CS536 Project Step 2 - Theme 2: Image to Image Translation

Saiyed SohailAbbas
Rutgers University
Department of Computer Science
ss3723@rutgers.edu

Sachin Wani
Rutgers University
Department of Computer Science
sw1051@rutgers.edu

Rasika Hasamnis
Rutgers University
Department of Computer Science
rmh229@rutgers.edu

1. Previous Step: Quantitative Evaluation of CycleGANs

The code submitted in previous work contained the part of quantitative evaluation. The reconstruction error provided identity loss from images Generated from $|H - G_H(G_Z(H))|$ and $|Z - G_Z(G_H(Z))|$

Other evaluation metric that was used was Inception Score, we found the evaluation between 1.8 and 2.9 over the epochs.

2. Introduction

GANs or Generative Adversarial Networks are used to synthesize photo- realistic images. GANs make use of two architectures: Generator which takes in an input noise and input image in case of Image-to-Image translation and outputs an image which it generates based on the task at hand. The discriminator makes use of Real images and generated images passed through a network to output whether the given image is real or generated. Both the architectures work in tandem to improve each other. In our project, we are using Pix2pix [3] architecture on the Dayton data set to carry out street-view to overhead-view image translation, and vice-versa.

3. Model Used

We are using a Pix2Pix model to translate the images. The Pix2Pix GAN is a general approach for image-to-image translation. It is based on the conditional generative adversarial network, where a target image is generated, conditional on a given input image. cGANs are trained on paired set of images or scenes from two domains to be used for translation. The simple GAN only focuses on generating a fake image without seeing how plausible it is. Hence, we condition the source image. The Pix2Pix GAN changes the loss function so that the generated image is both plausible in the content of the target domain, and is a plausible translation of the input image.

Pix2pix GAN has a Generator and Discriminator. The generator has an encoder which is a classification network like VGG/ResNet where you apply convolution blocks followed by a maxpool downsampling to encode the input image into feature representations at multiple different levels. This is followed by the decoder, which consists of upsampling and concatenation followed by regular convolution operations. In the network, the input is passed through a series of layers that progressively downsample (encoder), until a bottleneck layer, at which point the process is reversed (decoder).

The Discriminator is of Patch-GAN architecture and takes a pair of input images, target image and generated image. It works by classifying a patch in a image into real and fake rather than classifying whole image into real and fake. Thus it has better accuracy than a simple GAN. This also works faster than classifying whole image and has less parameters.

4. Dataset

We use the Dayton dataset [2], which contains images of street views and aerial views of roads. It is a dataset for ground-to-aerial (or aerial-to-ground) image translation. There are 76,048 images in total for each case of resolution 354X354 for aerial images and 230x230 for street view images.

5. Steps for model creation

The model and the running code is available in the attached ProjectTeam25-Pix2Pix-Step2.ipynb file in this submission. All extra references and motivations for the code has been added as comment within the Jupyter Notebook. Inspiration taken from open-source Pix2Pix implementation. [1]

5.1. Parameter setting

We set the parameters like generator learning rate, Discriminator learning rate, Epochs. We set size of the feature maps in the generator and discriminator. We set the buffer size and the batch size for the training. We use random jittering resizing as a parameter for data augmentation.

5.2. Directory Creation

We create the directory to store the data we need to train the model along with the checkpointed weights and images generated to be used for qualitative and quantitative evaluation in the later section.

5.3. Preprocessing

There are two parts in the Dayton dataset, which we are using – overhead images and street images of the same scene. Each of the images are scaled to 256X256 pixels and images of the same scene are paired, so that the resulting image becomes 256X512 necessary for training the pix2pix model. We have split the data into training and testing data in the ratio of 55000:21048.

6. Architecture definition

In pix2pix, we have the PatchGAN discriminator and used the U-Net architecture for generator.

7. Evaluation

7.1. Qualitative Evaluation

We can observe the results obtained by looking at the generated images from the pix2pix and looking at how realistic the generated images are compared to the ground truth. One qualitative measure to say that the model has generated images with high fidelity is that they are indistinguishable from the actual input images used to train the model.

7.2. Quantitative Evaluation

While Qualitative evaluation can help us see if the progress is happening or not, to objectively verify and compare the results, we need a quantitative measure. To evaluate our working of GAN, we have the following quantitative measures of evaluation:

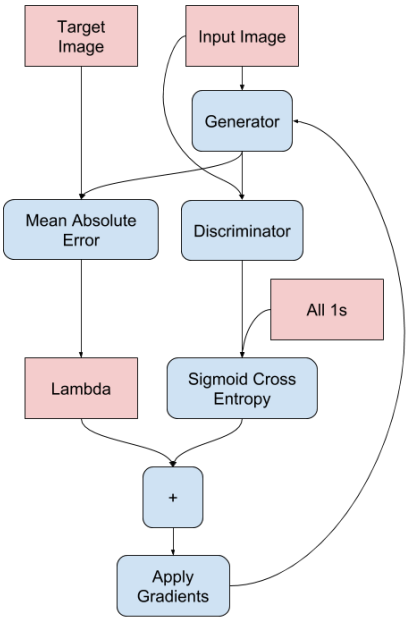

Figure 1. U-Net architecture for the generator

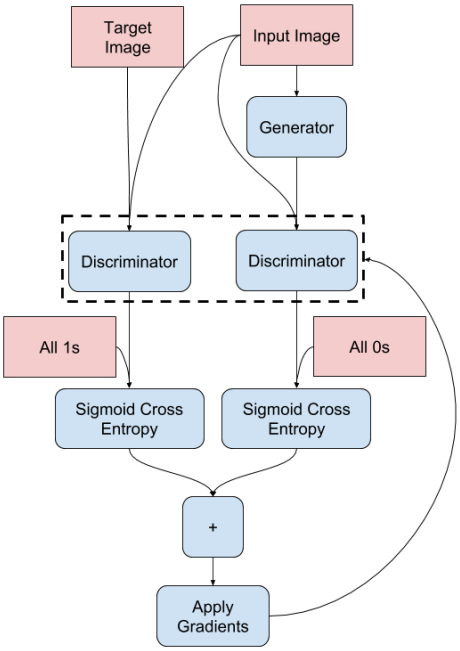

Figure 2. PatchGAN discriminator

7.2.1 Inception Score

The Inception score measures how well the image is clearly defined in the sense it is distinct, and makes sure the image has variety. It outputs a single floating-point number (higher

Figure 3. Predicted output of the paired image to image translation in Pix2Pix for Overhead to Street Conversion

Figure 4. Predicted output of the paired image to image translation in Pix2Pix for Street to Overhead Conversion

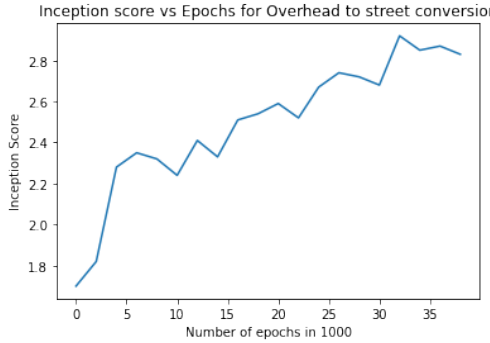

Figure 5. Inception score increasing vs epochs for Overhead to Street conversion in pix2pix

the better). The Inception Score takes images and returns probability distribution of labels for the image. If it is a narrow distribution, the image is well-formed, and if it is a broad distribution, the image is jumbled.

We obtain the following results for both the transformation in figure 5 and 6:

7.2.2 Frechet Inception distance

The Fréchet inception distance (FID) is a metric used to assess the quality of images created by a GANs. Unlike the earlier inception score (IS), which evaluates only the distribution of generated images, the FID compares the distribution of generated images with the distribution of real images that were used to train the generator.
Rather than directly comparing images pixel by pixel (like the reconstruction loss), the FID compares the mean and standard deviation of one of the deeper layers in a convolutional neural network named Inception v3. These layers are closer to output nodes that correspond to real-world objects

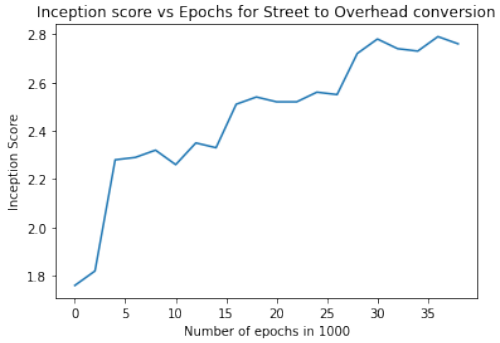

Figure 6. Inception score increasing vs epochs for Street to Overhead conversion in pix2pix

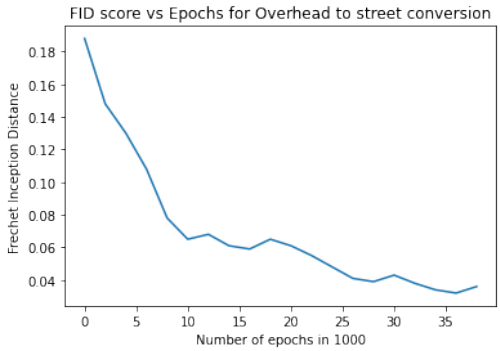

Figure 7. FID score decreasing vs epochs for Overhead to Street conversion in pix2pix

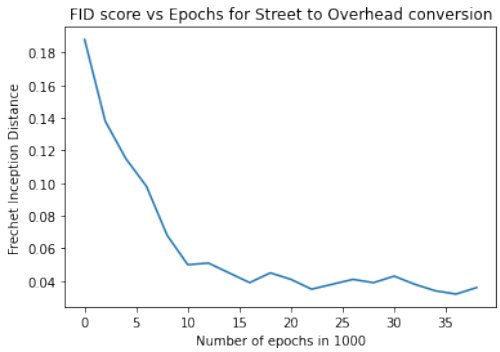

Figure 8. FID score decreasing vs epochs for Street to Overhead conversion in pix2pix

such as a specific breed of dog or an airplane, and further from the shallow layers near the input image. As a result, they tend to mimic human perception of similarity in images.

We obtain the results shown in Figure 7 and 8:

References

[1] P. Isola. pix2pix: Image-to-image translation with a conditional gan. https://www.tensorflow.org/tutorials/generative/pix2pix, 2017 (last update January 26, 2022). 2

[2] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks, 2016. 1

[3] N. Vo and J. Hays. Localizing and orienting street views using overhead imagery, 2016. 1