Department of Computer Science

Reconstruction of Maps from Satellite Image

CS713 – Applied Machine Learning

Guided By:

Dr. Howard Hamilton

Prepared By:

Meet Patel - 200467513 – MDP965@uregina.ca

Vedant Patel - 200469201 – VPP336@uregina.ca

Overview

Generation of maps and enhancement of geographical maps are a very valuable and time-consuming process. In every domain, there are multiple applications that extend the reliability of maps i.e, especially in commercial-like E-automobile companies, food or meal supply chain, E-commerce logistics and intelligence agencies (like CIA, RAW, and KGB). This human-computer interaction has many use cases; geospatial data collected by UAVs, imaging aircraft, or satellites are used to build more precise maps and update them on the ground in a timely manner which is associated with human-readable maps. But still, there is considerable latency between actual geographic/street views and publicly available human-readable maps. We have automated the task of converting a satellite image into a human-readable map which is a good cause to reduce this latency. It can be achieved by a generative adversarial network. In this work, the dataset of pix2pix by EECS, UC Berkeley consists of 1096 concatenated satellite images with their respective map image. The Conditional GAN model is trained on this focused and multispectral image dataset where model performance is enhanced by reciprocating the generator and discriminator as per requirements.

Dataset Description

The dataset is collected by UC Berkeley researcher Taesung Park using Google Maps API to get both satellite images and map images from the New York City Park. The dataset contains two directories named train and val. It contains 1096 concatenated satellite images corresponding to their map image. The actual image size is 1200x600 which is later preprocessed and converted into 256x256 dimensions.

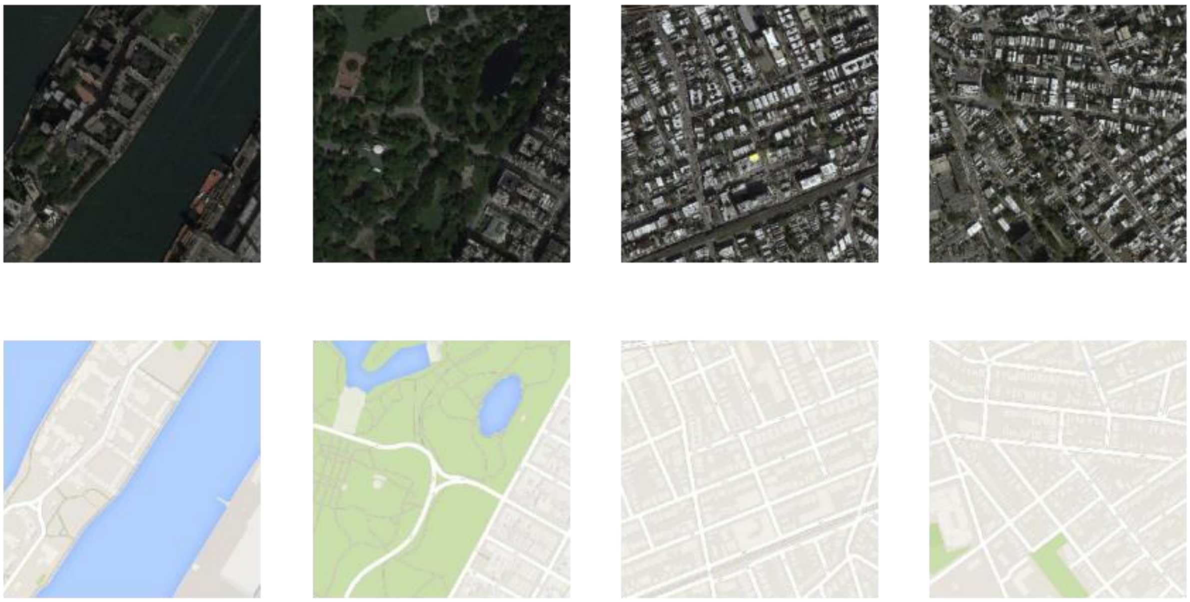

Figure: Pre-processed Data

The first row contains the satellite image and the second row has their respective target map image. The road and multiple noise objects are challenging tasks for the model to classify a real or fake image generated by the generator.

Generative Adversarial Networks (GAN)

Generative: Generates Fake Data

Adversarial: Generator and Discrimintor will compete with each other to generate data which looks real. Basically, Generator will try to fake and Discriminator will try not to be fooled by generator.

Network: We use deep convolution networks

The generator will take random noise (latent vector) as input and generate fake images. We need to generate images such that discriminator won't differentiate them as fake images.

We train discriminator which discriminates or differentiates between fake and real images. We give both real and fake images as input to the discriminator, and discriminator will act just as binary classifier which classifies whether given image is real or not. Then it will generate discriminator loss which is used to make robust discriminator by minimizing discriminator loss. Generator loss goes to generator which makes our generator robust by minimizing generator loss.

Goal of Generator: Maximize the probability of discriminator making a mistake
Goal of Discriminator: Maximize the probability of identifying real vs fake images correctly.

Benefits of Using Conditional GAN (cGAN) over Normal GAN

- GAN model just generates random images from noise.
- Relationship between latent space (noise) and generated images is hard to map.
- Using, cGAN we can train generator model with class label or other image modalities (Image-to-Image translation).

Working of Conditional GAN for Image to Image Translation

The generator will take a real satellite image as an input to generate a fake map image. Here, our Generator is based on U-net architecture which is used for image segmentation but for our use case and to get better results, we have modified it. Now that fake image is concatenated with a real satellite image. In discriminator, we pass two pairs of images one pair contains generated fake images of a map with its real satellite image and another pair contains a real map image with a satellite image. The discriminator will classify whether our generated map image is real or not for a given real satellite image. The discriminator will return three losses, Discriminator loss, Generator loss(GAN loss), and L1 loss. Discriminator loss is the difference between real and fake images for the discriminator, generator loss is a binary cross-entropy loss. L1 loss is new here, which is a MAE that says how much generated map image is fake from the original map image. We combine L1 loss with GAN loss in a 100:1 ratio.

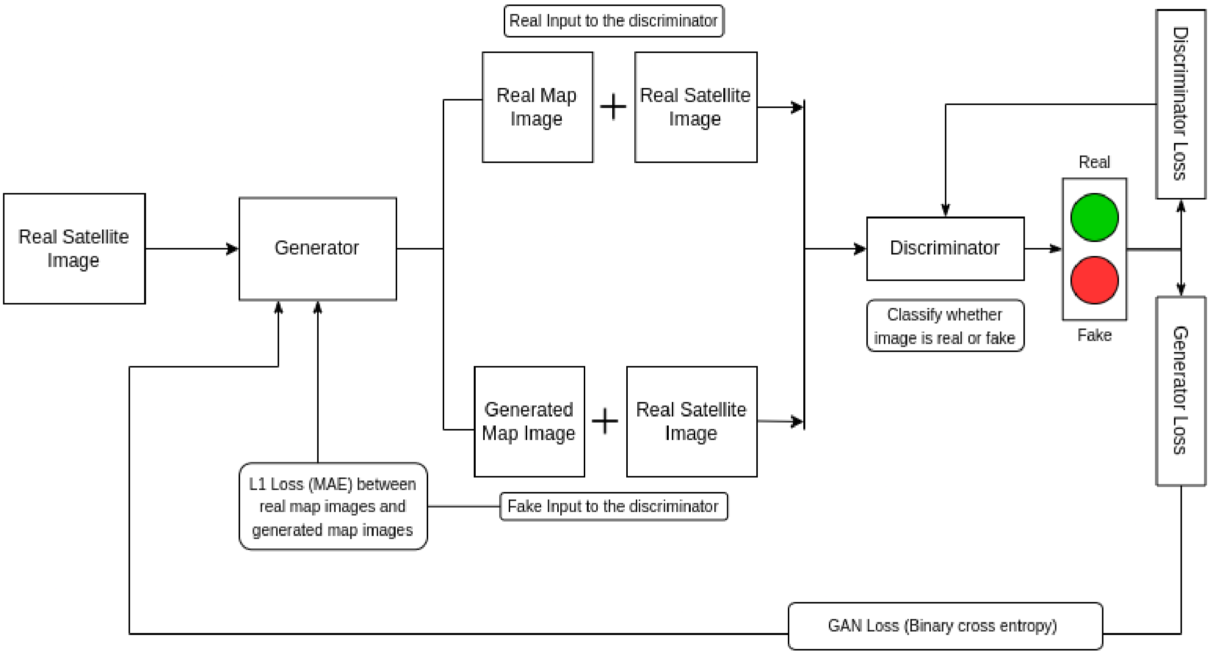

Figure: Implementation of cGAN

Implementation Details

All the required libraries are imported to explore, visualize, transform, train and predict. The dataset has been downloaded having size of 238.65MB and unzipped the tar file. In the exploration of dataset, we observed there are 1096 images having average dimension of 1200x600 in the train directory that we would use further for training the model. Firstly, we created architecture of generator in the form of function based on U-net model. Our generator function is divided into two parts encoder and decoder. Encoder consists of convolutional layers with filters 64, 128, 256, 512 x 4 respectively along with BatchNormalization and Leaky ReLU activation function [c64 - c128 - c256 - c512 - c512 - c512 - c512]. Likewise, decoder comprises transposed convolutional layers with filters 512, 512, 512, 512, 256, 128, 64 respectively along with BatchNormalization and Dropout layer by concatenating encoder layers [cT512 - cT512 - cT512 - cT512 - cT256 - cT128 - cT64]. The discriminator model is built using convolutional layers with filters 64, 128, 256, 512, 512, 256 and output layer respectively followed by BatchNormalization and Leaky ReLU activation. Now, we combined the both defined models as gan and compiled with Adam optimizer and binary crossentropy as loss. Two new functions for generating original data and dummy data has been defined, later it was used in result outline helper function in which we have visualized the generated results along with source and target images. The most important training image function carry out the training of images by calculated steps using batch per epoch and input the defined models. It calculates the discriminator and gan loss per batch; also stores the loss in a list by appending at every step and summarize the result by calling helper function. The models are defined and data has been preprocessed by rescaling the image to [-1, 1]. After the training, the trained model has been loaded and results are predicted using the preprocessed validation data. The losses are converted into dataframe and average losses are evaluated.

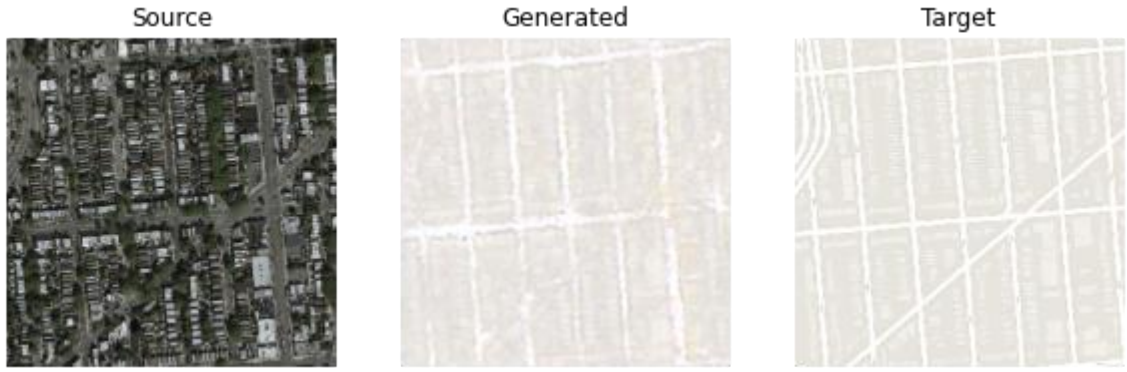

Figure: Predicted result from validation dataset

Evaluation

For evaluating our conditional Generative Adversial Network we have used three losses as follows:

1. Discriminator Loss (L2)
2. Generator Loss (GAN Loss)

3. L1 Loss

Discriminator Loss: It is generated by evaluating the real and fake images classified by the discriminator model.

L1 Loss: MAE between real map images and generated map images.

Generator Loss (GAN Loss): It is generated by evaluating the combined model of generator and discriminator models.

Following are the graphs of generated loss over every training step. We have trained our model over 10 epochs which results in more than 10000 training steps.

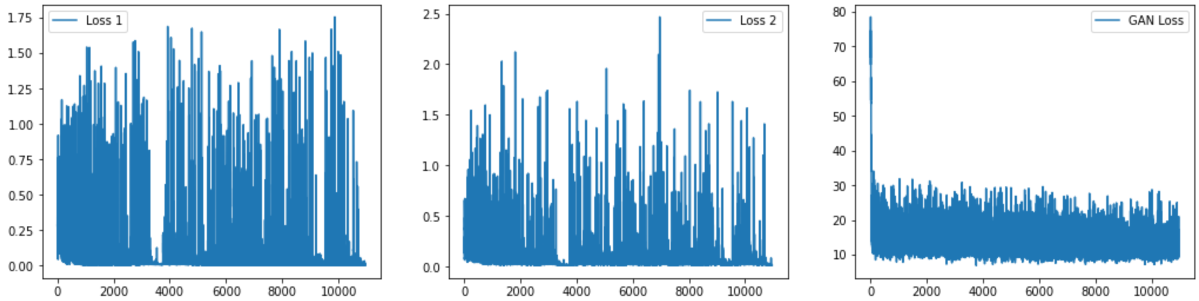

Figure: Loss Graphs

Type of Loss	**Average Loss**
Loss 1 (L1)	0.084
Loss 2 (L2)	0.088
GAN Loss (GLoss)	13.181

Table: Average Loss

Ideas adopted from earlier assignment and how w extended it.

We have taken idea from our GAN assignment where we have generated images of digits from MNIST dataset. We have extended our learnings from that assignment of generating images and created this project with advanced GAN i.e. conditional GAN where we instead of sending just random noise as input we are sending real data to generator so that the generator will know that which type of data it has to generate. In this project, we have used pixp2pix dataset of satellite and map images from EECS, UC Berkeley. We have read several research papers on Image-to-Image translation and generating maps from satellite images.

Our creative contributions to the project

In this project, we have proposed a new modified discriminator which works better than directly using architecture of PatchGAN without modifying it. We have modified the existing PatchGAN to make our discriminator more robust such that it classifies fake and real images in much better way and reduces loss.

Conclusion

We have utilized conditional generative adversarial network to generate map images from satellite images. Our generator will take satellite image as input and it will generate map image. We have used U-NET architecture for our generator model which is used for image segmentation and proposed a new discriminator model by modifying existing PatchGAN architecture.

References

1. Image to Image Translation with Conditional Adversial Networks
2. Comparing GANs for Translating Satellite Images to Maps
3. GeoGAN: A Conditional GAN with Reconstruction and Style Loss to Generate Standard Layer of Maps from Satellite Images