Image to image translation: Satellite to street maps

Chao Long
c3long@eng.ucsd.edu

Eli Uc
euc@eng.ucsd.edu

Haakon Hukkelaas
haakohu@stud.ntnu.no

Peter Greer
pbgreerb@eng.ucsd.edu

Stian Rikstad Hanssen
hanssen.stian@gmail.com

Abstract

Generative Adversarial Networks are a novel class of promising generative models and have been rapidly adopted in the field of Artificial Intelligence. Recent improvements to GAN's have shown great potential in overcoming the hurdles with the adversarial approach, and in this project we will experiment with several of these methods. We will investigate the power of the conditional GAN in mapping satellite images to street maps. With the initial model from the Pix2Pix model, we experiment with different training methods, network structures, loss functions and it's effectiveness on a new dataset.

1 Introduction

Translating satellite images into street maps is a challenging task with many real world applications. The task is far from solved, however with deep learning we've seen a great improvement. With the popularity with Generative Adversarial networks, and the inspiration from Phillip Isola et. al "Image to Image translation" paper[10] we will explore the power of a conditional GAN applied to this problem. We will explore on two datasets, one taken from the original Image-to-Image paper [4], and one based from the video game "Grand Theft Auto 5" which is a dataset we've not yet seen used in this task.

2 Related work

2.1 Image-to-image translation

Many researchers have leveraged adversarial learning for image-to-image translation with the purpose of translating an input image from one domain to another domain given input-output image pair as training data[19]. The adversarial loss has become a popular choice for many image-to-image task due to its adaptedness to the differences between the generated and real images in the target domain[19]. The recent pix2pix framework we base our experiment on uses conditional GANs(cGANs) for many applications like transforming Google map to satellite images[10].

2.2 Conditional GANs

Generative adversarial networks are an example of generative models and have been vigorously studied in the last three years[10]. Just like GANs learns a generative model of the data, cGANs learn a conditional generative model and this makes cGANs very suitable for image-to-image translation tasks[9] and a probabilistic one-to-many mapping[15]. In prior and current work cGANs mainly focus on discrete labels, text and images[17] guided by constraints[2], style transfer[12], and superresolution[13]. Several papers use unconditional GAN for image-to-image mappings and rely

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

on other regularization terms to force the input to be conditioned on the input[10], such as L1 loss. Unlike past work, Phillip Isola et. al uses cGANs for Image-to-image translation based on a "U-Net" architecture for generator and a convolutional "PatchGAN" classifier for discriminator[10].

2.3 Satellite to areal images

The task of mapping satellite images to areal images has been a challenging task, and a huge variety of methods has been applied to this challenge. Mni and Hinton showed promising results back in 2012 by applying a standard neural network to label roads in a satellite image[16]. Further with more "conventional" neural networks we have Marcu et. al that apply a dual-stream neural network for object detection in satellite images [14]. They show impressive results in segmentation, and multi-class labeling of pixels. They also acknowledge the difficulty of segmentating and understanding satellite images due to poor lighting, low image quality, occlusion and high degree of variations in objects structure and shape, even for high-performance deep neural networks.

Costea et. al shows the effectiveness of GAN's on this task by segmenting satellite images into black and white road maps, and introduces the model they call DH-GAN. [8]. The model creates both segmented maps, as well as a road graph that can be used for any graph search or matching algorithm.

3 Dataset

3.1 Grand theft auto (GTA)

The first dataset is high resolution satellite, road and aerial maps for the game Grand Theft Auto 5, that are sliced into smaller images, obtained from[1]. Since large portions of the maps are nothing but water we preprocessed the data, and saved sections that displayed significant features such as roads. In the preprocessing we also generated more data by rotating, and mirroring the data.

We chose this dataset for a few reasons. Since the "satellite" pictures were actually rendered from a game world we know that the corresponding overlay images generated from the same game world are extremely accurate. The images closely resembles real images, however slightly simplified making them a good starting point. Lastly, to our knowledge nobody has used a GAN to reproduce road and aerial maps for GTA5.

It is important to note that despite these positive properties, the dataset also pose a few challenges. A big flaw in the dataset is how underground roads are also shown on the road map. This means we will train the network on cases that are impossible for it to get correct. One can argue this is a useful property when transitioning over to real maps, as roads will also be obscured by objects such as trees. However, the very purpose of starting with this dataset is to start with easier cases and increase difficulty later. A second cause for confusion is how backgrounds change on the road map based on area. For example, cities have gray background. At this point, this is unnecessary as our biggest objective is to highlight roads.

3.2 Pix2pix

The second dataset we trained and tested our model on was the satellite to aerial map dataset used in the pix2pix[10] paper, an example shown in Figure 1. We obtained this dataset through one of the pix2pix authors project web pages[4] on the UC Berkeley engineering website. Images in this dataset were 600 x 600 pixels large and each sample included an input satellite image & target map image that used solid colors and outlines to identify roads, buildings, urban areas and bodies of water. In total the dataset contained 2194 samples, 1096 designated for training and 1098 designated for validation.

Due to the large image sizes and much greater number of samples to work with our ability to test this dataset was limited significantly by time constraints. In an effort to save time we trained and tracked data on the 1096 training samples, but only used a subset of varied validation sample outputs to track how well our model was generalizing. This setup wasn't ideal, but it gave us more opportunity to experiment with different parameters and settings which produced better images in the end.

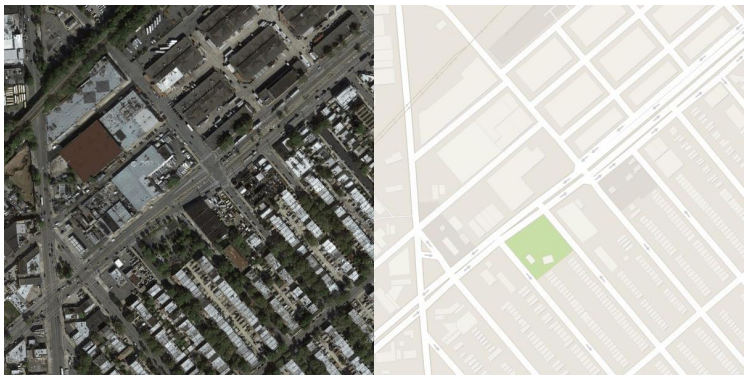

Figure 1: Sample image from dataset used in the paper "Image-to-Image Translation with Conditional Adversarial Networks"[10]

4 Method

The objective of a conditional GAN can be expressed as

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}_{x,y}[logD(x, y)] + \mathbb{E}_{x,z}[log(1 - D(x, G(x, z))] \tag{1}$$

where G tries to minimize the objective, and the discriminator tries to maximize it. We also introduce an additional error term to the generator in form of L1 or L2 distance. We will experiment with both, and previous experiments show to that L2 tend to more blurring versus the L2 distance [11]. Our final objective can then be expressed as

$$G^* = arg \min_G \max_D \mathcal{L}_{GAN}(G, D) + \lambda \mathcal{L}_{L1}(G, y) \tag{2}$$

where λ is the factor of L1 loss to be propagated to the generator, and $\mathcal{L}_{L1}$ the L1 distance between the generated image, and the target image y.

4.1 Generator

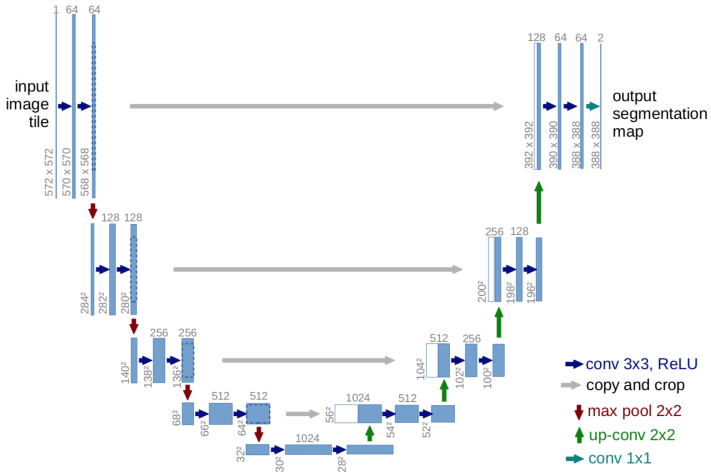

Figure 2: U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.[18]

For the generator we used a U-net model[18] with little adjustments, code was based on Pix2pix model[10]. The model of the U-net is shown in Figure 2. The original model in pix2pix they use a 5x5

constitutional kernel, while we have chosen to use a 3x3 kernel to reduce the expensive computation. Instead of maxpooling we are using a stride of 2, as suggested in [3]. The number of blocks we used was mostly somewhere between 5-7, giving us a minimal image resolution of $32x32$ and $128x128$, respectively.

A conditional generator is a model that learns a mapping from a random noise z, and a conditional input x. $G : \{x, z\} \rightarrow y$. However as others have noted, the generator learns to ignore the noise[11]. Therefore the noise we have in the generator is from dropout layers, which is present in each U-net block.

4.2 Discriminator

The discriminator is a model that learns the mapping from a input image x, and the target image y, to the prediction p $D : \{x, y\} \rightarrow p$. The prediction p predicts if the target image y is a real image, or a generated image. For our model we used a simple discriminator, as given in Table 1.

Type	Number of features in	Number of features out
Conv2d	6	64
Leaky ReLU	–	–
Conv2d	64	128
BatchNorm	–	–
Leaky ReLU	–	–
Conv2d	128	1

Table 1: Simples discriminator model, the Pixel discriminator based of Pix2Pix [10]

4.3 Progressive GAN

Karras et. al[5] introduced a very efficient and stable way of training GAN's with progressive growing networks and images. They start with a low resolution image, and progressively increasing the resolution by adding layers to the networks. Inspired of their results, we used a progressive image size on our training set, where we started with a 32×32 image resolution, and progressively increasing the resolution. This significantly improved training time, as well as the quality of our images.

We experimented with progressively adding layers to our networks as well, however experienced a lot of problems to get this properly working with the U-net. Having the structure it has, we added a U-net block at the bottom for each increase in size, and initialized it with Xavier normal. However this lead to the network generating completely gray images, and lead to an exploding gradient in the generator.

4.4 PatchGAN

PatchGAN is a discriminator architecture that penalizes structure at the scale of patches. This discriminator tries to classify if each N×N images is fake or real[10]. Since every neuron in our output convolutional layers learns a patch of input images. So If we change the number of layers in our PatchGAN. The receptive field(N×N) for our PatchGAN will change. In our experiments, we tried to minimize the output convolutional layers in our original model and explore the result of different PatchGANs.

5 Results

5.1 Evaluation of results

Evaluating image quality is one of the harder areas of a generative adversarial networks, and what we desire from the generated satellite images is little noise, accurate mapping of objects in the image such as houses and roads, and matching colors. We have used a L1 loss to give us a sense of the quality of the image, however the L1 can be misguiding, and give us a poor representation of the image quality. This can be seen in Figure 3. Even though the images has close to equal L_1 loss, the

Figure 3: Comparison of images with L1 loss of $L_1 = 0.08$. Image to the left is the target image, middle image is after 120 epochs, and rightmost image is from 150 epochs of training.

Figure 4: Generated images after 660 epochs

quality of the images are clearly different, where the middle is more accurate on color, and has less noise.

5.2 GTA

Training on the GTA set was more challenging then expected, as mentioned in subsection 3.1. By starting on a resolution of 32×32, then scaling up by 2 times after each 200 epoch, we achieved the result seen in Figure 4. The generator is clearly showing the capability of recognizing color background difference between cities and nature, and clearly recognizing where the roads are. The city image is a complex image to analyze, and the generated image has a lot of noise. However we do believe this is a very well generated image as it is clearly recognizing roads, and coloring the area filled with houses gray. The upper image the roads area clearly segmented, and it also recognizes that it is two separate roads that do not fuse together.

Figure 5 shows the loss of the generator, discriminator, and the L1 loss versus the target image. The Generator loss and the discriminator loss is very stable throughout the whole training period, and shows a balanced adversarial game. We notice after about 100 epochs the generator is clearly starting to overfit to the training data, and the test loss is not improving much in the area of $100 - 200$ epochs. The spikes in epoch $200, 400, 600$ is where the image resolution is doubled. For each of the spikes the generator is quickly reducing loss, however we do notice the overfitting here as well. For these spikes the L1 loss to the test set is reduced significantly, and it is stopped on a L1 loss of about 0.1.

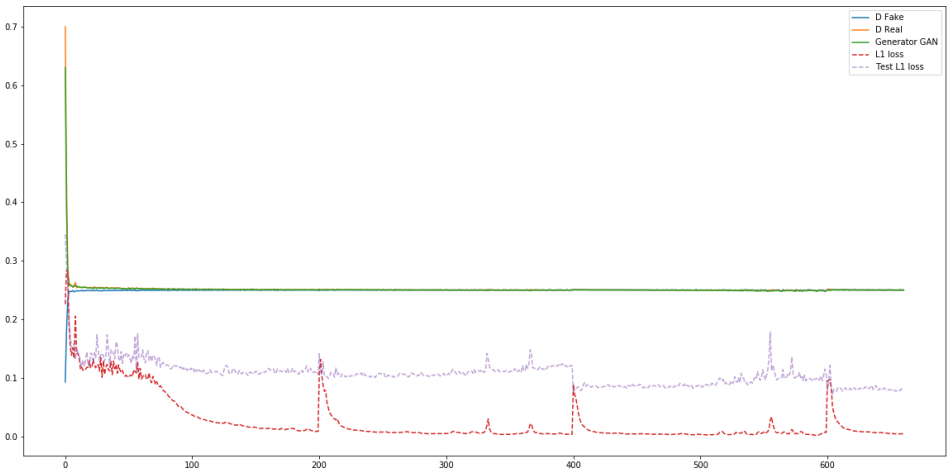

Figure 5: The loss for the generator, discriminator, and the L1 loss for train/test against the target image. Training stopped after 660 epochs.

Figure 6 shows the result of PatchGAN. We almost use the same method as one-pixel discriminator except for applying MSE loss and the fourth convolution layers for PatchGAN architectures in Phillip Isola et. al's paper. We can see the generated images show roads and coastline clearly. Although there are still a few noises, we do believe the generator creates meaningful images.

Figure 7 shows the loss of the generator, discriminator, and the L1 loss versus the target image for PatchGAN. The Generator loss and the discriminator loss is very stable throughout the whole training period, and shows a balanced adversarial game. We observe the generator is clearly starting to overfit to the training data at about 20 epochs, and the test loss is stable after 50 epochs. The spikes in epoch $200, 400, 600$ is where the image resolution is doubled as before.

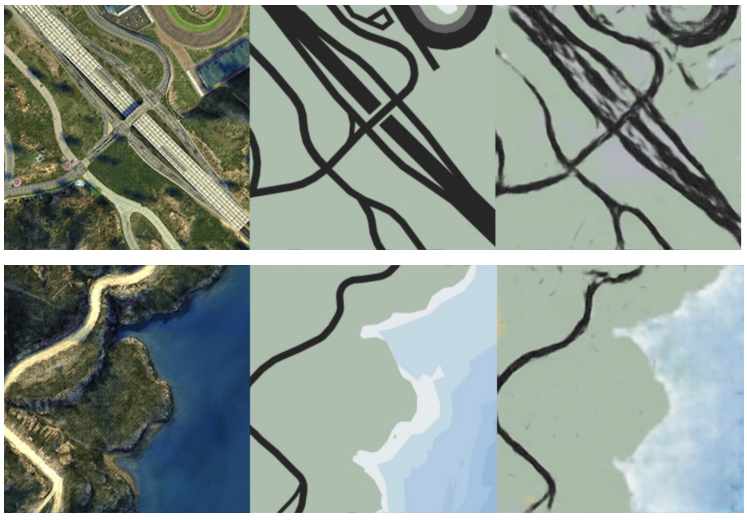

Figure 6: Generated images after 600 epochs(8×8 PatchGAN)

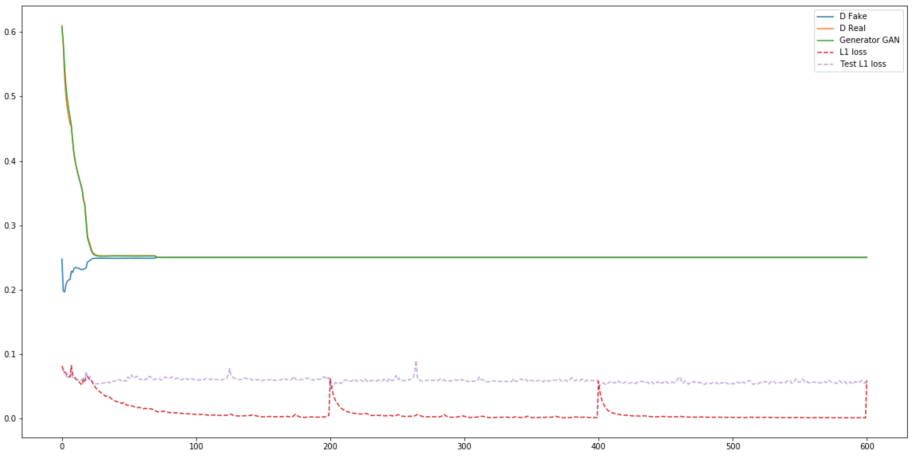

Figure 7: The loss for the generator, discriminator, and the L1 loss for train/test against the target image. Training stopped after 600 epochs(8×8 PatchGAN).

5.3 Pix2pix

Time constraints limited us from truly optimizing our model for the pix2pix dataset, but the results we obtained are still very telling about strengths and weaknesses of our model. We can see from figures 8 and 9 that our model excels at mapping out urban environments with standard city planned blocks. The general structure of the output image in these figures was recreated nearly perfectly, excluding small aberrations in the roads which could likely be improved with further model optimization and training. The model also generally displayed no issue detecting large bodies of water due to their distinctive dark blue color that rarely appeared on land in the training set.

Figures 10 and 11 highlighted the shortcomings of our model, which could on the whole be described as a weakness to portraying natural terrain. To a certain degree this is to be expected, because the target images provided to us have built into them arbitrary rules on what zones deserve an urban gray background and what zones should be displayed as green. Knowing this it is easy to understand why figure 10 generated splotches of green in urban areas covered by trees, because the model learned to associate natural green colors with non-urban areas. However, we can see these trees not only produced green but also partially covered roads or darkened them with their shadows, which significantly hindered the generators ability to draw straight roads.

We can see in figure 11 that the model reacted extremely poorly when roads or even bodies of water intrude into natural terrain. Not only were large sections of the road entirely missed due to being obscured by trees, but the few sections that were visible were drawn poorly and inaccurately. To make matters worse the lake in figure 11 blends into the background with the assistance of tree shadows in a way that only leaves a few splotches of blue in the generated image to give off a vague impression that there may be a body of water somewhere. We found similar issues in other results such as figure 12 where the road along the coast was completely ignored by the generator despite being clearly visible. From what we've seen we attribute this to the way the road naturally curves with the coast and blends in with the rocks, but really this shows us just how bad our model can be at identifying roads in more natural environments.

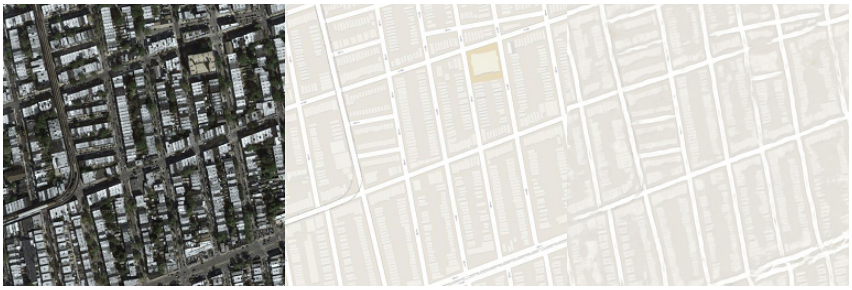

Figure 8: Urban Environment 1

Figure 9: Urban Environment 2

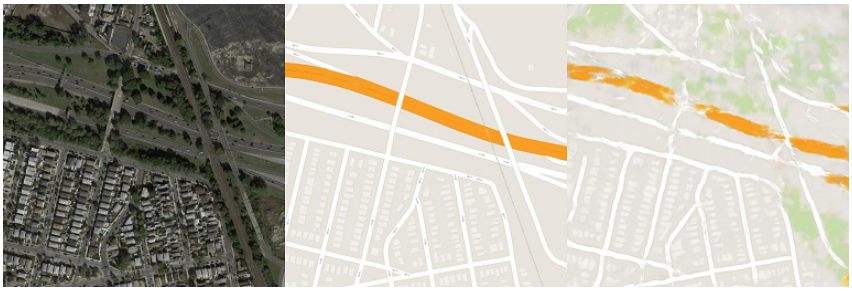

Figure 10: Trees in an Urban Environment

Figure 11: Forest Environment

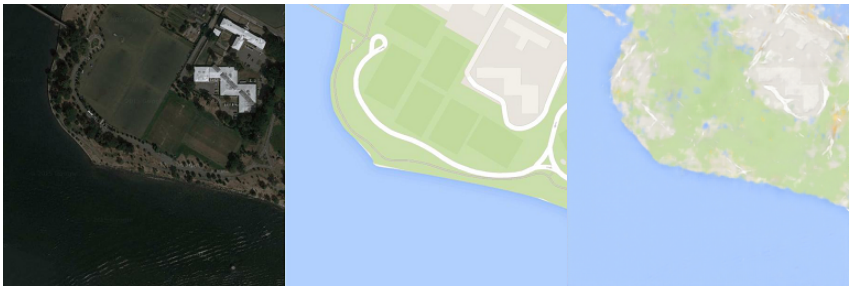

Figure 12: Road along a Coast

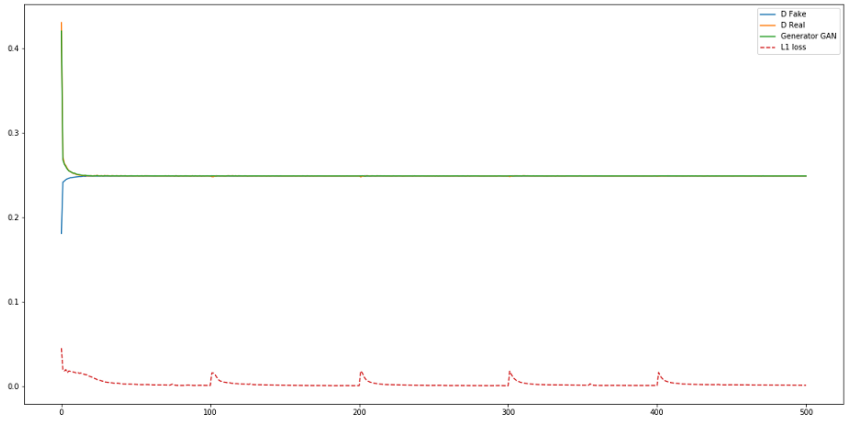

Figure 13: 500 epochs of Pix2Pix training statistics, used to produce results in figures 8, 9, 10, 11, 12

6 Further work

A model we have yet to test on our problem is the Boundary Equilibrium Generative Adversarial Networks (BEGAN) [7]. These networks try a slightly different approach to the GAN currently used. The discriminator is an auto-encoder where the loss is based on how well the discriminator auto-encodes the image. The loss is derived from the Wasserstein distance [6] for training of the auto-encoder. This can be useful because this method is meant to balance the generator and discriminator during training, as well as providing a new approximate convergence measure. A BEGAN may provide more stable training, which could potentially lead to faster learning. The paper also show a way to control the trade-off between diversity and quality. Being able to tune the network towards quality would be useful in this particular case. The convergence measure would be helpful in the experimentation process and eventually for early stopping or knowing when the network has collapsed.

7 Conclusion

By using the methods of progressive growing of resolution, and the models introduced in the Image-To-Image paper, we clearly show the power of the conditional GAN being able to learn the mapping between satellite images to street maps. It is a challenging problem, especially where the images are of poor quality, has inconsistent lightning, and there is noise in the image. There is still a lot of work left, and further development to the model could be to look at improved loss functions such as the Wasserstein Loss, or improvements such as a Boundary Equilibrium GAN.

References

[1] Fourm post - high resolution maps: Satellite, roadmap, atlas. `http://gtaforums.com/topic/595113-high-resolution-maps-satellite-roadmap-atlas/`.

[2] Generative visual manipulation on the natural image manifold. `https://people.eecs.berkeley.edu/~junyanz/projects/gvm/eccv16_gvm.pdf`.

[3] How to train a gan? tips and tricks to make gans work. `https://github.com/soumith/ganhacks`.

[4] Pix2pix - maps dataset. `https://people.eecs.berkeley.edu/~tinghuiz/projects/pix2pix/datasets/maps.tar.gz`.

[5] Progressive growing of gans for improved quality, stability, and variation. http://research.nvidia.com/publication/2017-10_Progressive-Growing-of.

[6] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *CoRR*, abs/1701.07875, 2017.

[7] David Berthelot, Tom Schumm, and Luke Metz. BEGAN: boundary equilibrium generative adversarial networks. *CoRR*, abs/1703.10717, 2017.

[8] Dragos Costea, Alina Marcu, Marius Leordeanu, and Emil Slusanschi. Creating roadmaps in aerial images with generative adversarial networks and smoothing-based optimization. *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 2100–2109, 2017.

[9] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *CoRR*, abs/1406.2661, 2014.

[10] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016.

[11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016.

[12] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802, 2016.

[13] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. *CoRR*, abs/1604.04382, 2016.

[14] Alina Marcu and Marius Leordeanu. Dual local-global contextual pathways for recognition in aerial imagery.

[15] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

[16] Volodymyr Mnih and Geoffrey E. Hinton. Learning to detect roads in high-resolution aerial images.

[17] Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *CoRR*, abs/1605.05396, 2016.

[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[19] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *CoRR*, abs/1711.11585, 2017.

8 Appendix

8.1 GTA Training details

For initialization we used Xavier uniform for both the discriminator, and the generator. To optimize the Generator we used Adam Optimizer, while for the discriminator SGD optimizer was used. For each epoch the batches was shuffled, and a validation set of 80 images was kept out of the training loop to validate our model.

8.2 pix2pix Training details

To train our model for pix2pix we worked off of our approach to the GTA training set with small modifications. We still initialized our discriminator and generator using an Xavier uniform distribution, but did not shuffle the data so as to use the same training set as the pix2pix paper. We found our best results using an Adam optimizer for both the discriminator and generator. Our final results were obtained from a fixed 500 epoch test run where images were downscaled to 32 x 32 for the first 100 epochs, then 64 x 64 for 100 epochs all the way up to 512 x 512 for the last 100 epochs. Even though this progressive scaling was extremely helpful early on for reducing the training time we ran into issues with memory which prompted us to reduce our mini-batch size to 4 images.

9 Team member contribution

9.1 Stian Rikstad Hanssen

- Contributed significantly to Final Report

9.2 Haakon Hukkelaas

- Built initial model based on the pix2pix GAN
- Experimented extensively with hyperparameters, image size scaling, and network structures.
- Contributed significantly to Final Report

9.3 Chao Long

- Implemented PatchGAN for GTA dataset
- Experimented patch size and hyperparameters for networks
- Contributed significantly to Final Report

9.4 Peter Greer

- Experimented with Parameters and optimizer functions.
- Tested and produced results for the pix2pix dataset.
- Contributed significantly to Final Report

9.5 Eli Uc

- Ran tests and experimented on the pix2pix datasets
- Contributed significantly to Final Report