Image to Image Generation using Conditional GANs

Rohan Varadkar[1], Mohnish Karri[2], Sourabh Singh[3]

[1-3] *Information Technology Engineering, Padmabhushan VasantDada Patil Pratishthan's College of Engineering, Maharashtra, India*

---***---

Abstract – *We are looking into conditional Generative Adversarial Networks as a general-purpose solution to image-to-image generation problems. In this approach, the networks learn to map features themselves from a given input image to an output image. We are going to convert a google earth view image to its corresponding google maps image. Thus, we'll be implementing this approach on pairs of images, where the first image acts as a condition to map to the second image. And thus, we use conditional adversarial networks for doing so. This approach can be applied to various different types of image pairs and so we no longer have to hand-engineer our mapping functions and achieve good results.*

Key Words: **GANs, Generator, Discriminator, Google maps images, Google earth view images, U-net Encoder Decoder Architecture, Conditional mapping**

1. INTRODUCTION

In computer science, there are a lot of image related tasks. And one of the most difficult one of them is translating an input image to a corresponding output image. Generative Adversarial Networks allow us to do the same. Using GANs or Conditional GANs we can translate one given input image into its corresponding output image. The community has taken a lot of steps in Convolutional Neural Networks (CNNs) which act as a very common and powerful framework for image prediction problems. Even though CNNs only help in predicting images, they form a crucial part for GANs, since one of the tasks of the adversarial network is image prediction (fake image vs real image). So essentially what CNNs is to image prediction, GANs or conditional GANs is to image translation. CNNs learn to minimize a loss function, so a lot of hard work is required for designing effective losses. If this loss to be minimized is not designed correctly, the neural network might end up giving incorrect results. So, even after the learning process of Convolutional Neural Networks being automatic, a lot of hard work and expertise is required for designing the loss function effectively. Now, when GANs come into picture, they learn the loss themselves that tells whether the output image is fake or real, while at the same time training the model to minimize this loss. GANs have been studied a lot in the last few years, and in this paper, we use GANs in conditional setting i.e., cGANs or Conditional GANs to output a google maps image from a google earth view image.

1.1 Literature Survey

In the first paper "Generating Adversarial Examples with Conditional Generative Adversarial Network", they have proposed two methods to generate images to attack a trained DNN classifier, namely cGAN-F and cGAN-Adv. The cGAN-F can output the attack images in one stage, while the cGAN-Adv in two stages. The experiments conducted showed the possibility of generating images directly by using GAN, and such images can be misclassified by the classifier while being correctly classified by humans. Compared with previous work, their work produced attack images more efficiently.

In the second paper "Image Edge Detection based on Conditional Generative Adversarial Nets", they developed the quality of reconstructing objects from edge detection. They used Bijection for optimization so that they could solve the issue of the pix2pix models where it cannot guarantee that all samples in X and all samples in Y are reasonably corresponding. In conclusion, they observed bijective mapping to improve the overall performance of the model.

In the third paper "A Survey of missing data imputation using Generative Adversarial Networks", they proposed global and local context discriminator and both of them were trained to distinguish between real and fake images. The approach used CNNs when data type was image and fully neural network when data type was numeric. In conclusion, their model filled the missing hole of the image by copying surrounding similar patches.

In the fourth paper "Generative Adversarial Network for Deblurring of Remote Sensing Image", the approach was to obtain a sharper frame of images to resolve the blur images to give better quality images. The proposed method is a kernel-free blind deblurring learning method, which can avoid the inaccuracy brought by the prior blur assumption. The experiments showed the proposed method can give better results than other methods.

In the fifth paper "GAN-D, Generative Adversarial Network for Image Deconvolution", this network mentioned in the paper restores visual data from distorted images, applied to multiple dominant degradation problems such as noise, blur, saturation, compression etc. It outperformed the ODCNN on public benchmark datasets. GAN-D produces high quality deconvolution results.

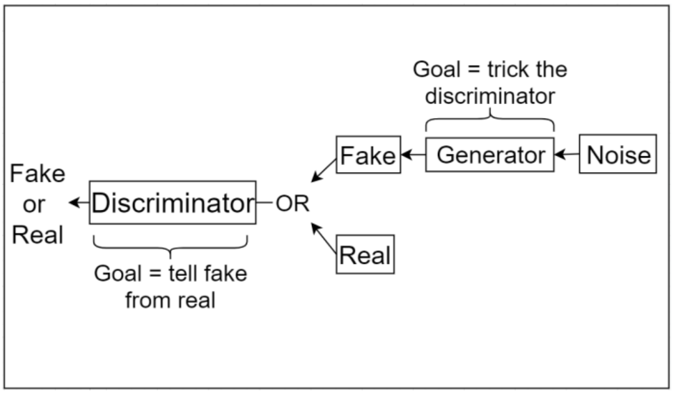

Figure 1: GANs Architecture

2. METHODOLOGY

GANs are generative models that learn a mapping from random noise vector z to output image y, G: z => y [6]. Contrary to this, conditional GANs learn from an additional image x and noise z, i.e., G: {x, z} => y. The Generator *G* generates images(fake), then Discriminator *D* looks at those images generated by the Generator and also looks at real images. By this process, the Discriminator *D* learns about the image, and also at the same time provides feedback to the Generator *G*. This training procedure is mentioned in Figure 1. The dataset of maps is obtained from Kaggle.

3. OBJECTIVE

To compile and train a model which could translate a given google earth view image to its corresponding google maps image using Conditional Generative Adversarial Networks.

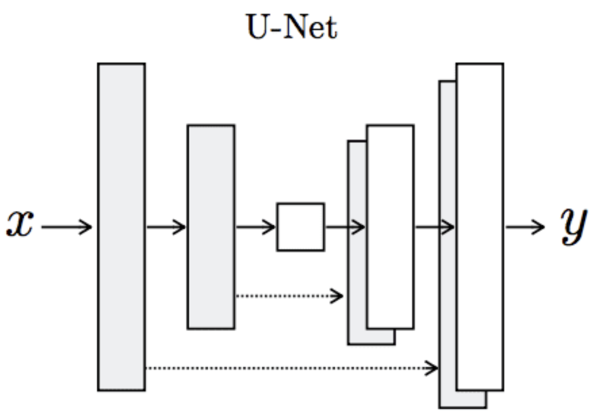

Figure 2: U-Net Generator Architecture

4. PROPOSED SYSTEM

The architecture consists of two models namely the Generator and the Discriminator. The Discriminator and the Generator both use the same structure which includes the Convolution layer => Batch Normalization => ReLU Activation function.

4.1 Discriminator

Discriminator is a Deep Convolutional Neural Network (CNN) that performs image classification. Also, in our case this CNN performs condition-image classification. The inputs to the discriminator are google earth view image (source image) and its corresponding google maps image (target image). The job of the Discriminator is to tell the likelihood of the target image being a real or fake translation of the input google earth view image. The Discriminator model being used in this approach is 70*70 PatchGAN Discriminator model. The advantage of using PatchGAN is that it can be applied to input of different sizes. The current size of the input image we are providing to the discriminator is 256*256 pixels. Also, the output of the model is dependent on the size of input image, so as to keep it consistent we convert every image to 256*256 pixels and then process it. This model is directly trained on real and fake(generated) images.

4.2 Generator

The Generator is based on U-Net Architecture. This U-Net allows low-level information to shortcut across the network and thus, produce high quality results. This model only takes in one input i.e., google earth view (source) image and produces a google maps (target) image. This U-Net help generator by first encoding the image and reducing its size and then decoding it to the size of the output image. Thus, it is an Encoder Decoder based U-Net Generator model. In this U-net, skip connections are added between the encoding and corresponding decoding layers, thus forming a U-shaped structure. This U-Net Architecture is depicted in Figure 2. Generator is trained with the help of Discriminator. It is updated in such a way that it minimizes the loss predicted by the Discriminator for generated images marked as "real". Thus, it motivates the Generator to generate more plausible images. The Generator is also updated to minimize the L1 loss between real image(target) and fake image(generated). So as the Generator can achieve better results i.e., produce more believable target images.

4.3 GANs Composite Structure

This model involves the connection of the Generator model and the Discriminator model. The Generator is placed on top of the Discriminator. An earth view (source) image is provided as input to Generator and Discriminator both and the output of Generator is given to Discriminator as the corresponding google maps (target) image. The discriminator then does its job of predicting the likelihood of the image being a real translation of the source image. This structure is mentioned in Figure 3.

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 04 | Apr 2021 www.irjet.net p-ISSN: 2395-0072

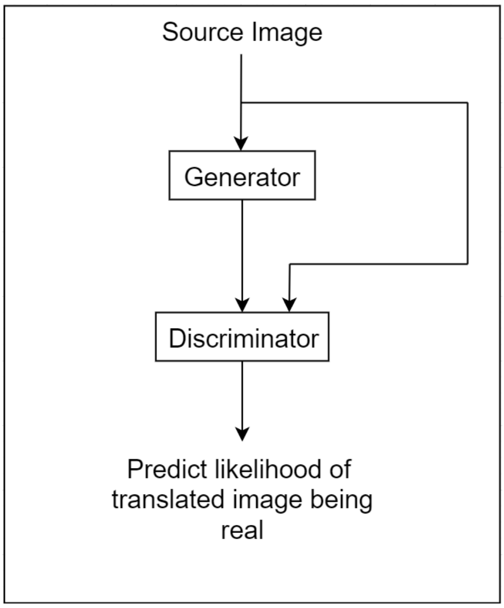

Figure 3: GANs Composite structure

Figure 4: Output samples after 1st epoch

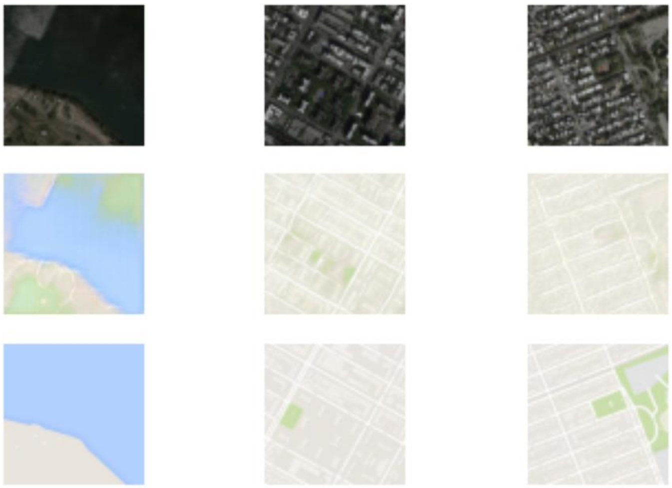

Figure 5: Output samples after 21st epoch

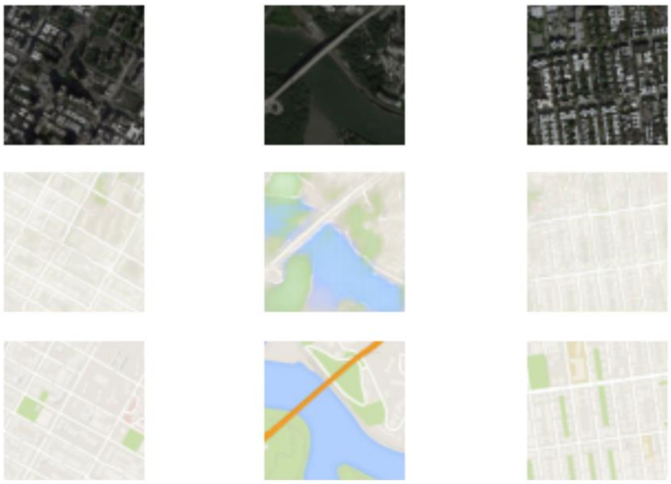

Figure 6: Output samples after 45th epoch

5. RESULTS

The output samples generated after different epochs is depicted in figures 4, 5 and 6. Each figure consists of 3 random samples. In each figure, the first row consists of the source (google earth view) image, the second row consists of corresponding target images generated by the Generator and the third row consist of corresponding target images (real google maps images). As we can observe, the samples generated after our very first epoch are way too blurry and aren't plausible at all. But, after training for a few more epochs the target image quality improves. The output samples obtained after the 21st epoch are way better than the output samples after the first epoch. And, the samples after the 45th epoch are even better and more meaningful, for example, big structures like garden areas, roads, marine areas are at accurate enough places, although the border joining them isn't exactly straight.

6. FUTURE SCOPE

We could train our model for more epochs to check whether the output quality improves or remains same with only but very minor improvements. We will use this approach of Conditional Generative Adversarial Networks over various other types of inputs. There are various different types of GANs out there, we can explore them and compare them with our current approach.

7. CONCLUSION

Conditional Generative Adversarial Networks are great for Image-to-Image generation problems. They can adapt several patterns for different inputs without hand-engineering mapping functions and achieve great results.

REFERENCES

[1] Ping Yu, Kaitao Song, Jianfeng Lu, "Generating Adversarial Examples with Conditional Generative Adversarial Net"; (ICPR 2018)

[2] MINGYUN HE, YULUN WU, XIAOFANG LI, JINYI LIU, XIAOFENG GU, "Image Edge Detection Based on Conditional Generative Adversarial Nets"; (International Centre for Wavelet Analysis and Its Applications, University of Electronic Science and Technology of China)

[3] Jaeyoon Kim, Donghyun Tae, Junhee Seok, "A Survey of Missing Data Imputation Using Generative Adversarial Networks"; (School of Electrical Engineering, South Korea)

[4] Yungang Zhang, Yu Xiang, Lei Bai, "Generative Adversarial Network for Deblurring of Remote Sensing Image"; (Yunnan Normal University, China)

[5] Ha Yeon Lee, Jin Myung Kwak, Byunghyun Ban, Seon Jin Na, Se Ra Lee, Heung-Kyu Lee, "GAN-D: Generative Adversarial Networks for Image Deconvolution"; (Korea Advanced Institute of Science and Technology, Korea)

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, "Generative adversarial nets"; (NIPS 2014)