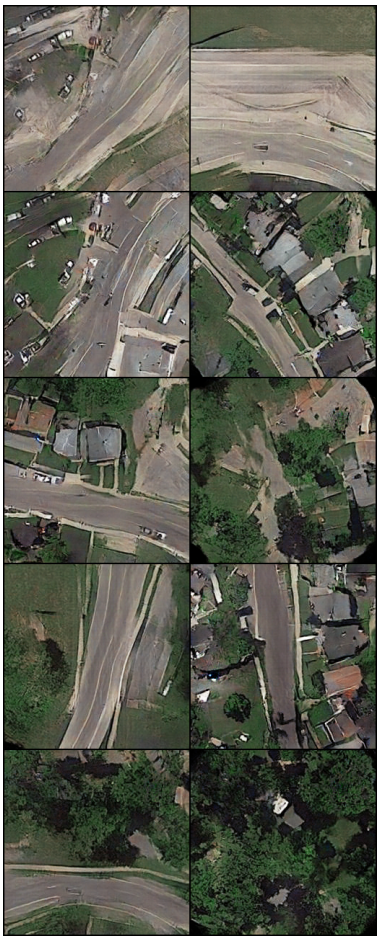

Figure 7. Street to Aerial View image generated at batch size 24 at epoch 150

Aerial to Street

Similar to street to aerial for aerial to street also we

Figure 8. Aerial to Street View images generated at batch size 128 and epcoch 150

Figure 9. Aerial to Street View images generated at batch size 128 epoch 200

Figure 10. Aerial to Street View image generated at batch size 24 epoch 100

observed that we were able to generate better results with a lower batch size.

We also observed that after 150 epochs for batch size 128, the images quality got deteriorated, they were highly blurred and made no sense. We think that after 150 epochs for higher batch size as seen in 9 the model overfits. This can be attributed to the L1 regularization in the Pix2Pix loss.

Figure 11. Aerial to Street View image generated at batch size 1 epoch 20

Further we can observe that the results obtained on training for batch size 1 11 at epoch 20 were much better than the results obtained after a 100 epochs of batch size 24 and 128.

In Aerial to street, the buildings, curved streets, houses were difficult to regenerate and we can observe a blur around these areas specifically.

5. Images

5.1. Street to Aerial

Figure 12. Street to aerial epoch 10 batch size 48

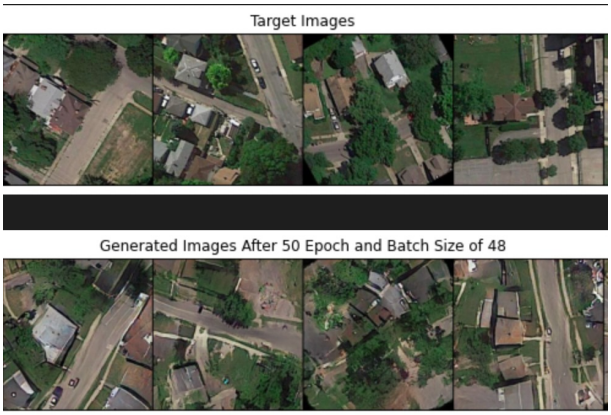

Figure 13. Street to aerial epoch 50 batch size 48

Figure 14. Street to aerial epoch 100 batch size 48

Figure 15. Street to aerial epoch 150 batch size 48

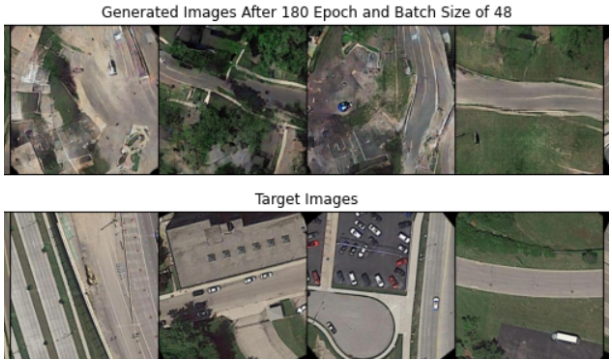

Figure 16. Street to aerial epoch 180 batch size 48

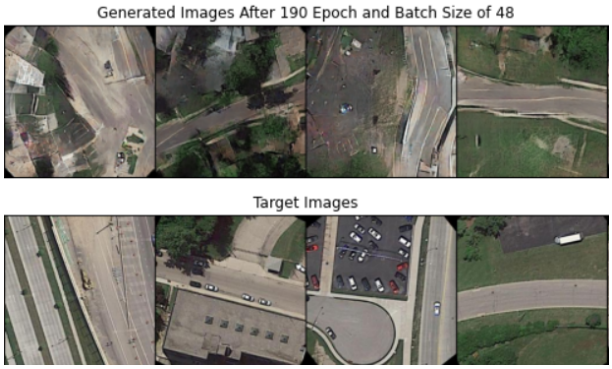

Figure 17. Street to aerial epoch 190 batch size 48

Figure 18. Street to aerial epoch 200 batch size 48

5.2. Aerial to Street

Figure 19. Aerial to Street, batch size 24, epoch 5

Figure 20. Aerial to Street, batch size 24, epoch 50

Figure 21. Aerial to Street, batch size 24, epoch 70

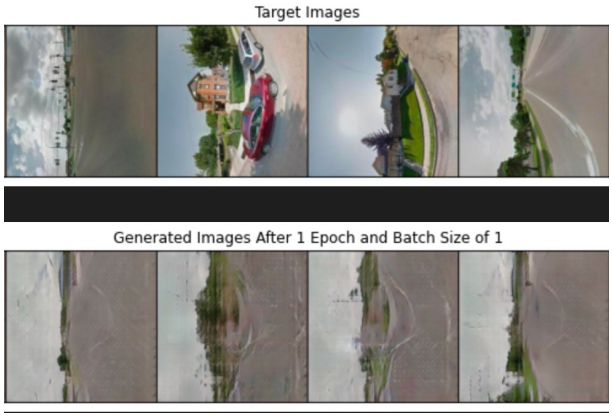

Figure 22. Aerial to Street, batch size 1, epoch 1

Figure 23. Aerial to Street, batch size 1, epoch 6

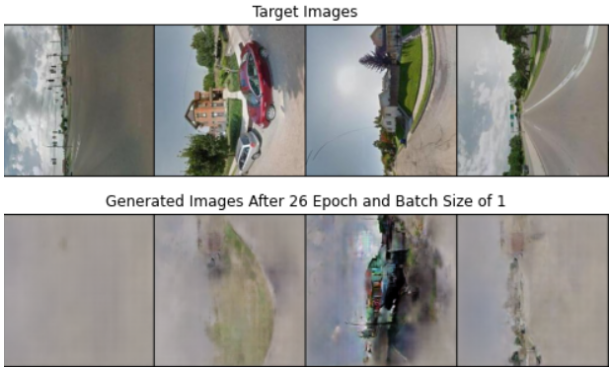

Figure 24. Aerial to Street, batch size 1, epoch 26

6. Quantitative Evaluation of the Models

We chose Frechet Inception Distance [1] and Inception score [4] same as Step-1 for our evaluation metrics and calculated both the metrics on the batch size of 24.

6.1. Frechet Inception Distance

The Frechet Inception Distance (FID) [1] measures the distance between the distributions of synthesized im- ages and real images that are used to train the genera- tor. A good model generates images with a lower FID scores.
The FID score of our model trained on a batch size of 24 can be seen from 28 and 27.

6.2. Inception score

Inception score [4] uses a pretrained Inception v3 model, and predicts the class probabilities for each generated image. The score is limited by what the Inception (or other network) classifier can detect, which is directly linked to the training data. 25 and 26.

7. Analysis

7.1. Street to Aerial Inception Score

For the initial epochs, we can observe that the inception score [4] for street to aerial is approximately 1.5 and on training more epochs, we can observe a gradual increase in the inception score.

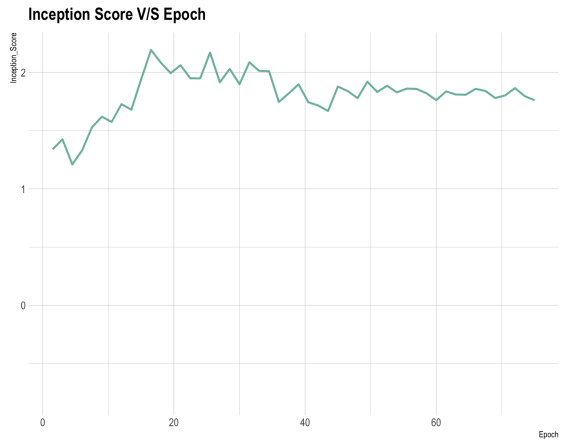

Figure 25. Street to aerial Inception score(IS)

7.2. Aerial to Street Inception Score

For the initial epochs, we can observe that the inception score [4] for aerial to street is approximately 0.8 and on training more epochs, we can observe a gradual increase in the inception score reaching a score of 2 after 85 epochs. This was the expected result, as the performance of the model should improve with increase in the number of epochs resulting in a higher Inception Score.

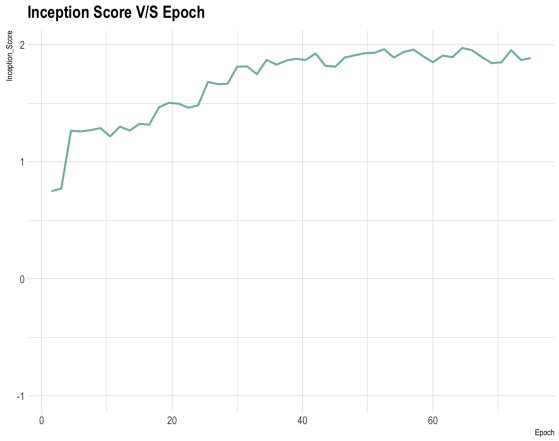

Figure 26. Aerial to street Inception score(IS)

7.3. Street to Aerial FID

In the case of the street to aerial model, initially, the value of the FID score [1] is approximately 386. We can see a decrease in the FID score with increase in the number of epochs, as the quality of the generated images improves with increase in the number of epochs. The FID score finally reduces to approximately 355. We can observe that the FID score is slightly higher for the street to aerial model as compared to the aerial to street model.

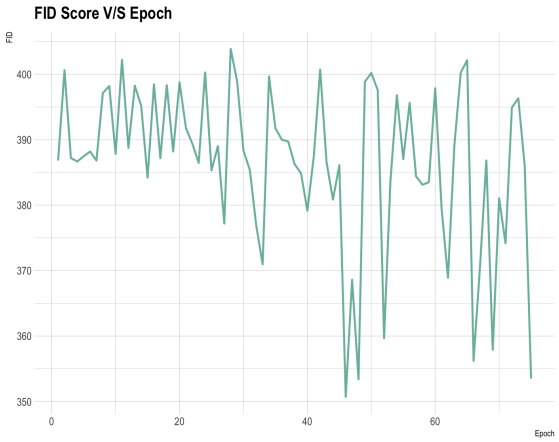

Figure 27. Street to aerial Frechet Inception Distance (FID) score

7.4. Aerial to Street FID

The FID Score for the aerial to street model was 242.5 initially. With increase in the number of epochs, there was

a gradual decrease in the FID score [1] and we can see that it decreases to 237. This was consistent with the quality of the images generated, as the quality improved with increase in the epochs, implying a lower FID score.

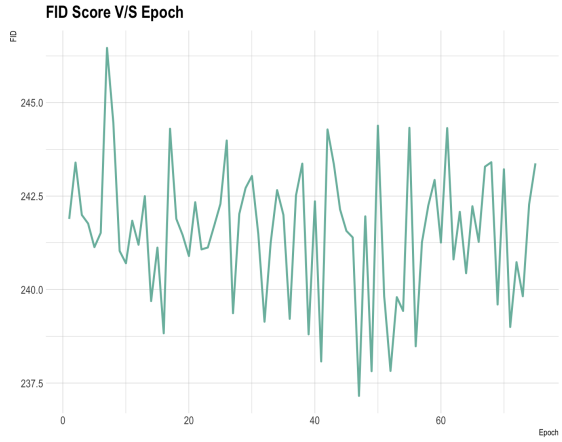

Figure 28. Aerial to street Frechet Inception Distance (FID) score

8. Conclusion

In this project step we tried to tackle the paired Image-to-image translation problem which is challenging in it's nature as it often requires specialized models and loss functions for a given translation task or dataset at hand.To solve this problem we explored the Pix2Pix GAN that models the loss function using a combination of L1 Distance and Adversarial Loss with additional novelties in the design of the Generator and Discriminator that allows us to generate images that are both plausible in the content of the target domain, and is also a plausible translation of the input image.

We were able to reproduce the results of the Pix2Pix architecture. The results obtained were in accordance of our evaluation metrics, FID and IS. The images that had a lower FID and high IS were also the ones that performed good in qualitative evaluation. The FID for both the transformations Street to Aerial and Aerial to Street decreased over increasing epochs and the Inception Score increased on increasing epochs.

The future scope of work would be to train both the unpaired Pix2Pix networks for a batch size of 1 and observe the results. Also to optimize the runtime one can explore learning schedules.

References

[1] 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. 2017. 2, 9, 10

[2] P. Isola, J.-Y. Zhu, T. Zhou, and A. 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, 2011. 1, 3

[3] K. Regmi and A. Borji. Cross-view image synthesis using conditional gans. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, page 3501–3510, 2018. 2

[4] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans, 2016. 2, 9

[5] N. N. Vo and J. Hays. Localizing and orienting street views using overhead imagery. *European conference on computer vision*, page 494–509, 2016. 1, 2, 4