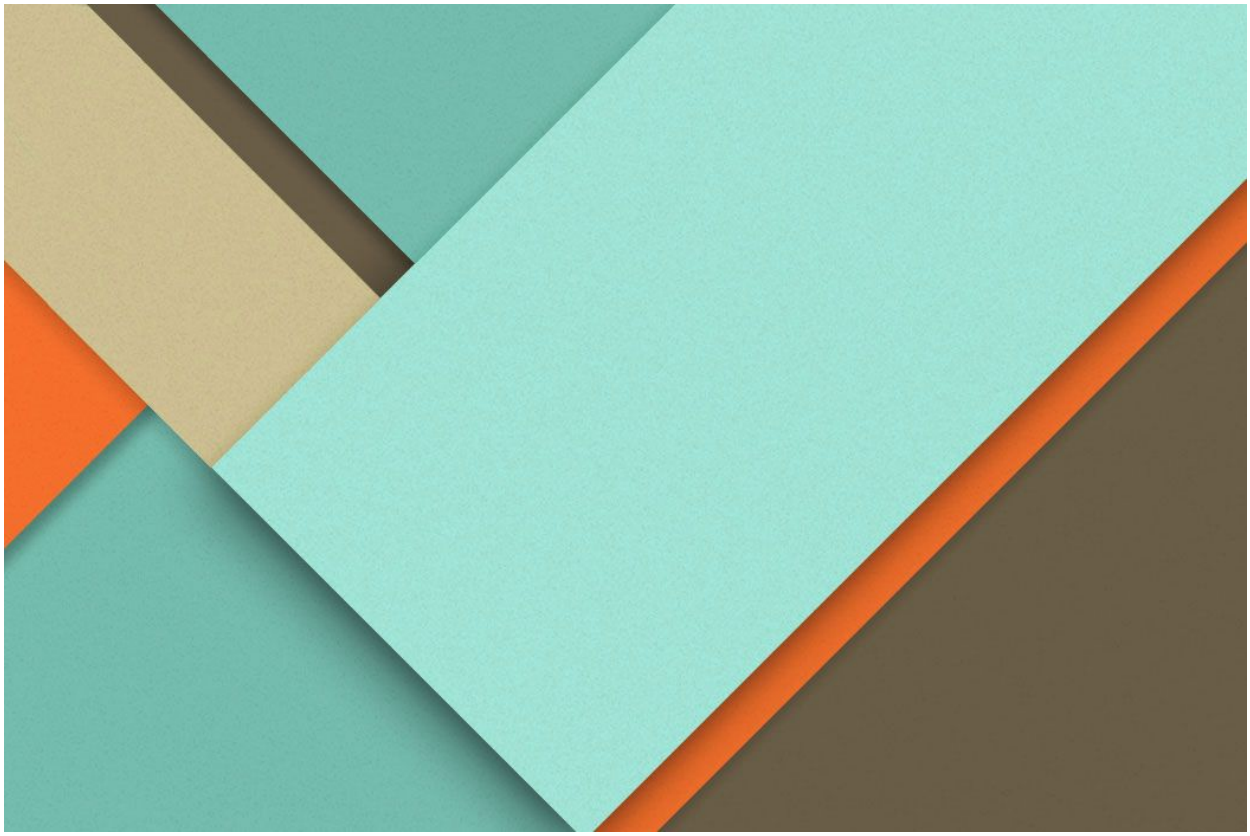


Image-to-image translation using Conditional Adversarial Networks

December 2018

Table of Contents

Introduction

Many problems in image processing, computer graphics, and computer vision can be posed as **"translating" an input image into a corresponding output image**. Just as a concept may be expressed in either English or French, a scene may be rendered as an RGB image, a gradient field, an edge map, a semantic label map, etc. In analogy to automatic language translation, we define **automatic image-to-image translation** as the task of translating one possible representation of a scene into another, given sufficient training data. Traditionally, each of these tasks has been tackled with separate, special-purpose machinery, despite the fact that the setting is always the same: **predict pixels from pixels**.

Aerial-to-map images

The community has already taken significant steps in this direction, with convolutional neural nets (CNNs) becoming the common workhorse behind a wide variety of image prediction problems. **CNNs learn to minimize a loss function** – an objective that scores the quality of results – and although the learning process is automatic, a lot of manual effort still goes into designing effective losses. In other words, we still have to tell the CNN what we wish it to minimize. If we take a naive approach and ask the CNN to minimize the Euclidean distance between predicted and ground truth pixels, it will tend to produce blurry results. This is because Euclidean distance is minimized by averaging all plausible

outputs, which causes blurring. Coming up with loss functions that force the CNN to do what we really want – e.g., output sharp, realistic images – is an open problem and generally requires expert knowledge.

It would be highly desirable if we could instead specify only a high-level goal, like **"make the output indistinguishable from reality"**, and then automatically learn a loss function appropriate for satisfying this goal. Fortunately, this is exactly what is done by the recently proposed Generative Adversarial Networks (GANs). **GANs learn a loss that tries to classify if the output image is real or fake**, while simultaneously training a generative model to minimize this loss. Blurry images will not be tolerated since they look obviously fake. Because GANs learn a loss that adapts to the data, they can be applied to a multitude of tasks that traditionally would require very different kinds of loss functions.

In this project, we explore image-to-image translation using Conditional GANs (**C-GAN**s), in which we take an input satellite image and generate the desired output google map image using GAN conditioned on the input image. Just as GANs learn a generative model of data, **C-GANs learn a conditional generative model**. This makes C-GANs suitable for image-to-image translation tasks, where we condition on an input image and generate a corresponding output image.

Data Sample

Literature Survey

Structured losses for image modeling

Image-to-image translation problems are often formulated as **per-pixel classification** or regression. These formulations treat the output space as "unstructured" in the sense that each output pixel is considered conditionally independent from all others given the input image. **Conditional GANs instead learn a structured loss**. Structured losses penalize the joint configuration of the output.

A large body of literature has considered losses of this kind, with methods including conditional random fields, the SSIM metric, feature matching, nonparametric losses, the convolutional pseudo-prior, and losses based on matching covariance statistics. The conditional GAN is different in that the loss is learned, and can, in theory, penalize any possible structure that differs between output and target.

Conditional GANs

GANs in the conditional setting have recently gained popularity. Prior and concurrent works have conditioned GANs on discrete labels, text, and, indeed, images. The image-conditional models have tackled image prediction from a normal map, future frame prediction, product photo generation, and image generation from sparse annotations. Several papers have also used GANs for image-to-image mappings, but only applied the GAN unconditionally, relying on other terms (such as L2 regression) to force the output to be conditioned on the input. These papers have achieved impressive results on inpainting, future state prediction, image manipulation guided by user constraints, style transfer, and superresolution. Each of the methods was tailored for a specific application.

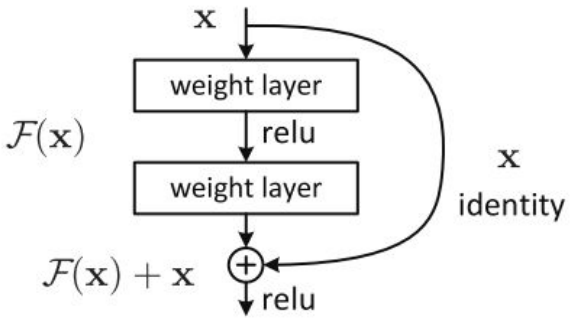
To make the GAN framework more flexible for a wide range of image translation tasks, Isola et al. proposed to use conditional adversarial network in their seminal research ("pix2pix") to learn a structured loss so that the network adapts to the tasks and data. Their work differs from the prior works in several architectural choices for the generator and discriminator. Unlike past work, for their generator they used a "**U-Net**"-based architecture, and for their discriminator they used a convolutional "**PatchGAN**" classifier, which only penalizes structure at the scale of image patches.

Residual-based Networks for images

Deep residual networks have taken the deep learning world by storm. The robustness of ResNets has been proven by various visual recognition tasks and by non-visual tasks involving speech and language.

Network depth is of crucial importance in neural network architectures, but deeper networks are more difficult to train. The residual learning framework eases the training of these networks, and enables them to be **substantially deeper**—leading to improved performance in both visual and non-visual tasks. These residual networks are much deeper than their 'plain' counterparts, yet they require a similar number of parameters (weights).

Considering a shallower architecture and its deeper counterpart that adds more layers onto it, there exists a solution to the deeper model by construction: the layers are copied from the learned shallower model, and the added layers are identity mapping. The existence of this constructed solution indicates that a deeper model should produce no higher training error than its shallower counterpart.



Residual Networks

Our Work

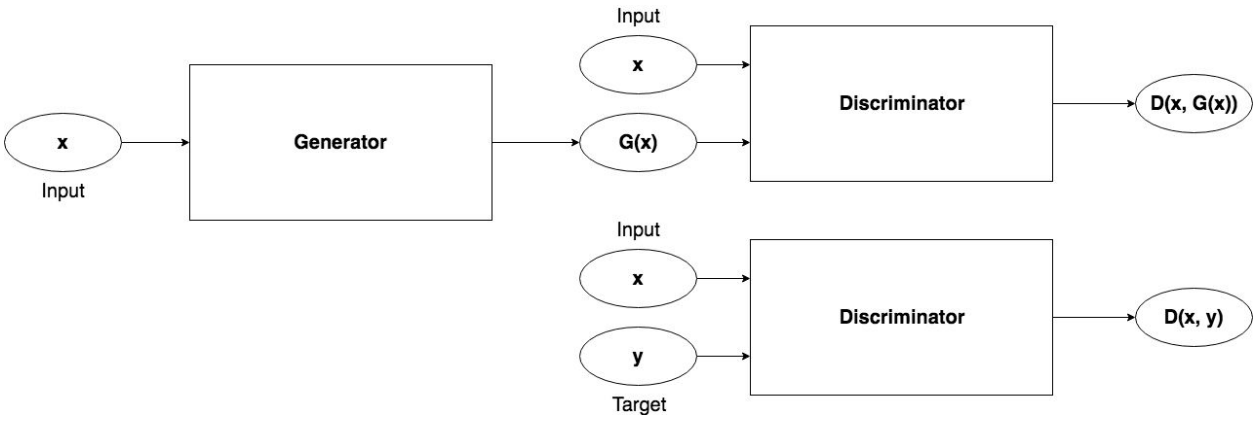
Our method differs from the previous works in architecture choices, and utilization of important ideas developed in perceptual loss, residual connections, and deep convolutional adversarial generative networks. In our study, we experimented with **residual-based network** for the generator and also investigated the effects of augmenting **random noise** to the generator's input in order to capture full entropy of the conditional distribution. We also experimented with a modified approach for training the model.

Method

GANs are generative models that learn a mapping from random **noise vector z** to **output image y**, **G : z → y**. In contrast, **conditional GAN**s learn a mapping from **observed image x** and **random noise vector z**, to **y**, **G : {x, z} → y**. The generator **G** is trained to produce outputs that cannot be distinguished from “real” images by an adversarially trained discriminator, **D**, which is trained to do as well as possible at detecting the generator’s “fakes”.

C-GAN

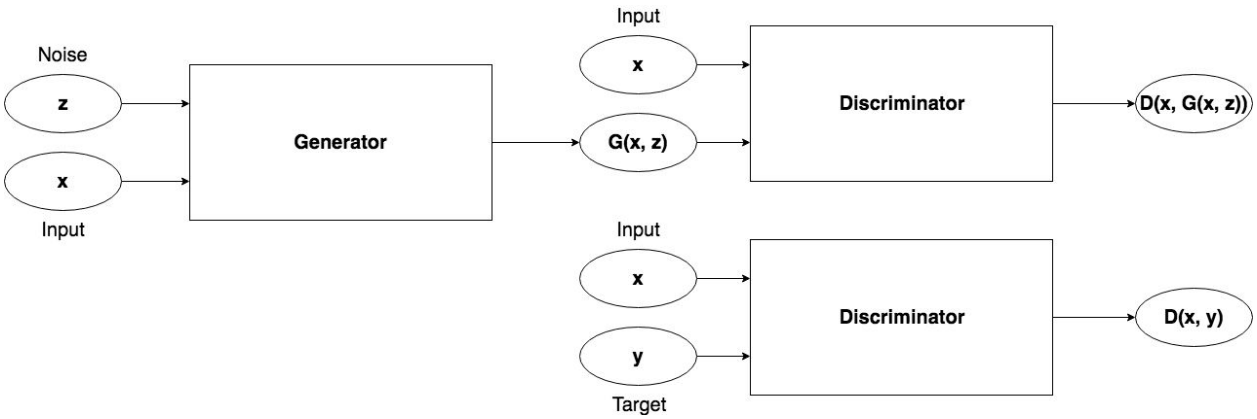
The conditional-GAN consists of **two major parts: generator G and discriminator D**. The task of generator is to produce an image indistinguishable from a real image and "fool" the discriminator. The task of the discriminator is to distinguish between real image and fake image from the generator, given the reference input image. The following figure illustrates the C-GAN architecture.



Conditional GAN

C-GAN with Noise

In this method, we augment the input of G with **randomly generated noise 'z'**. Without z, the net could still learn a mapping from x to y, but would produce deterministic outputs, and therefore fail to match any distribution other than the underlying distribution of the training dataset. Thus noise helps the model to generalize better to unseen data and capture full entropy of the underlying distribution of the data. The following figure illustrates the C-GAN architecture with noise inputs.



Conditional GAN with noise input

Loss Formulation

The objective of a conditional-GAN is composed of of two parts: ***adversarial loss*** **and** ***L1 loss***. The adversarial loss can be expressed as:

$$\mathcal{L}_{\text{cGAN}}(G, D) = \mathsf{E}_{x,y}[logD(x, y)] + \mathsf{E}_{x}[log(1 - D(x, G(x))]$$

L1 distance is added to the generator loss to encourage the low-frequency correctness of the generated image. L1 distance is preferred over L2 distance as it produces images with less blurring. Thus our full objective for the minimax game is:

$$(G^*, D^*) = \arg \min_G \max_D (\mathcal{L}_{\text{cGAN}}(G, D) + \lambda \mathcal{L}_{\text{L1}}(G))$$

Training

For each iteration during training, we alternate between one step of gradient descent on D and then G. We use **binary cross-entropy loss (BCE)** for the adversarial loss and non-saturated version of the discriminator loss. This translates to:

$$\mathcal{L}_{\text{gen}}(G, D) = BCE(D(x, G(x)), 1) + \lambda \mathcal{L}_{\text{L1}}(G)$$
$$\mathcal{L}_{\text{dis}}(G, D) = BCE(D(x, G(x)), 0) + BCE(D(x, y), 1)$$

We use **mini-batch SGD** and apply the **Adam solver** with **learning rate 0.0002**, momentum parameters **β1 = 0.5**, **β2 = 0.999** and **λ = 100**. The dropout rate for each decoder block is 0.5, and no dropout is employed for the encoder blocks. We train the model for **200 epochs** until the loss plateaus.

We choose to use the aforementioned hyperparameters because they have proven to work well with the C-GAN model developed by Isola et al. We experimented with different values of learning rate, L1 loss weight scale λ, and dropout rate; different normalization methods including batch normalization, instance normalization, and no normalization, and different initialization methods including uniform initialization, normal initialization, Xavier initialization and Kaiming initialization. In our experience the previously listed hyperparameters help our model achieve the good performance.

In our experiments we observed that in our best model, the discriminator is overpowered by the generator during the training process. Hence we experiment with training the model again with higher training rate of **0.0004** for the discriminator.

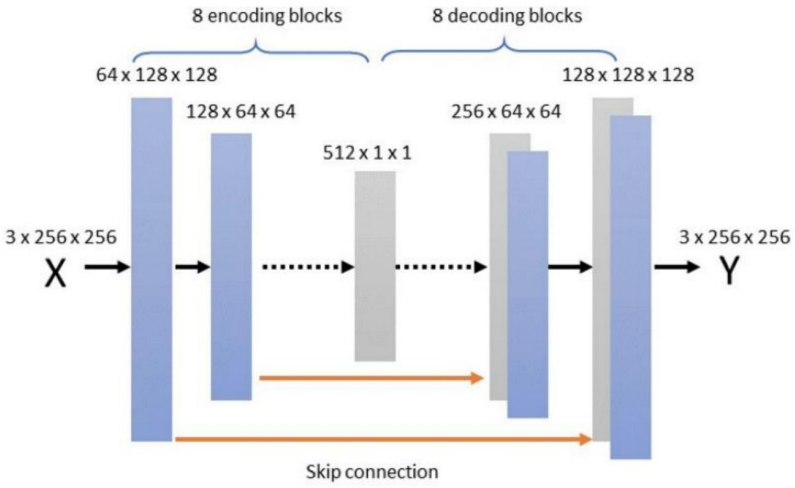
Generator Network Architecture

We present the following 2 architectures which we will be experimenting with in our study. Input to the following networks is a ***3 x 256 x 256*** image. Here, 3 is the number of channels in the image (namely - RGB) and rest of the dimensions represent size of the image. In the case when we include noise inputs in the model, the input images are of dimensions ***4 x 256 x 256***, where the 4th channel is for noisy inputs and other 3 are same as before. We only show models for non-noisy inputs in this section, the models with noise can be easily inferred from these. Noise channel values are drawn from a **uniform distribution** of real values in the range **[-1, 1]**.

U-Net

The U-Net generator is an encoder-decoder network with symmetrical long skip connections. The network consists of **8 encoding layers and 8 decoding layers**, with skip connections from layer ***i*** to layer ***n - i***, where **n** is the total number of layers. Each encoding and decoding block follows:

1. Convolution / Deconvolution
2. Batch Normalization
3. LeakyReLU



U-Net Architecture

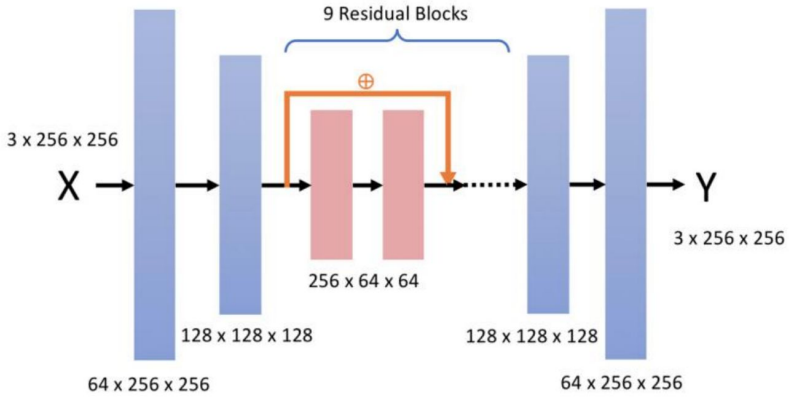
ResNet-9

The ResNet-9 generator is a residual-based network based on the ResNet model in Johnson et al. Our network is composed of **2 encoding blocks, 9 residual blocks, and 2 decoding blocks**.

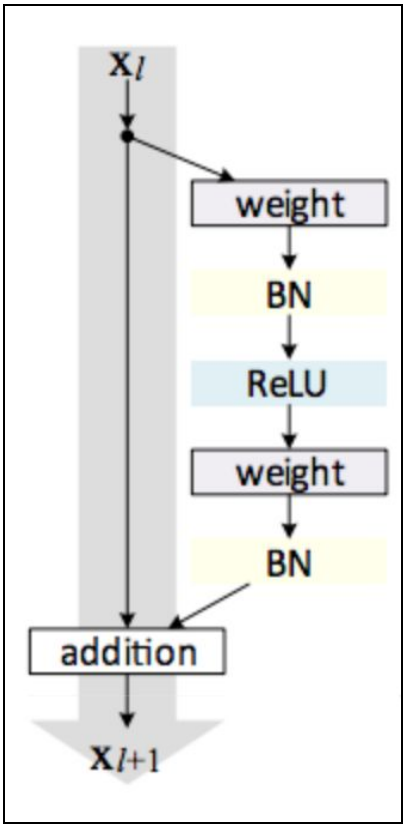
Each encoding or decoding block follows -

1. 2-stride Convolution / Deconvolution
2. Instance Normalization
3. ReLU

Each residual block follows - Convolution, InstanceNorm, ReLU, Convolution, InstanceNorm, Residual connection structure.



ResNet-9 Architecture



Residual Block configuration

Discriminator Network Architecture

Isola et al. measured the performance of various networks for the discriminator. Use pick the discriminator which performed the best in their study.

PatchGAN

We will use a convolutional "PatchGAN" classifier with architecture similar to the classifier in pix2pix as our discriminator. PatchGAN discriminator determines whether an image is real or fake by using local patches of size ***70 x 70***, rather than the entire image. The discriminator takes in two images, the input image (**x**) and the unknown image (**G(x)** or **y**), pass them through 5 blocks of the form -

1. Downsampling
2. Convolutional
3. Batch Normalization
4. LeakyReLU

Output is a ***30 x 30*** matrix, in which each element corresponds to the classification of one patch.

Dataset

We use the **aerial-to-map** dataset from the pix2pix datasets. The data are in the format of paired aerial and map view of the same region scraped from Google Maps. The dataset consists of **1097 training image pairs** and **1098 test image pairs**. Each image is composed of RGB channels and of size **600 x 600**. Dataset is obtained from the pix2pix project site.

Sample image

Data Preprocessing

The original aerial input image size is **600 x 600**. We first resize the input images to **286 x 286** due to computation constraint. Then we perform data augmentation including random cropping to **256 x 256** and random horizontal flipping. Lastly, we normalize all the image pixel values to between **[-1, 1]** for easier training.

Work Done

Experiments

In our study we investigated the performance of U-Net generator proposed by Isola et al. in comparison to the Residual Neural Network based generator GAN. We also investigate the effect of **addition of noise** to the input of the generator in order to help the generator generalize better to unseen data and capture full entropy of the underlying distribution of the data. We performed experiments with four different models for the generator in our study.

1. **U-Net**
2. **U-Net + Generator Noise**
3. **ResNet-9**
4. **ResNet-9 + Generator Noise**

Additional Experiment

As we will mention in the results section, the ResNet-9 model performs the best. Although, we observe that while training this model the generator overpowers the discriminator after a few epochs of training and the discriminator doesn't train well. Hence we investigate the result of training the model after doubling the training rate from 0.0002 to **0.0004** for the discriminator to let it train faster. All the other parameters remain the same between this model and ResNet-9.

Evaluation

Each model was evaluated by taking the pixel-by-pixel **Mean Square Error (MSE)** between the generated images to the ground truth map images.

Implementation

We implemented the aforementioned model using **Tensorflow** in Python. The code for the all the experiments is available at https://github.com/sdnr1/c-gan_pix2pix.

Results

Visual Analysis

The generated images from different architectures are shown below. The baseline is the **C-GAN** model proposed by Isola et al. Our results show that both **U-Net** and **ResNet-9** generators can capture the general features of the aerial images. Our residual connection based generators outperform the U-Net generators in this aerial-to-map translation task. ResNet-9 is able to identity the highways and produces straighter street blocks in the map. ResNet generators are effective because residual connections make it easy for the network to learn the identity function, and allow easier training in deep networks. Both of these two properties are highly appealing in image translation tasks.

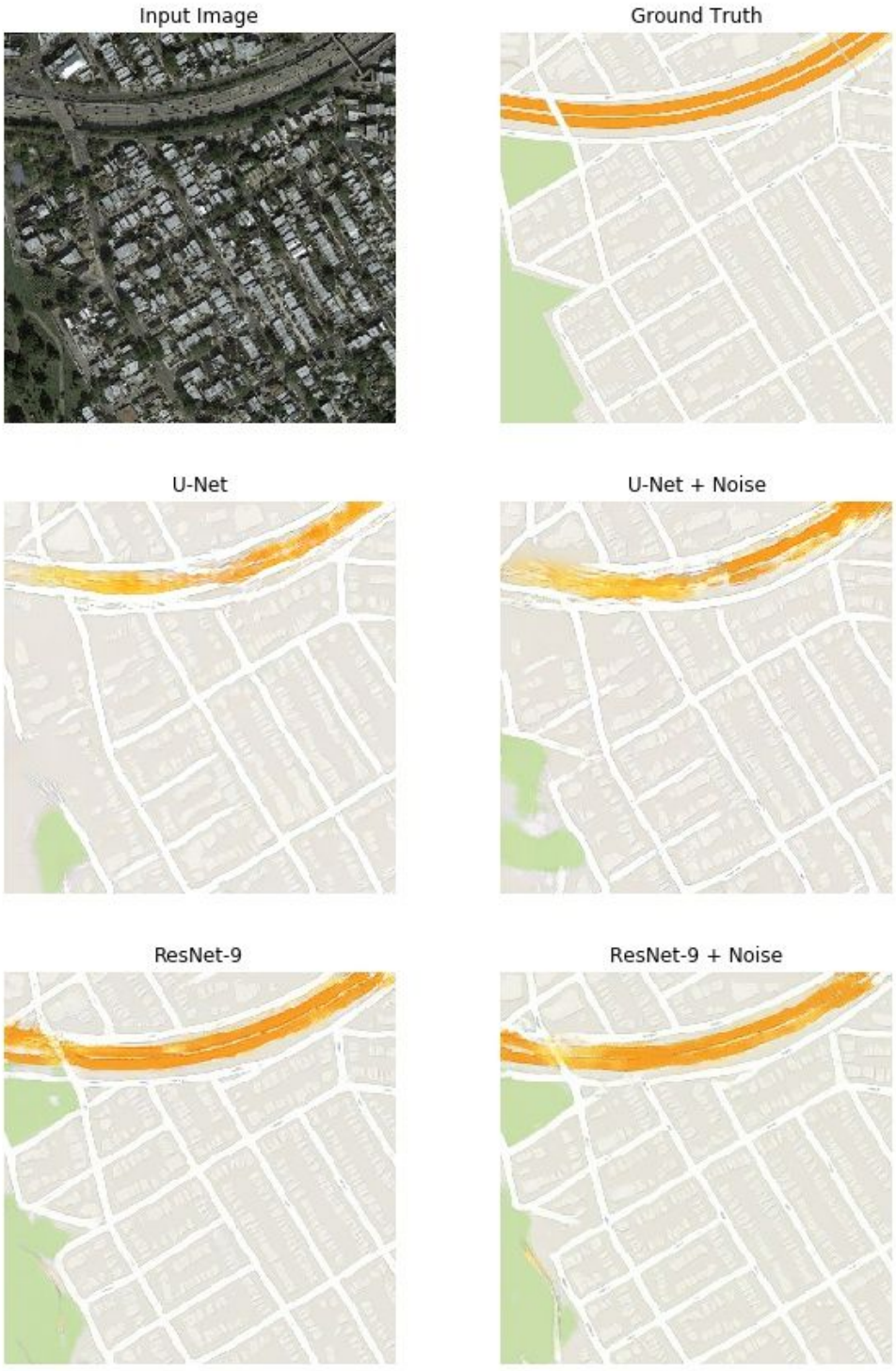
A few samples are presented in the following pages for a visual comparison.

Model Evaluation

The following table shows the **Mean Squared Error (MSE)** observed on the four different generator networks.

Model	MSE
U-Net	0.01834
U-Net + Generator Noise	0.01852
ResNet-9	**0.01443**
ResNet-9 + Generator Noise	0.01476

ResNet-9 model performs the best in our analysis. It shows an improvement of about **21.3%** over the U-Net model proposed by Isola et al.


Input Image
Ground Truth
U-Net
U-Net + Noise
ResNet-9
ResNet-9 + Noise

Sample 1


Input Image
Ground Truth
U-Net
U-Net + Noise
ResNet-9
ResNet-9 + Noise

Sample 2

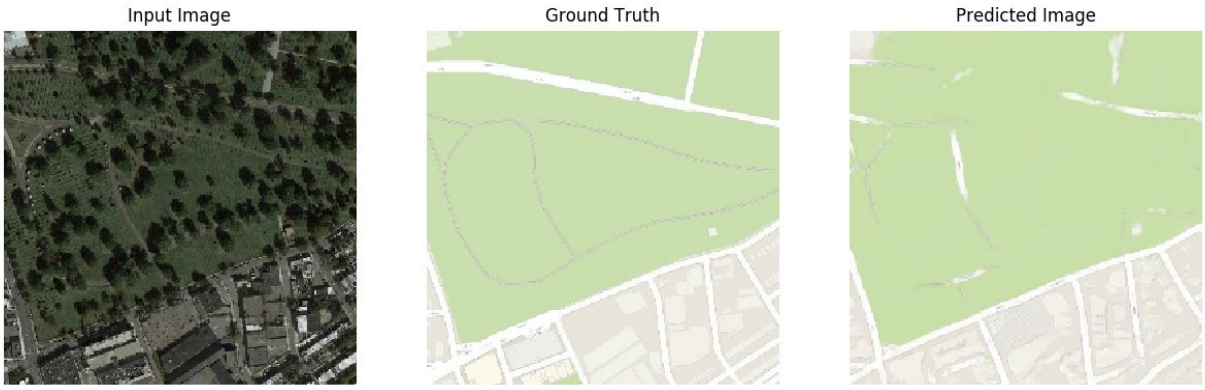

Input Image
Ground Truth
U-Net
U-Net + Noise
ResNet-9
ResNet-9 + Noise

Sample 3

Discussion

We compared the performance of the U-Net based generator GAN model proposed by Isola et al. to the Residual based ResNet-9 generator GAN for translating aerial images to map images. It was observed that **ResNet-9 model performed significantly better** in the given task, which is reflected by both the lower mean squared error of the model and better quality results in visual analysis.

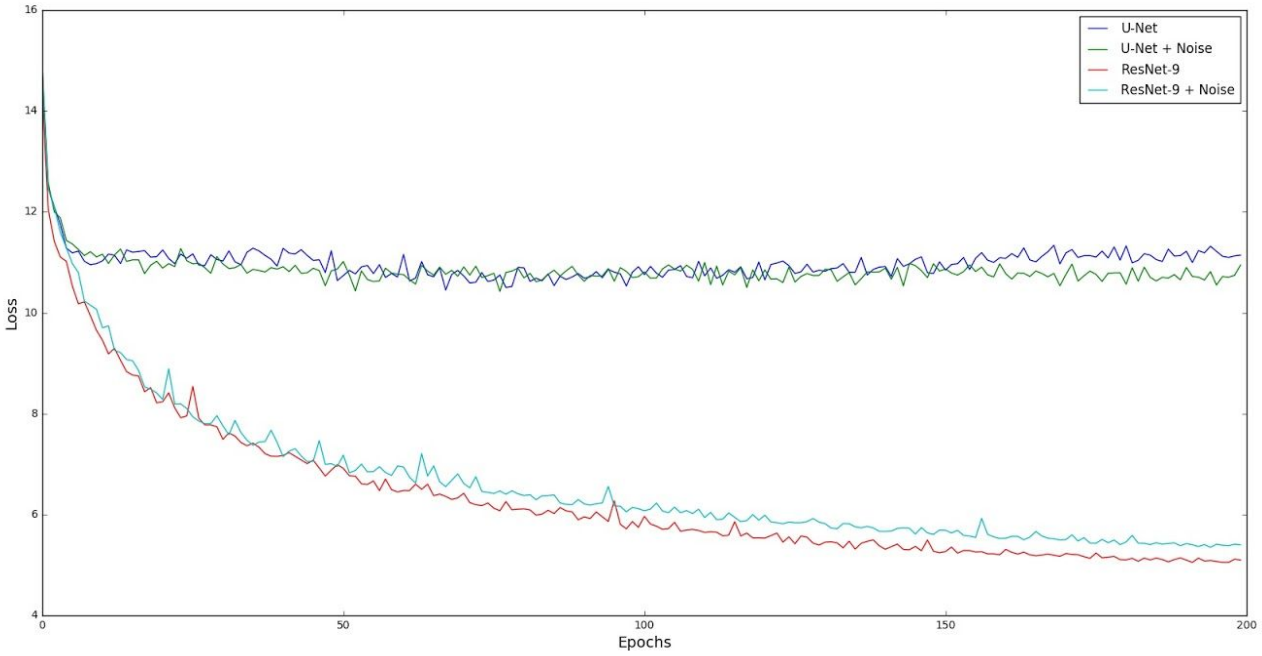
Although, all models struggle more with capturing scenes with large area of green lawns and curvy walkways. We think the unsatisfying performance with these more complex scenes is due to the data distribution of our training set - there are more training examples with grid-like streets than with water and parks with irregular pedestrian walkways. An example of such a case is shown below.



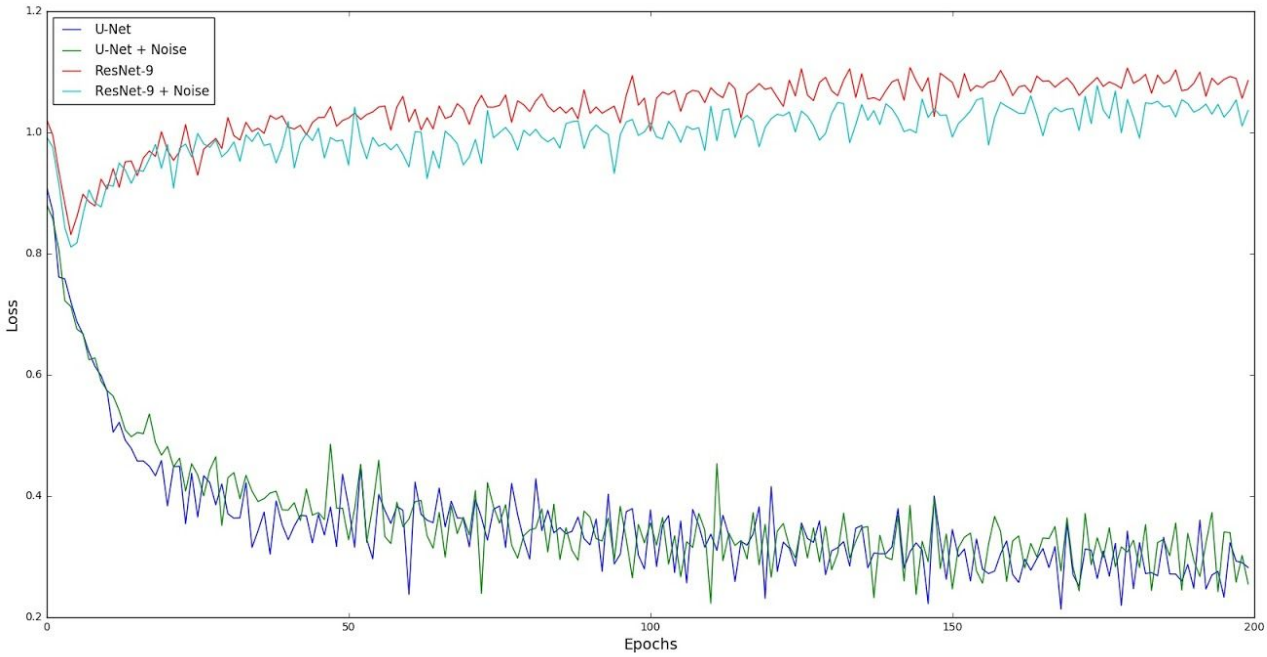
Sample showing shortcoming

Despite of that, we can still very much conclude that **ResNet-9 performs better than U-Net** by quite a margin. Also we can see that addition of noise hardly brings any benefit. Rather, models with noise inputs perform slightly worse as compared to their corresponding versions without noise inputs.

Further analysis of the loss function during training reveals why this is the case. The following shows the plot of the loss functions during training the model from the beginning to 200 epochs.



Generator Losses

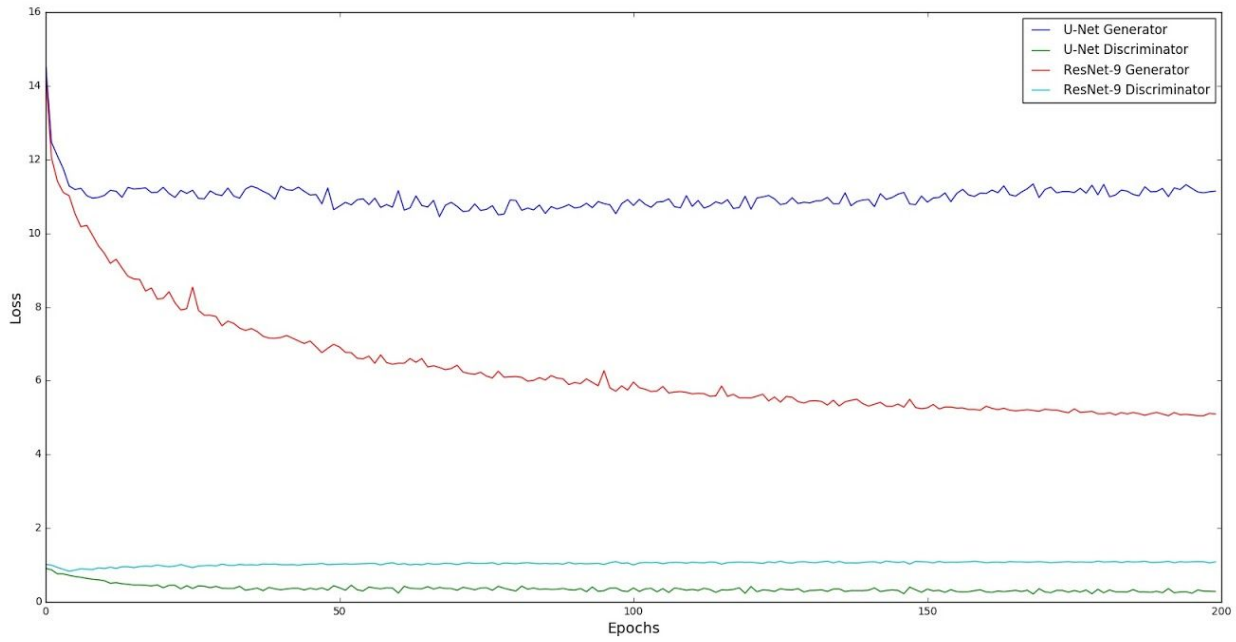


Discriminator Losses

Inferences

★ **No major differences** can be seen between noisy and their non-noisy versions. This means that the neural network is not incorporating noise into the generation of the output and it is learning to **ignore noise** altogether. Moreover, noise only seems to hinder the training process which can be possible explanation for slightly worse performance of the models with noise inputs.

★ In the learning curve of the **U-Net**, the generator loss is not minimized very well. The generator seems to be **overpowered** by the discriminator within a few epochs and doesn't train after that. On the other hand, the generator is **not overpowered** by the discriminator in the **ResNet** model and the model is trained much better in this case. In other words, the U-Net generator is not able to learn to generate the desired output with good accuracy. This is the reason why ResNet-9 performs much better than U-Net. To show a clearer comparison, the figure below shows learning curves for the generators and discriminators of the both U-Net and ResNet-9 model (non-noise inputs model only).



U-Net vs ResNet comparison (non-noisy models)

★ It can also be argued that in the after upgrading the generator from U-Net to ResNet-9, the discriminator might as well require an **upgrade** to keep up with the more powerful generator. It can be clearly seen in the learning curve of ResNet-9 discriminator that the it does not train very well and the loss value is as good as randomly guessing the output. Although, this does not mean that discriminator is not contributing at all. The discriminator is simply overpowered in this setting.

Additional Experiment

In order to see if we can further push the performance of the model by implementing an alternate training strategy we conducted another experiment.

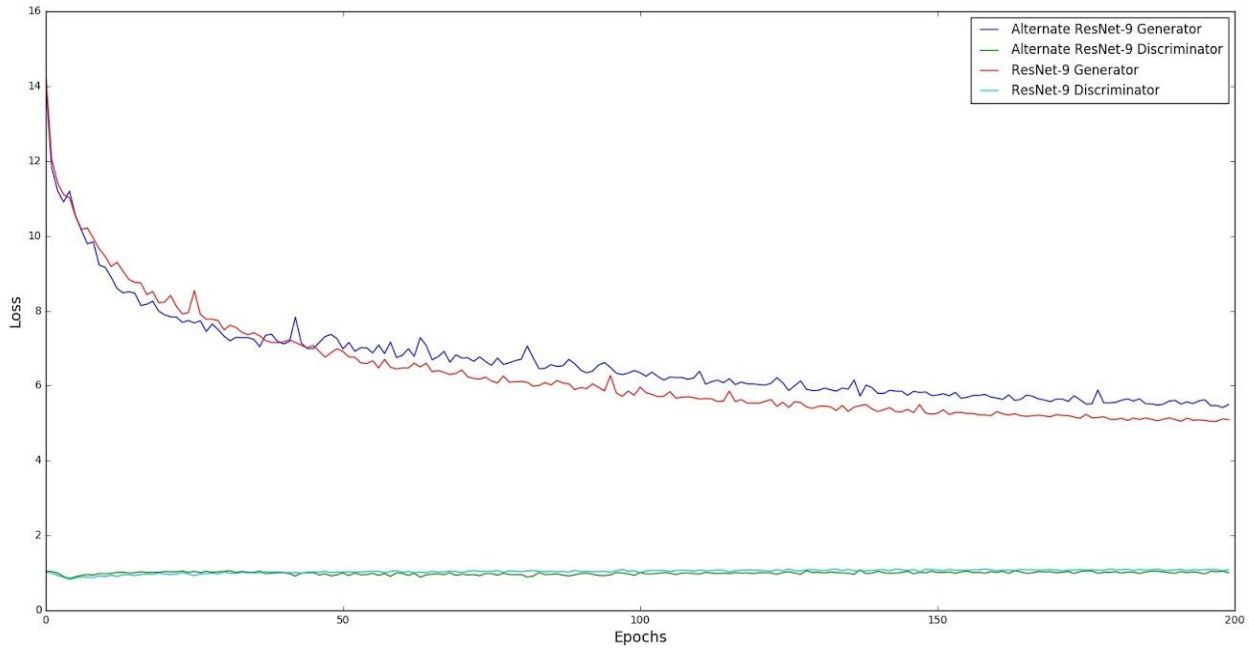
The reason for doing so is that the discriminator in the ResNet-9 model is quickly overpowered by the generator as mentioned above. We believe so because the discriminator loss goes up as the model is trained (even though it does decrease initially for a few epochs). Hence, we **increase the learning rate** at which the discriminator trains and investigate the results of this alternative training strategy.

Evaluation

Model	MSE
ResNet-9	0.01443
Alternate ResNet-9	0.01432

Discussion

Hardly any difference was observed in the Mean Squared Error of both the models. As show below, there is no significant difference between the training curves of these models. Any differences between the 2 models are **within stochastic tolerance** and can be attributed to the probabilistic nature of the training process. For the sake of brevity, we do not show samples of images generated by the Alternate ResNet-9 model since they were very much similar to those generated by the original ResNet-9 model.



Alternate ResNet-9

Inference

It can be inferred from the above result that there might be a need for a **better discriminator** network to boost the results of the ResNet-9 model. Further improvements of the discriminator network are beyond the scope of our study and also beyond our reach due of limitations on hardware resources to conduct further experiments since these may involve upgrading the discriminator network or experimenting with different training frequencies for generator and/or discriminator.

Conclusion

The results suggest that **conditional adversarial networks** are a promising approach for many image-to-image translation tasks, especially those involving highly structured graphical outputs. These networks learn a loss adapted to the task and data at hand, which makes them applicable in a wide variety of settings.

In our study, we first generated results for the model using **U-Net architecture** for the generator. We then used the **ResNet architecture** for the generator. It was observed that using the ResNet based generator model improved the performance and reduced the mean squared error of U-Net based generator model from **0.01834** to **0.01443**, which is a significant improvement. Hence, residual-based networks outperform U-Net model and baseline in aerial-to-map translation. Residual connection not only makes deeper networks easier to train, but also allows learning to be more end-to-end in the sense that the model chooses where to keep or discard information from the previous layer at any point, while U-Net is forced to only pass information from the first layer to the last and so on.

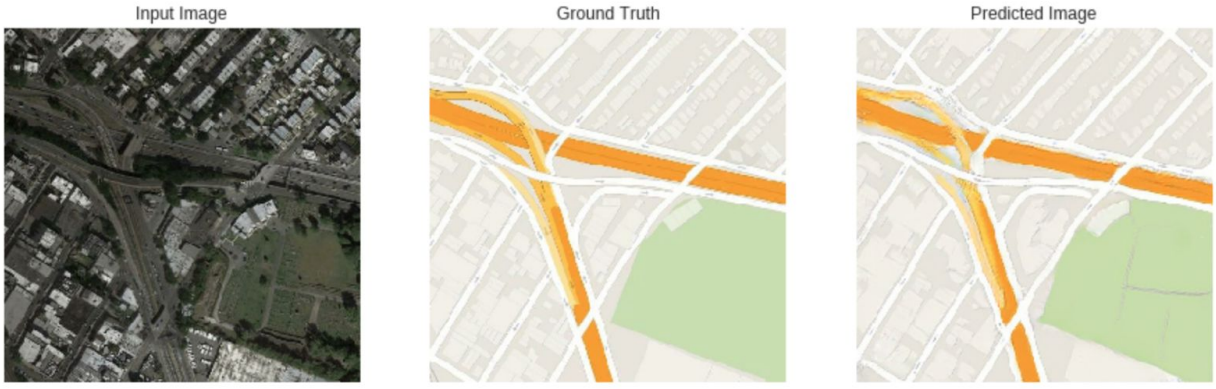


Image generated using ResNet-9 generator

We also added noise to the input of generator to avoid the model from **overfitting** on the given training dataset. This was intended to help the generalize better to unseen data and to help capture the full entropy of the conditional distributions. We generated random noise and added it to the input to the generator so that our model could learn to translate an image it had never seen before. However, we observed that the model learns to ignore the noise altogether. Further, noise might even be hindering the training process as

indicated by the learning curves and the slightly higher mean squared error of the noisy input based models.

It was also observed that all **models struggle** more with capturing scenes with large area of green lawns and curvy walkways. The unsatisfying performance with these more complex scenes might be due to the data distribution of our training set - there are more training examples with grid-like streets than with water and parks with irregular pedestrian walkways. Also, some of the results, like the shown below, bring out an inherent **shortcoming** of automatic translation of aerial-to-map images. It was seen that parts of roads heavily covered by foliage or other objects are hard to detect. All models generate unsatisfactory results in such a case, although ResNet-9 does improve over U-Net in these situations as well.

Shortcoming

One major observation was that the **generator and the discriminator didn't train at the same rate**. We took the best performing model in our study, ResNet-9, and experimented with changing the training strategy to offset the differences in training. Since, the generator was performing better than the discriminator, we increased the learning rate for the model's discriminator network from 0.0002 to 0.0004. But, no change was observed in performance or even the training curves for the alternative training strategy. Thus, more aggressive changes like **changing training frequencies** or an **upgrade to discriminator network** may be required to improve the model. These changes are out of the scope of our study.

Future Scope

C-GAN is an effective solution for translating image from one visual domain to another. The applications of C-GAN are not limited to aerial-to-map translation. C-GAN can be applied to translations between other domains of images such as black/white-to-color, day-to-night, edges-to-photo, etc. C-GAN also has potential in image segmentation tasks.

Future work include exploring **residual-based network for discriminator**, and experimenting with different training frequencies generator and/or discriminator. Also, dynamic training frequencies can also be used to allow generator and/or discriminator to train more often in the beginning and gradually slow down.

Furthermore, state-of-the-art image based networks like the deeper **ResNet-50** and **DenseNet** can be used for the generator network to produce even better results. We could not incorporate these models in our study due to the lack of computing resources.

Finally, the dataset used in our study in not sufficient to capture the full entropy of aerial-to-map image translation. As mentioned before, there are more training examples with grid-like streets than with water and parks with irregular pedestrian walkways in the dataset. An **augmentation** to the dataset is required to offset this issue and ensure better generalization of the network to unseen data. More training data is also necessary for training larger models like ResNet-50 and DenseNet.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[3] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340. ACM, 2001.

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

[5] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

[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. arXiv preprint, 2016.

[7] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision*, pages 702–716. Springer, 2016.

[8] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, and Zhen Wang. Multi-class generative adversarial networks with the L2 loss function. *CoRR*, abs/1611.04076, 2016.

[9] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[10] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[12] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv: 1511.06434, 2015.

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