Image-to-Image Translation with Conditional-GAN

Jason Hu
Department of ERE
Stanford University
jhu7@stanford.edu

Weini Yu
Department of CS
Stanford University
weiniyu@stanford.edu

Zhouchangwan Yu
Department of EE
Stanford University
zyu21@stanford.edu

Abstract

We investigate image-to-image translation using conditional-generative adversarial networks (C-GAN) on aerial-to-map images. We start by reproducing the C-GAN model proposed by Isola et al., and then explore various network architectures, loss functions, and training strategies. We present qualitative and quantitative evaluations of different models and conclude that the residual-based model generates superior quality images with only 1000 training examples.

1 Introduction

Image-to-image translation are tasks that take in input images and generate or manipulate them into a different visual space. Traditionally, different tasks such as converting grayscale to color, image to semantic labels, and edge-map to photograph require different hand-crafted machinery. Recent development of GANs [1] offers a more general approach for such similar tasks through learning an "adversarial" loss that adapts to data. In this project, we explore image-to-image translation using C-GANs [4], in which we take an input image and generate the desired output image using GAN conditioned on the input image.

2 Related Work

The predecessor to image-to-image translation was "image analogies" proposed by Hertzmann et al. [3], in which a nonparametric multiscale regression model was used to learn a filter to apply to the input image. Traditional methods for image generation are neither flexible nor efficient. Fortunately the recent development in generative adversarial networks shed new light on image generation tasks. GAN in unsupervised setting has achieved remarkable results in image inpainting [10], style transfer [7], single image super-resolution [6]. To make the GAN framework more flexible for a wide range of image translation tasks, Isola et al. [4] proposed to use conditional adversarial network to learn a structured loss so that the network adapts to the tasks and data. Our method differs from the previous works in architecture choices, and utilization of important ideas developed in perceptual loss [5], residual connections [2], and deep convolutional adversarial generative networks [12].

3 Dataset

We use the aerial-to-map dataset from the pix2pix datasets[4]. 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, 1098 validation image pairs, and 1098 test image pairs. Each image is composed of RGB channels and of size 600×600. Dataset is obtained from the pix2pix project site.

3.1 Data Preprocessing

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

CS230: Deep Learning, Spring 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

4 Methods

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. Figure. 1 illustrates the conditional-GAN architecture.

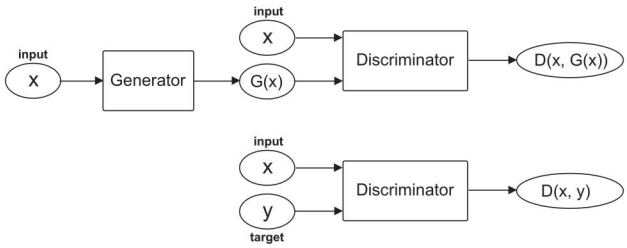


Figure 1: Network architecture of conditional GAN model.

4.1 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:

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

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 [4]. Thus our full objective for the minimax game is:

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

4.2 Network Architecture

4.2.1 Generator

We develop our training framework in PyTorch [9]. Both variations of our generator processes an input image x of size $3 \times 256 \times 256$ and generates an output image $G(x)$ of the same size.

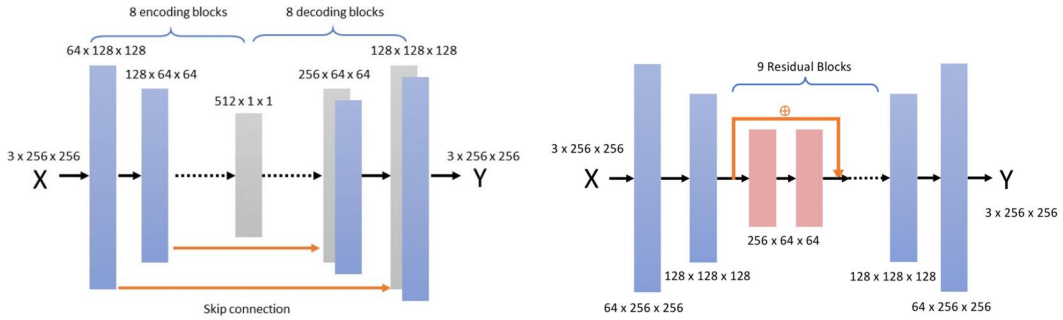
U-Net The U-Net generator is an encoder-decoder network with symmetrical long skip connections [13]. The network consists of 8 encoding layers and 8 decoding layers, with skip connections from layer i to layer $n - i$, where n is the total number of layers. Each encoding and decoding block follows the form of convolution/deconvolution-BatchNorm-LeakyReLU. Figure. 2(a) illustrates the U-Net architecture.

ResNet To improve model performance, we develop a residual-based network based on the ResNet model in Johnson et al. [5]. Our network is composed of 2 encoding blocks, 6 or 9 residual blocks (ResNet-6 or ResNet-9), and 2 decoding blocks. Each encoding or decoding block follows the two-stride convolution/deconvolution-InstanceNorm-ReLU structure, and each residual block follows the convolution-InstanceNorm-ReLU-convolution-InstanceNorm residual connection structure. Figure. 2(b) illustrates our ResNet architecture.

ResNet-50 To further increase the capacity of our generator, we use the ResNet-50 network by He et al. [2]. We discard the last fully connected layer, and add five deconvolutional blocks to upsample the encoded features in order to produce output image of size $3 \times 256 \times 256$.

4.2.2 Discriminator

PatchGAN We use a convolutional "PatchGAN" classifier with architecture similar to the classifier in pix2pix [4] as our discriminator. PatchGAN discriminator determines whether an image is real or fake by using local patches of size 70×70, 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 downsampling convolutional-BatchNorm-LeakyReLU layers, and outputs a matrix of 30×30, in which each element corresponds to the classification of one patch.



(a) U-Net (b) ResNet-9

Figure 2: Generator Network Architecture

ImageGAN ImageGAN uses 7 downsampling convolutional-BatchNorm-LeakyReLU layers, and classifies whether an image is real or fake with receptive field of the entire image.

4.3 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

$$\mathcal{L}_{\text{gen}}(G, D) = BCE(D(x, G(x)), 1) + \lambda \mathcal{L}_{\text{L1}}(G) \tag{3}$$

$$\mathcal{L}_{\text{dis}}(G, D) = BCE(D(x, G(x)), 0) + BCE(D(x, y), 1) \tag{4}$$

We also explore the option to replace BCE with least squares loss for the adversarial loss because least squares loss is proven to generate higher quality images, and stabilizes the training process [8].

The discriminator trains faster than generator because classification is a relatively simpler task compare to generation. As a result we need to slow down the rate at which the discriminator learns relative to the rate generator learns. In our study we slow down $\mathcal{L}_{\text{dis}}(G, D)$ by a factor of 2 as suggested by Isola et al.. We use two separate Adam solvers for the generator and discriminator with $\beta_1 = 0.5, \beta_2 = 0.999$, learning rate $\alpha = 0.0002$, $\lambda = 100$. 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 hyperarameters because they have proven to work well with the C-GAN model developed by Isola et al [4]. 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 best performance. We also explored different training frequencies for the discriminator which we will discuss further in section 5.3.

5 Results and Discussion

5.1 Visual Analysis

The generated images from different architectures are shown in Figure. 3. The baseline is the pre-trained C-GAN model from Isola et al. [4]. Our results show that both U-Net and ResNet generators can capture the general features of the aerial images. Our U-Net model with finely tuned parameters produces better results than the baseline, and generates maps with clearer definition of roads.

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 [2], and allow easier training in deep networks. Both of these two properties are highly appealing in image translation tasks. In our study ResNet-50 does not produce good quality maps. We speculate the reasons are: 1) More hyperparameter tuning is needed since ResNet-50 is a much deeper network 2) Our dataset might be too small for training such a deep network.

3) ResNet-50 largely increases the capacity of the generator but discriminator is kept the same, so there might be an performance imbalance between the two.

In the study we also investigate the impact of PatchGAN and ImageGAN discriminators on the performance of our C-GAN model. PatchGAN processes information on 70×70 local patches instead of an entire image, has fewer parameters, and is faster to train. The generated images from PatchGAN as the discriminator have better quality than ImageGAN (column 3 vs. column 4). Thus, PatchGAN is our default discriminator.

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.

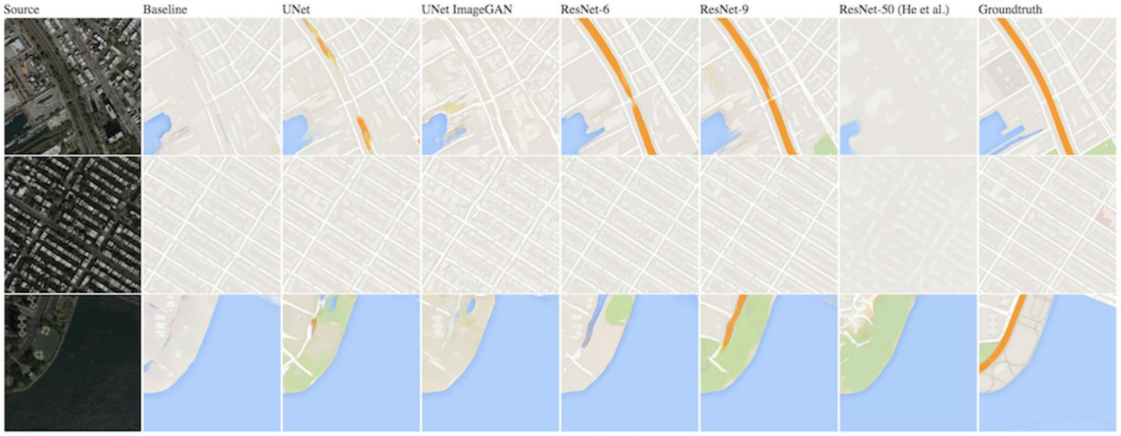


Figure 3: Generated map images of different architecture and hyperparameters. From left to right are source aerial images, baseline, U-Net, U-Net with ImageGAN, ResNet-6, ResNet-9, ResNet-50, and ground truth map images

5.2 Model Evaluation

Mean Squared Error The pixel by pixel Mean Square Error (MSE) between the generated images to the ground truth map images is calculated for baseline, U-Net, ResNet-6, ResNet-9, and ResNet-50 generators, as shown in Table. 1 . ResNet-9 has the lowest MSE, which means it is the closest to the ground truth.

t-SNE T-Distributed Stochastic Neighbor Embedding (t-SNE) dimension reduction technique is employed to visualize the distribution of generated images. We randomly choose five generated maps from each generator and project them into 3-dimensional space [11], shown in Figure. 4 . Since the image-to-image variation is greater than the model-to-model variation, the generated maps of each input image are clustered together. In most cases, the results from ResNet-9 are closer to the ground truth, which is consistent with our observations and MSE calculations.

Model	MSE
Baseline	1.5661×10^{-2}
U-Net	1.5058×10^{-2}
ResNet-6	1.3298×10^{-2}
ResNet-9	$\mathbf{1.1969 \times 10^{-2}}$
ResNet-50	1.5848×10^{-2}

Table 1: Mean Squared Error of generated images

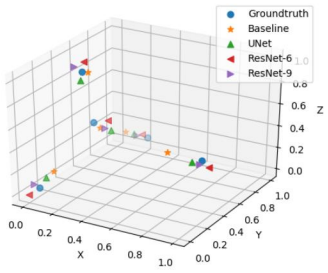


Figure 4: t-SNE plot of 5 random generated images from different models and ground truth

5.3 Discriminator Score

To have a deeper understanding of how our model performs, we visualize the discriminator scores of ResNet-9 generated images and ground truth images from training set as shown in Figure. 5 (left two). We see that the discriminator does a good job in classifying the majority of the ground truth images as real, and most of the generated images as fake. We suspect the score distribution shown in Figure. 5 (left two) is an indication of the discriminator performing better than the generator, so we experiment with letting the generator train more often by updating the discriminator every 3 steps the generator updates. As a result almost all images are classified as real by the discriminator (Figure. 5 right two). Interestingly, neither visual nor quantitative evaluation show obvious difference from this change. This indicates L1 loss might be more dominant than the adversarial loss in our overall loss objective.

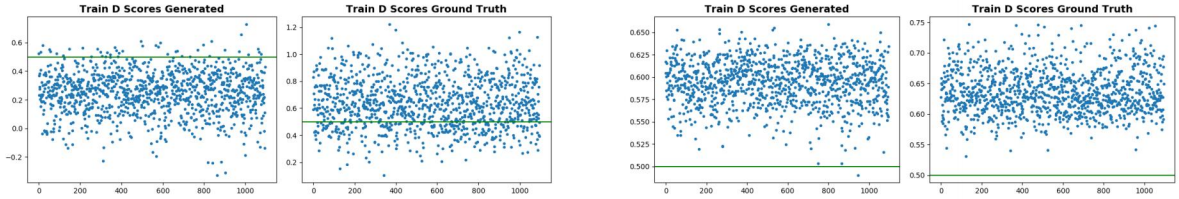


Figure 5: Discriminator scores of ResNet-9 generated and ground truth images from training set. Left two plots show updating D every step; right two plots show updating D every 3 steps.

5.4 Learning the Default from Unpaired Data

When we ran the model on GPU for the first time, the input images and target images read by our DataLoader were not paired. After training for 100 epochs, we found that the generated images were all the same!

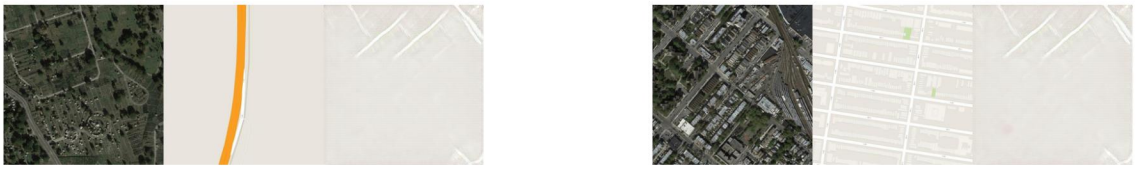


Figure 6: Input, target, and generated images when the input and target images are not matched.

The model did not learn anything meaningful about aerial-to-map translation but to generate a "default" image that has the minimum loss with all the target images in the training set with any input image. After sorting the images in our datasets, our model was able to process paired input and target images and generate reasonable results after training, as shown in section 5.1.

6 Conclusion and Future Work

C-GAN is an effective solution for translating image from one visual domain to another. Our 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.

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, and has potential in image segmentation tasks.

Future work include exploring residual-based network for discriminator, and experimenting with dynamic training frequency to allow generator train more often than discriminator in the beginning and gradually slow down. We would also like to explore adding more random noise to the generator to help capture the full entropy of the conditional distributions.

7 Contributions

All three authors contributed equally to the project. Weini worked on the training infrastructure, integration of components and visualization. Zhouchangwan worked on the architecture of generator and discriminator. Jason worked on implementing ResNet and data processing. All three authors contributed to model architecture development and tunning, result analysis and report writing.

Our code is on GitHub at `https://github.com/yuzhoucw/230pix2pix`.

Acknowledgements

We would like to thank Abhijeet Shenoi for his helpful feedback during office hours. We would also like extend our gratitude to the Center for Energy-Efficient Computing and Applications of Peking University for their computer cluster and NVIDIA V100 GPUs.

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.