Wasserstein GANs for Image-to-Image Translation

Noah Makow
Stanford University
nmakow@stanford.edu

Abstract

A large number of tasks in computer vision can be represented as the translation from an input to an output image. The application of recent deep learning approaches, in particular conditional generative adversarial networks (cGANs), to the image translation problem has shown promising results. In this paper we seek to combine a proven framework for image translation with a recently proposed GAN, the Wasserstein GAN (WGAN), that offers a number of benefits compared to the vanilla GAN. Ultimately, we find that while WGAN offers some benefits, it does not produce a significant difference in the quality of generated samples and thus may not be worth the increased training time. We also present a new metric for measuring the quality of generated samples, VGG cosine similarity.

1. Introduction

A large number of tasks in computer vision can be represented as the translation from an input to an output image. For example, a given scene might be represented as an RGB image, a field of gradients, an edge or blob map, or a semantic label map. Recent research has defined *image-to-image translation* as the task of translating from one scene representation to another given sufficient training data [3]. Historically these mappings have been obtained through specialized techniques for each specific problem, such as SIFT or HoG to obtain gradient fields or the Laplacian of Gaussian method for edge detection. Recent research has explored the possibility for a single, general-purpose framework capable of translating images between arbitrary domains.

The problem remains an open area of research, and for a number of the problems mentioned above, traditional computer vision techniques remain superior due to more rigid assumptions and the rich understanding of the rules governing transformations between image representations. In recent years, as data and compute has become increasingly available, deep learning solutions are just now approaching competitive performance on these tasks. Despite vast amounts of data generated, labelled data is fairly difficult to find but remains central to the performance of deep learning approaches.

The application of recent deep learning approaches, in particular conditional generative adversarial networks (cGANs), to the image translation problem has shown promising results. At the same time, significant work has been invested in exploring alternative GAN formulations that help improve to stability, reduce mode collapse, and otherwise improve GAN training. In this paper we seek to combine a proven framework for image translation with a recently proposed GAN, the Wasserstein GAN, that offers a number of benefits compared to the vanilla GAN. We explore whether or not the proposed benefits of WGAN hold for the task of image translation, comparing WGAN with the standard Pix2Pix framework.

2. Background & Related Work

2.1. Pix2Pix [3]

Isola et al. introduced a unified framework known as Pix2Pix to address the image-to-image translation problem in its most general form. Given a image pairs (A, B), the Pix2Pix framework can be trained to "translate" images A into their corresponding form B, or vice versa. Pix2Pix uses a U-Net architecture for the generator and a deep convolutional discriminator. It is trained using conditional GAN loss with the addition of an L1-distance term to ensure that the generated B is close to the true B. As Pix2Pix serves as the foundation of our architecture, we explore the technical details of the architecture in greater detail in the Approach section below.

2.2. Wasserstein GAN [1]

Arjovsky et al. explore an alternative to traditional GANs that takes a different perspective on GAN learning objective. Whereas the normal GAN loss function will attempt to minimize the KL-divergence between generated distribution and the true data distribution, the WGAN loss formulation instead minimizes the Wasserstein distance between the two distributions. Arjovsky et al. claim that this approach has several benefits:

- an interpretable loss metric that correlates with generator's convergence and sample quality
- increased stability of the optimization process
- reduced mode collapse

The first point is especially helpful for allowing effective hyperparameter tuning, which has been historically difficult with GANs.

3. Approach

3.1. Conditional Generative Adversarial Networks

GANs are generative models that learn a mapping from a random noise vector z to an output image y: $G : z \rightarrow y$ [2]. In contrast, cGANs learn a mapping from an observed image x and a random noise vector z to an output image y: $G : \{x, z\} \rightarrow y$ [4, 3]. The objective of the generator G is to produce outputs that are indistinguishable from "real" images to an adversarial discriminator D. G and D are trained simultaneously until G is able to successfully fool the D – that is, D is unable to distinguish real images from generated images with probability greater than 0.5.

The cGAN objective can be expressed as:

$$L_{cGAN}(G, D) = \mathbb{E}_{x,y\sim p_{data}(x,y)}[\log D(x, y)]+ \\ \mathbb{E}_{x\sim p_{data}(x),z\sim p_z(z)}[\log(1 - D(x, G(x, z))].$$

Note that we want D to output high probability for real images $\log D(x, y)$ and low probability for fake images $\log(1 - D(x, G(x, z))$. We compare the cGAN objective to the vanilla GAN objective, in which G and D do not directly observe x:

$$L_{GAN}(G, D) = \mathbb{E}_{y\sim p_{data}(y))}[\log D(y)]+ \\ \mathbb{E}_{x\sim p_{data}(x),z\sim p_z(z)}[\log(1 - D(G(x, z))].$$

In addition to the L_{cGAN}, prior research [5] that has found benefit to include more traditional loss functions such as L1 or L2 distance between the generated image and the ground truth output image. In this scheme we only alter G in the sense that it now must not only fool the discriminator, but also produce outputs that are close to the ground truth:

$$L_{L1}(G) = \mathbb{E}_{x,y\sim p_{data}(x,y),z\sim p_z(z)}[||y - G(x, z)||_1], \\ L_{total}(G, D) = L_{cGAN}(G, D) + \lambda L_{L1}(G).$$

This method is commonly referred to as *Pix2Pix*, although in the Pix2Pix framework the random noise vector z is not necessarily included as input to the generator as in general conditional GANs.

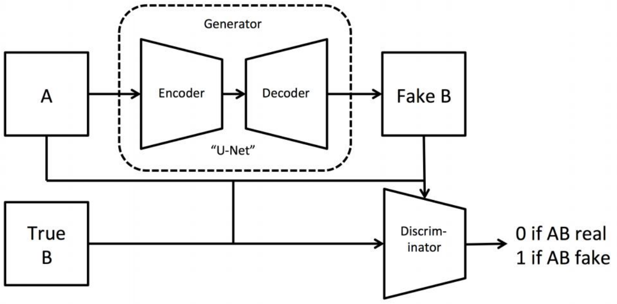

Figure 1. A visualization of the full image translation architecture. Image A is fed into the generator network, a "U-net" architecture to produce the Fake B. During training, the pairs (A, True B) and (A, Fake B) are passed to the discriminator network, a deep CNN. For WGAN, the discriminator outputs a real-valued score rather than a probability.

3.2. Wasserstein GAN

The key difference in WGANs is a differing loss function that shifts the objective from minimizing KL-divergence to Wasserstein distance between the generated and true distributions. In WGANs, the discriminator is often referred to as a "critic" – instead of outputting a probability of the sample being real or fake, it instead outputs an unbounded score of how real the sample is (the name is inspired by the concept of actor critics from reinforcement learning). Omitting the L1 loss, the WGAN loss can be expressed as:

$$L_{WGAN}(G) = -\mathbb{E}_{x\sim p_g(x)}[D(x)] \\ L_{WGAN}(D) = \mathbb{E}_{x\sim p_g(x)}[D(x)] - \mathbb{E}_{x\sim p_r(x)}[D(x)]$$

This has the effect of minimizing the Wasserstein distance between probability distributions p_g and p_r.

3.3. Network Architecture

See Figure 1 for a visual overview of the image translation architecture.

3.3.1 Generator: U-Net [3]

Our generator network consists of a U-Net encoder-decoder architecture. The U-Net is a standard deep convolutional encoder-decoder network with the addition of "skip" connections. Both the encoder and decoder components have $L = 8$ layers, and the output of layer i in the encoder is concatenated (or "connected") with the output of layer $L - i$ in the decoder. Encoder layers contain LReLU→CONV→BN, while decoder layers are DECONV→BN(→DROPOUT). The output is an image of the same spatial dimensions as the input image but with the number of channels corresponding to the output representation.

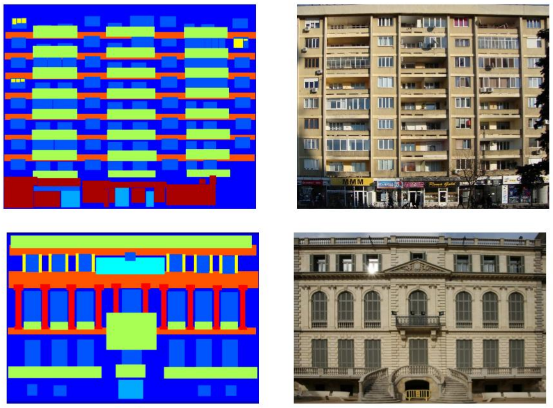

Figure 2. Example image pairs from the CMP Facades dataset. On the left we have images of building facade, and on the right we have the corresponding hand-labelled segmentation of different components of the facade. There are 12 classes specified in the original report [5]: *facade*, *molding*, *cornice*, *pillar*, *window*, *door*, *sill*, *blind*, *balcony*, *shop*, *deco*, and *background*.

3.3.2 Discriminator: Deep CNN

We utilize a five-layer deep CNN as the discriminator. As input to our discriminator, we concatenate depth-wise the input image x with either the real output image (for real examples) or the generated image y (for generated examples). The discriminator is a fairly standard CNN with CONV→BN→LReLU layers and the output being a single sigmoid neuron predicting if the input image pair is real or fake.

4. Experiments

4.1. Dataset

For the image-to-image translation problem, the general structure of the training data are input and output pairs of images. Each corresponding input and output image represents the same scene in the desired representation (e.g. an input image might represent an edge map of a particular image and the output image the full RGB pixel tensor).

We make use of the CMP Facades dataset, which include building facade images assembled at the Center for Machine Perception, including 606 rectified images of facades from various sources that have been manually annotated. The facades are from different a number of different cities and include varying architectural styles. We select 400 images for the training set and hold out 100 images for each of the validation and test sets. See Figure 2 for example image pairs from the dataset.

4.2. Model Details

We utilize Pix2Pix as a baseline and implement the WGAN loss and training algorithm atop it.

4.2.1 Pix2Pix

We train the Pix2Pix framework described above for 100 epochs using an Adam optimizer with a high learning rate (0.001). In contrast to the original Pix2Pix framework, we make two generator updates for each discriminator update. For all other hyperparameters such as the L1 loss weight, we use the values mentioned in the original Pix2Pix paper. The wall-time for training was roughly 7 hours.

4.2.2 WGAN-Pix2Pix

For our WGAN-Pix2Pix model we swap out the loss function of the generator and discriminator networks. We follow in the footsteps of the original WGAN paper for selection of hyperparameters. We update the critic network five times for each generator update (or 25 times every 500 updates). We use weight clipping to enforce the Lipschitz constraint rather than use gradient penalty. We train WGAN-Pix2Pix for 100 epochs using an RMSProp optimizer with a small learning rate (0.00005). Because of the increased number of updates per iteration, the wall-time for training was roughly 18 hours.

5. Results

5.1. Quantitative Results Summary

Model	L2 Distance	VGG Cos-Sim
Random	40990.1	0.15610
Pix2Pix	**25702.1**	**0.46944**
WGAN-Pix2Pix	25889.5	0.43776

Figure 3. A summary of the average L2 distance and VGG cos-sim on the test set. We see that both models significantly outperform a naive random baseline, and Pix2Pix appears to have the slight edge in both evaluation metrics over WGAN-Pix2Pix.

Deriving a quantitative measure for the quality of generated samples is a challenging open problem. Here we report two measures, *L2 distance* and *VGG cosine similarity*. L2 distance is defined to be the euclidean distance between the two images true B and fake B, averaged over the samples in the dataset. VGG cosine similarity is a measure that we also report, under the intuition that similar images should produce similar feature vectors in deep CNNs. We pass both the true B and fake B through a VGG16 network pretrained on ImageNet, and extract the final feature vector before the output layer. We report the average cosine similarity between the corresponding VGG feature vectors of the true B and fake B. See Figure 3 for a comparison of the quantitative results from Pix2Pix and WGAN-Pix2Pix.

We compare both models to a naive random baseline that generates random images and random feature vectors. As expected, both models significantly outperform the random baseline under both metrics. Pix2Pix has the slight edge in both metrics over WGAN-Pix2Pix after both models are trained for 100 epochs.

5.2. Inspecting Generated Samples

It is worthwhile to compare the samples generated by both models. In Figure 4 we present an array of generated samples along with the true output.

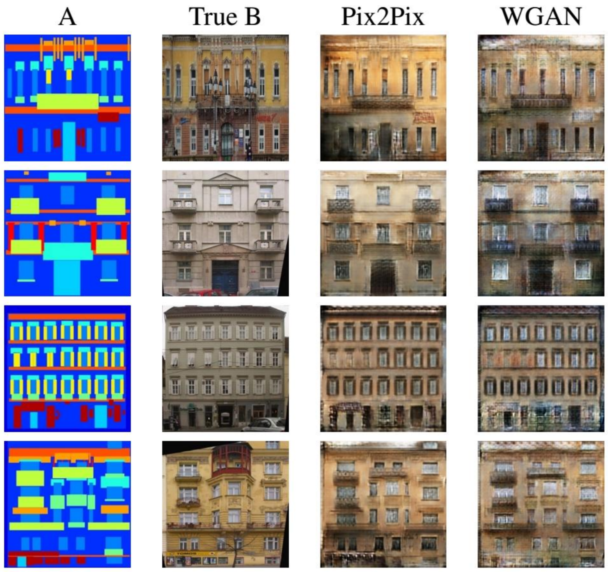

Figure 4. Samples generated from test set after both models were trained for 100 epochs. The first column is the input to the model, and the second column is the ground truth output.

Despite Pix2Pix having the slightly quantitative edge, both models generate similar samples in terms of appearance and quality. Both models do a good job of generating fairly common components such as windows and balconies, but struggle more with rarer and more varied components like facades, moldings, or backgrounds. Empirically there appears to be an averaging effect, where the regions generated appear to be nearly an average of those regions observed in the training set. This is opposed to other GANs such as BicycleGAN or cVAE-GAN that more nearly sample a particular output from the distribution rather than necessarily generating an "average" sample. As described further below, neither Pix2Pix nor WGAN-Pix2Pix is particularly effective at generating diverse samples and these alternative GANs can be more helpful in this respect.

5.3. Challenges

A number of challenges make it difficult to produce high-quality samples. Some of these are particular to the CMP facades dataset, but others are relevant to image-translation at large.

5.3.1 Sparse Images

Empirically, the quality of samples with fairly sparse semantic input maps was generally very poor. Figure 5 shows an example of the poor sample quality when this is the case. This appears to be because the generator does not have a direct sense of global consistency – that is, there is nothing to drive the generator to produce a wall with a consistent pattern, color, or texture. As a result we see seemingly random patches throughout the wall which do not appear to be high quality.

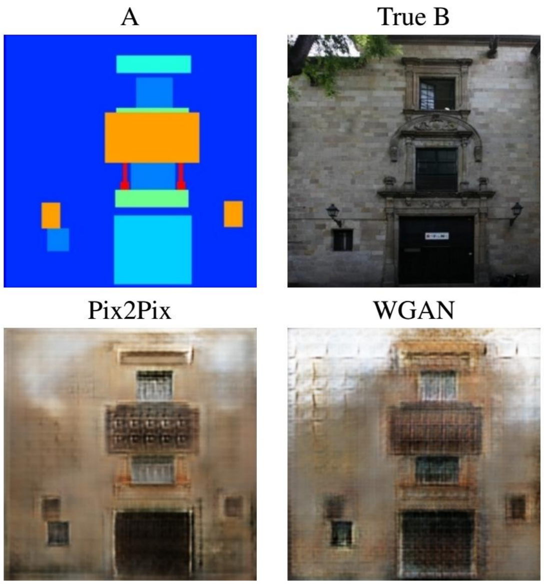

Figure 5. Both Pix2Pix and WGAN-Pix2Pix struggle to fill in large regions of background or wall with no notion of global consistency across components or the image.

5.3.2 Warping and Occlusions

Another issue stems from the noise found in the CMP facades dataset. In particular, a number of images are warped in order to bring the building facade parallel to the image plane. We show an example of this issue in Figure 6. Because the semantic map has no notion of this unused background space, our GAN must deal with this as noise and a misrepresentation of what backgrounds should look like. This results in poor quality samples as our model attempts to mimic this noise which is actually undesirable in the generated samples.

5.3.3 Sample Diversity

An issue shared by both Pix2Pix and WGAN-Pix2Pix is the lack of sample diversity. By sample diversity we refer to the diversity in appearance of generated samples for a particular input. In traditional conditional GAN settings, this

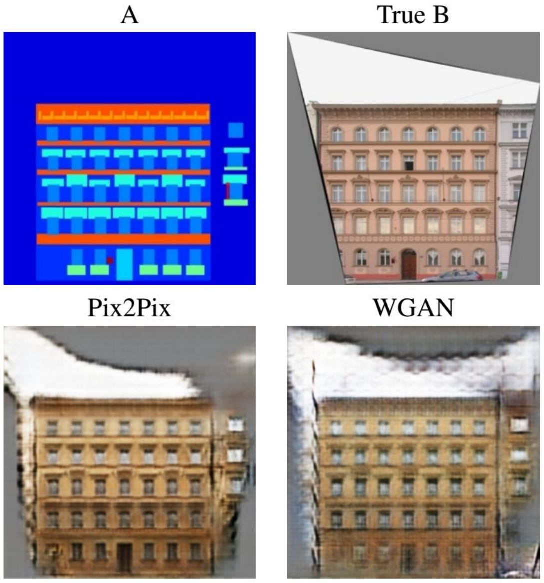

Figure 6. An illustration of how warped images in our dataset result lead to poor quality samples. The grey background region is noise that is undesirable in generated samples.

randomness is governed by the random noise input vector z. With Pix2Pix, the generated image is only a function of the input image. Other GAN architectures such as BicycleGAN and cVAE-GAN address this issue and have been shown empirically to produce a greater diversity of samples [6].

6. Conclusion

From the experiments carried out in this project, it is not clear that WGAN-Pix2Pix is a better option than vanilla Pix2Pix, despite the proposed benefits of the WGAN loss and optimization process. In fact, Pix2Pix appears to have a slight edge quantitatively and qualitatively after the same number of training epochs. Additionally, WGAN-Pix2Pix took roughly three times as long to complete the same number of training epochs and arrive at a comparable quality level as Pix2Pix. With these drawbacks in mind, adopting WGAN for image-to-image translation may not be a promising avenue. We should note that this should not be considered conclusive evidence as the significant training times and limited access to computing resources limited our ability to perform a complete hyperparameter search. However, a number of competing GAN architectures have been shown empirically to produce higher quality samples and with increased sample diversity.

Future work in the realm of image-to-image translation might involve experimenting with modifications to the loss function (such as the L1 loss term) to encourage higher quality output samples. In addition, further work is needed to determine meaningful quantitative evaluation metrics for image generation tasks in general. In this paper report one new metric, VGG cosine similarity, that is well-suited for this task. A more rigorous analysis regarding the effectiveness of this metric and its potential pitfalls is in order.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[3] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[4] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[5] R. Tyleček and R. Šára. Spatial pattern templates for recognition of objects with regular structure. In *German Conference on Pattern Recognition*, pages 364–374. Springer, 2013.

[6] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, pages 465–476, 2017.