Developing GAN for Image-to-Image translation

Kaushal Rai
Department of Computer Science
University of Wisconsin-Madison
kkrai2@wisc.edu

Ansh Jain
Department of Computer Science
University of Wisconsin-Madison
jain98@wisc.edu

Abstract

In this project, we develop a Generative Adversarial Network (GAN) for the task of Image-to-Image translation. We use the conditional GAN (cGAN) variant because unlike traditional GANs the output, in this case, is not generated from a random noise vector, but is dependent on the input image. During the course of this project, we also study different GAN architectures and experiment with different parameters and training setups to gain an in-depth understanding of the cGAN. Further, we train a neural style transfer architecture for the same task and present the result. We try to understand whether it is possible to get acceptable results on the task of converting aerial satellite images to map images using style transfer.

1 Introduction

We analyze the hypothesis that generative models trained through an adversarial process achieve appreciable results for Image-to-Image translation. The proxy used is the performance of cGANs on the maps dataset that converts satellite images to their corresponding map views. This setting is a subset of the Image-to-Image task and hence can be used to evaluate the hypothesis. We further compare the adversarial model, with another algorithm called the style transfer. The style transfer algorithm is chosen, as it performs a similar operation of changing the representation of an image while keeping the content intact. Therefore it seems intuitive that this algorithm can perform sufficiently well on the maps dataset. During the course of this project, this claim is also analyzed, in addition to our main hypothesis and the results are compared to the outputs generated using GANs.

GANs are powerful neural network architectures, built via an adversarial process (zero-sum game) between a generative and a discriminative network. We model a conditional GAN (cGAN) [8] that generates output conditioned on an input image rather than random noise. The outcome of such a model is an output image with similar semantics, but in a different domain as compared to the input image.

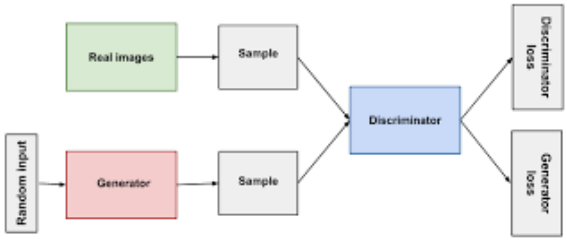

Figure 1: GAN Architecture [1]

We focus on the task of converting an Image from one domain to another through GAN. For the purpose of training and testing, we use the Pix2Pix dataset [5]. Pix2Pix dataset includes input and output images coupled together as a single image. The dataset includes multiple groups of mapped images such as label to the street scene, aerial to map view, label to the facade, black and white to color, day to night, etc. We have implemented the model for aerial to map view images and reported the results for it. The dataset includes 2194 images for aerial to map view with the size of each image being 600*1200*3.

Figure 2: Satellite to Map View Image in Pix2Pix [5]

To establish the hypothesis mentioned above, we present the results generated by GANs Furthermore, to showcase the superiority of the algorithm we also try to compare its results with a neural style transfer (NST) architecture [3]. NST refers to generating images or videos that have a visual style of another image. Style transfer has a wide range of applications for example creating artificial artwork from photographs. We assume that the task of converting a satellite view to a map view is similar to "imposing" a map style on a satellite image and hence the results should be acceptable. We imported an NST model, trained it on pix2pix, and compared its results with the ones generated using GANs. This comparison, however, could not be completed as the NST architecture could not perform well on the translation task and outputs poor results. This is discussed in depth in the subsequent sections.

2 Related Work

Generative Adversarial Networks are a relatively new discovery in machine learning but have been hugely successful for a variety of applications including photo-realistic image generation, data augmentation, super-resolution, image-inpainting, and image-to-image translation. This adversarial training model for a generative network was introduced by Goodfellow *et al.* [4] in 2014 and since then, many different versions have been developed by various researchers. Some of these include CycleGAN [9], super-resolution generative adversarial network (SRGAN) [6], conditional-GAN (cGAN) [8], Laplacian Pyramid Adversarial Framework[2] etc. The SRGAN [6] is used to create high-resolution images for up to 4 times upscaling factor. They model their training algorithm using 2 losses i.e. an adversarial loss (to push the output to be close to natural looking) and a content loss (to encourage perceptual similarity between input and output). Another variation is called the CycleGAN [9] model is developed to generate a mapping from domain X to Y when paired examples from the corresponding domains are not available. This is different from our project as we have the corresponding pairs available for all the images in our dataset.

For this project, however, we use a variation of cGAN which is introduced in the paper [5] by Isola *et al.*. The paper presents a general-purpose cGAN capable of achieving acceptable results on a wide range of Image-to-Image translation tasks. We develop the model from scratch and analyze its performance on the "maps" dataset, which contains satellite images and their corresponding map views. The input to the GAN is conditioned on the satellite image to produce a photorealistic map view. The detailed method and results are explained in the subsequent sections.

In addition to GANs, we also model the task of pix2pix conversion by using the concept of style transfer. We consider one of the map views as the "style" and transfer that style to different satellite

images. The similarity in the task of Image-to-Image translation and Style Transfer is intuitive as the output image can be considered similar to the input image in content but with a different style. The semantics of the input and output images do not change, only the view domain and representation are translated. In the project, we also analyze this hypothesis and present our findings. The algorithm used here is from the work of [3]. We use the implementation of this algorithm from the following code repository https://github.com/leongatys/PytorchNeuralStyleTransfer.

3 Method

To evaluate our hypothesis we develop a cGAN model using Keras and train it on the maps dataset. The details of the architecture are explained in the subsections below. Further, we also delineate the architecture of the Neural style transfer model and compare the results with GAN.

3.1 cGAN Model

Conditional GAN is a variation of the traditional GAN, in which the input to a generator is conditioned on an input image, rather than random noise. The objective of this kind of architecture can be written as follows:-

$$\min_G \max_D L(G, D) = E_{x \sim p_{data}(x)}[\log D(x|y)] + E_{z \sim p_z(z)}[\log(1 - D(G(z|y)))] \quad (1)$$

This objective is similar to traditional GAN with the difference, that the generator and the discriminator are fed data conditioned on some "y". The "y" in our case would be the corresponding satellite image. This enables us to control the output and not obtain just random images.

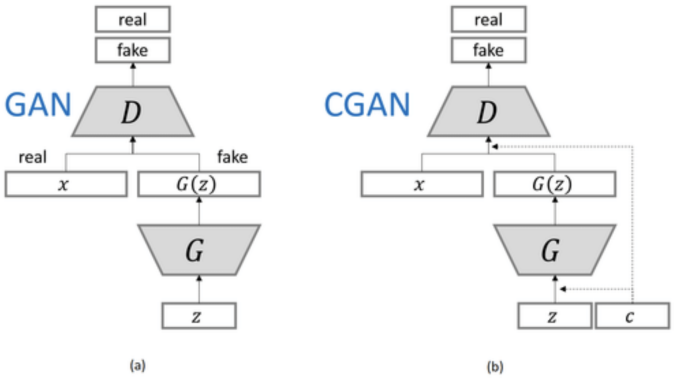

Figure 3: GAN and CGAN [7]

Specifically, for our implementation, the discriminator consists of six sequentially concatenated convolution layers. We use a fixed kernel size of 4*4 across all convolution layers with a stride of 2*2. Batch normalization is used after each layer (except first) to standardize the inputs. We use a leaky Relu activation function after each layer (with alpha 0.2). The loss function used is binary cross-entropy loss along with Adam optimizer. Another important aspect of the discriminator is that we use a patchGAN architecture. This type of model outputs a class (real/fake) for a local patch instead of an entire image. This works on the assumption that pixels separated by more than the patch diameter are independent of each other.

The generator involves concatenating a group of encoders followed by groups of decoders. For developing our generator we follow the U-net architecture [5] shown in Fig. 3. U-Net is a special type of encoder-decoder that includes "skip" connections between mirrored layers in encoder and decoder. These connections allow low-level information to shortcut across layers. We use 7 layers each in the encoder and decoder phase. The output of the encoder passes through the Relu activation function before getting into the decoder. The output of the decoder uses the TanH activation function

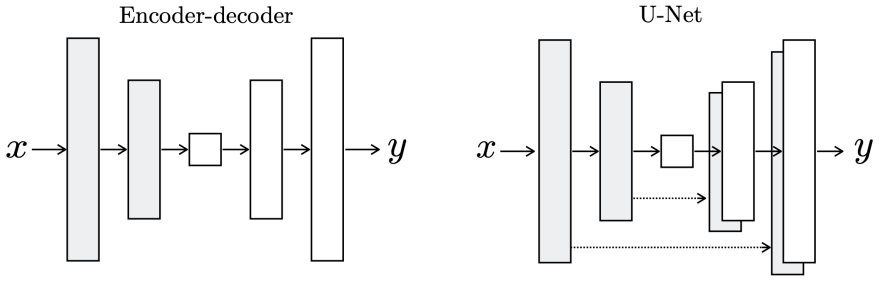

Figure 4: Generator Architectures [5]

to generate the final result. Each encoder has a convolution layer followed by a Leaky Relu layer. Each decoder layer has a transposed convolution layer followed by a batch normalization layer. To experiment with different types of encoder and decoder, we have included an encoder variant having a batch normalization layer and a decoder variant having a dropout layer. Another important aspect of our implementation is that during the training of the generator, the final loss is a weighted average of two separate losses i.e. L_{real} and L_{target}. The loss L_{real} indicates how close to a real image the output is, whereas the value L_{target} informs about the fact that how close the generated output is to the target value. The final loss for training the generator is therefore of the form:-

$$L = w1 * L_{real} + w2 * L_{target} \tag{2}$$

For our algorithm, we use w1 as 1 and w2 as 100. We run this architecture for 40 epochs and report the results for it.

3.2 Neural Style Transfer

To get the results of Neural Style Transfer on the task of satellite to map view, we imported the model implemented by Gatsy et al. in their research [3]. The authors have implemented a 19 Layer VGG Network, having 16 convolutional and 5 pooling layer. The network was normalized to ensure that the mean activation in each filter over images and position remains one. None of the fully connected layers was used and an average pool operation instead of max pooling was used because it gave better results for image-related tasks. The model was then trained with a randomly selected satellite and map image pair. The satellite image acted as the "content image" and the map image acted as the "stylized image". We ran 1000 iterations in this setting and generated the final resultant image.

4 Experimental Setup

4.1 Hypothesis

In this project, we examine the hypothesis that Generative Adversarial Networks achieve satisfactory and quality results for Image-to-Image translation tasks. To examine its superiority, we also compare the results with the output from Neural Style Transfer (NST). The underlying assumption here is that the NST algorithm was developed for a similar task and can be used to convert images from one domain to another while keeping the content the same. This assumption is also scrutinized in the project.

4.2 Proxy

To establish the hypothesis, we develop the conditional variant of a GAN architecture and train it on the pix2pix dataset. The dataset contains different satellite views and their corresponding map images. This is a type of Image-to-Image translation task and therefore is suitable to analyze the hypothesis. Further, we use the NST architecture from [3] to train on the same task and check the results.

4.3 Expected Results

- GAN - We expect the model to converge after sufficient training to a stage where the map views generated are realistic and showcase the same information as to their corresponding satellite views. Due to the limitation of resources, we train the model till we achieve an acceptable result that is close to the ground truth and evaluates the results with the test set. We also experiment with different GAN setups as listed in the results section to find the best configuration. Among the different setups, we expect the PatchGAN architecture to have the best performance as theoretically it should capture high-frequency features and produce a sharper image.
- NST - This algorithm requires a "style" which is overlaid on the "content" images. For example, a "night" style could be transferred to an image taken during the day so that the image appears to be taken at night but does not change the content otherwise. In our scenario, we expect the results to have the same behavior if we consider the map views as style and satellite views as the corresponding content images. The output should be close enough to the ground truth so that it can be used to compare with GAN.

5 Results

We perform the following experiments for this project and present the results:-

- Results for cGAN Architecture
- Results for cGAN plus patchGAN architecture with 16*16 patches
- Results with L_{target} as MSE loss
- Results with L_{target} as MAE loss
- Results with NST Architecture

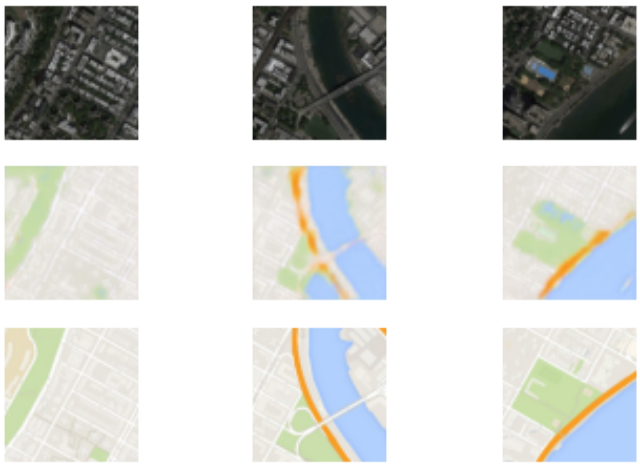

Figure 5: Generated outputs (the first row represent satellite images, the bottom row is the target map views and the middle row contains output generated through our algorithm) for PatchGAN and cGGAN Architecture with MAE Loss

Fig. 5 shows the result generated by the cGAN + patchGAN architecture ran for 20 epochs. The top row shows the satellite images, the bottom row shows the ground truth map view images and

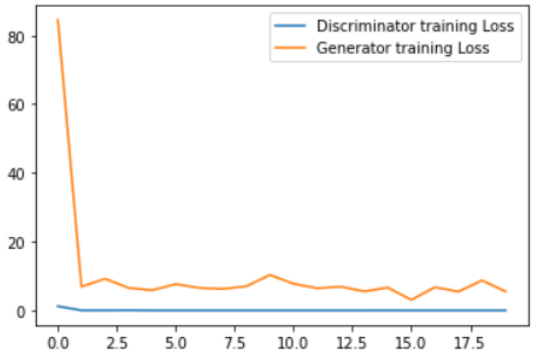

Figure 6: Learning curve cGAN and patchGAN (MAE Loss) architecture

the middle row shows the image generated by the model. As evident, the model can distinguish the boundaries between different segments in the satellite view such as land, water, road, etc. Due to training hardware limitations (it took about 20 hours to run 20 epochs), the boundary lines in the images can be seen to be a little hazy. But it is evident from the images generated in previous epochs, that the model will be able to generate realistic images if the epochs are increased to 50. The Root Mean Square Error (RMSE) for PatchGAN + cGAN architecture was 0.1278. The learning curve for the architecture is also shown in Fig. 6

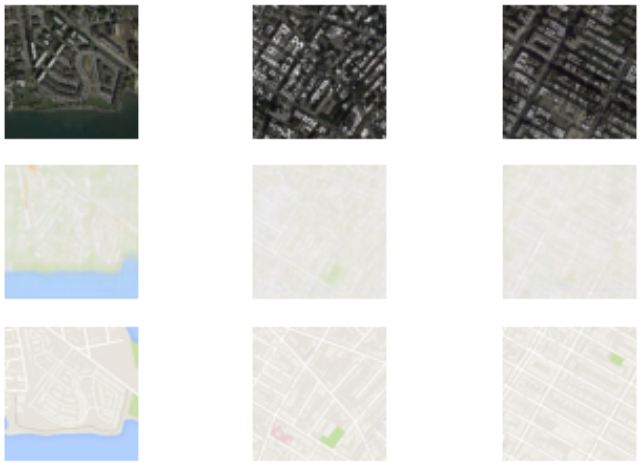

Figure 7: Generated outputs (the first row represent satellite images, the bottom row is the target map views and the middle row contains output generated through our algorithm) for PatchGAN and cGGAN Architecture with MSE Loss

We also experimented with training the generator with mean squared error loss (MSE) instead of mean absolute error loss (MAE). The image generated by MSE is shown in Fig. 7. The RMSE error generated on the following configuration was 0.1283, which wasn't significantly different from the RMSE with MAE loss. However, the quality of images was much better with MAE Loss as compared to MSE Loss. The learning curve for MSE loss is shown in Fig. 8.

We also tried to experiment with a non-patchGAN architecture, although the outputs are blurry, as seen in Fig. 9, and leads to a higher RMSE score. This is because this architecture is unable to

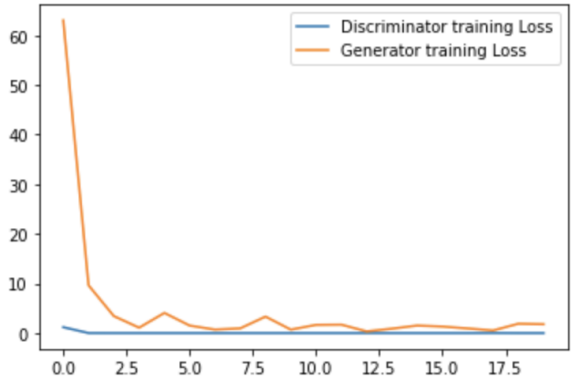

Figure 8: Learning curve cGAN and patchGAN (MSE Loss) architecture

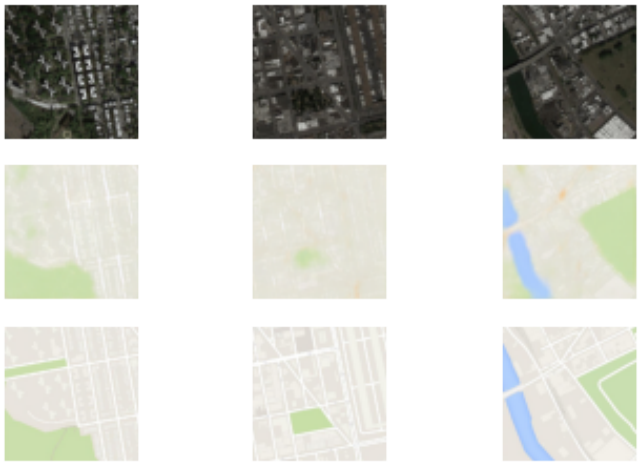

Figure 9: Generated outputs (the first row represent satellite images, the bottom row is the target map views and the middle row contains output generated through our algorithm) for non-PatchGAN Architecture

capture high-frequency features. This model variation calculates a single real/fake label for the entire image rather than at the patch level. The learning curve for it is shown in Fig. 10.

Fig. 11 shows the result generate by running the neural style transfer algorithm on satellite to map view images. We imported the model defined by [3] and added the map view as a "stylized image" along with the street view as the "content image". However the image generated, as expected, is unable to identify the boundaries in the image and their appropriate color scheme.

Algorithm	RMSE
PatchGAN + CGAN (MAE Loss)	0.1278
PatchGAN + CGAN (MSE Loss)	0.1283
Non-PatchGAN + CGAN (MAE Loss)	0.1289

Table 1: Table showing RMSE for various architectures experimented.

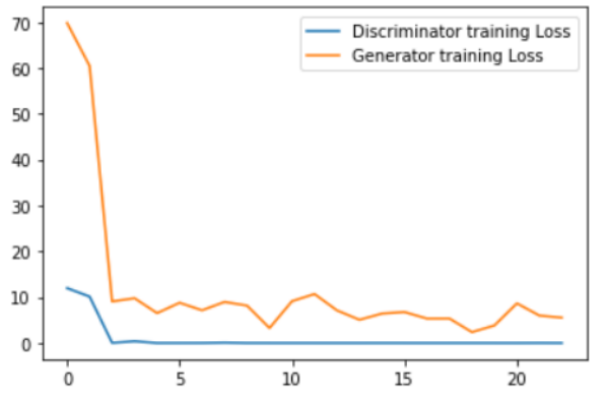

Figure 10: Learning curve for non-patchGAN architecture

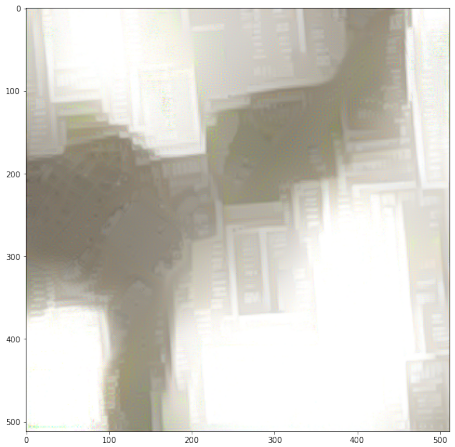

Figure 11: Output Generated by NST Architecture

6 Conclusion

In conclusion, the experimental results presented in the report support our hypothesis that GANs perform well for the Image-to-Image translation task. We even try different variations of the architecture and summarize the RMSE values in 1. As evident from the generated images, PatchGAN + cGAN with MAE loss gives the best result. The PatchGAN variation helps in reducing blurriness in the generated images thereby generating a sharper image.

Furthermore, we find that the NST model fails to give a good result for the task(Fig. 11). This behavior might be due to the fact that we are choosing a random map view as a "style" for multiple satellite images. This is in contrast with GANs, where we are learning by comparing each "generated map image" with its corresponding "ground truth map image". Further, the style transfer algorithm chosen by us works only for a pair of images (1 style and 1 content) and would fail to learn the semantics of the content (such as segmentation of water, land, buildings, etc).

In the future, other variations can also be analyzed that might improve the performance of GANs even further. Also, the shortcomings of the NST algorithm, we believe, can be overcome by changing the architecture to give appropriate results. Hence, this project leaves a lot of scope for future study and experimentation.

References

[1] https://developers.google.com/machine-learning/gan/gan_structure, 2019.

[2] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. *arXiv preprint arXiv:1506.05751*, 2015.

[3] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[5] 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.

[6] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[7] Ajkel Mino and Gerasimos Spanakis. Logan: Generating logos with a generative adversarial neural network conditioned on color, 2018.

[8] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[9] 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.