Explore Image Translation in Various Domains

Yejun Li
Electrical and Computer Engineering
A15706823

Jiawen Zeng
Electrical and Computer Engineering
A15709448

Abstract

Our reimplmented the conditional Generative Adversarial Network (cGAN) introduced by Phillip Isola et al. [2018] Our input would be a low-information image, such as semantic segmentation image or instance segmented and our output would be a realistic image. We plan to investigate the difference in generated performance when feeding in with different type of segmented labels.

1 Introduction

The problem we try to tackle is that given an image in low information (e.g. objects are represented by color block) how do we develop a model to learn to produce natural images from this limited information. This problem is a subset of image-to-image translation, which transforms images from one domain to images from another domain with different characteristics and style. The formulation of our problem is: we have an image from color block domain X and an image from the natural domain Y. We try to learn a mapping $G : X \rightarrow Y$, such that the generated $G(X)$ would try to have a distribution that is indistinguishable from Y.

Current existing works has image translation from semantic segmentation to realistic photo. We want to also experiment on image translation of (instance segmentation $\rightarrow$ realistic photo) and (panoptic segmentation $\rightarrow$ realistic photo). We want to **test the hypothesis** 1) given instance segmentation, what would be the model learn to generate in the void region and 2) given panoptic segmentation with more detailed information, will the generated image has finer detailed on each instance?

In our case, we utilized Conditional Generative Adversarial (cGAN) model, which has two components. The Generator tried to learn a mapping to transform images from various types of segmentation (semantic segmentation, instance segmentation, panoptic segmentation) and to images with realistic RGB street scene; and the Discriminator tries to discriminate real images from the generated image to help with the learning. The difference between GAN and cGAN is that GAN maps from random noise z to output y and cGAN maps from an observed image x and the random noise z to the ouput y. Since in the process of generating the realistic image, we have information from the segmented image that could be based on. Thus we choose cGAN over GAN in modeling.

The ability to produce close-to-realistic images from segmented images might help us decoding the information lost in compressing the image during transmission. We could imagine semantic segmentation/panoptic segmentation would be utilized for self-driving decision due to its ability to maintain the "useful" information from the view and ignore the unnecessary detail and promising for information transfer. Generating close-to-realistic images could help us have a better understanding from our human point of view in the decision-based segmented images and possibly help on debugging the decision process.

2 Related Work

The fundamental idea of cGan is similar to GAN model. In cGan, there also exists a Generator and discriminator. The loss function of cGan is similar to the loss function of GAN. The difference is that

the discriminator is conditioned on both the generated images and the realistic image. In addition, the loss of cGan also adds extra L1 loss inspired by the work of [1] that adding L2 loss boost performance for inpainting. cGan is not the only paper utilize GAN and conditioned on extra input. As mentioned in the cGan paper,other papers such as context inpainting[1],super-resolution image [2],image style transfer[3],future state prediction[4]. Comparing to these model, cGan was designed for general image-to-image translation problem instead of specific tasks.

In addition to the usual CNN model, cGAn also includes UNet[5] as its generator and introduced a model called PatchGan as its discriminator, which also has a similar structure to capture local style statistics with scale of image patches as in [3].

3 Method

Our method is implemented around the Conditional Generative Advasarial Network mentioned in "Image-to-Image Translation with Conditional Adversarial Networks"[6]. At a high level, the architecture involves a Generator and a Discriminator.

The Generator takes in a segmented image x as input and produces a generated image $G(x)$ as output. We adapted the U-Net structure for our generator (Fig. [1]) to retain some low-level information shared between the input and the output. We further experiment an enhancement of Generator structure by using the Attention Block. Since in Homework 4, we observed the improvement of image learning by using self-attention blocks to capture the information in high frequency. Our attention-based U-net structure is modified by adding attention block between two convolution layers in encoder network as illustrated in Fig. [2]. .

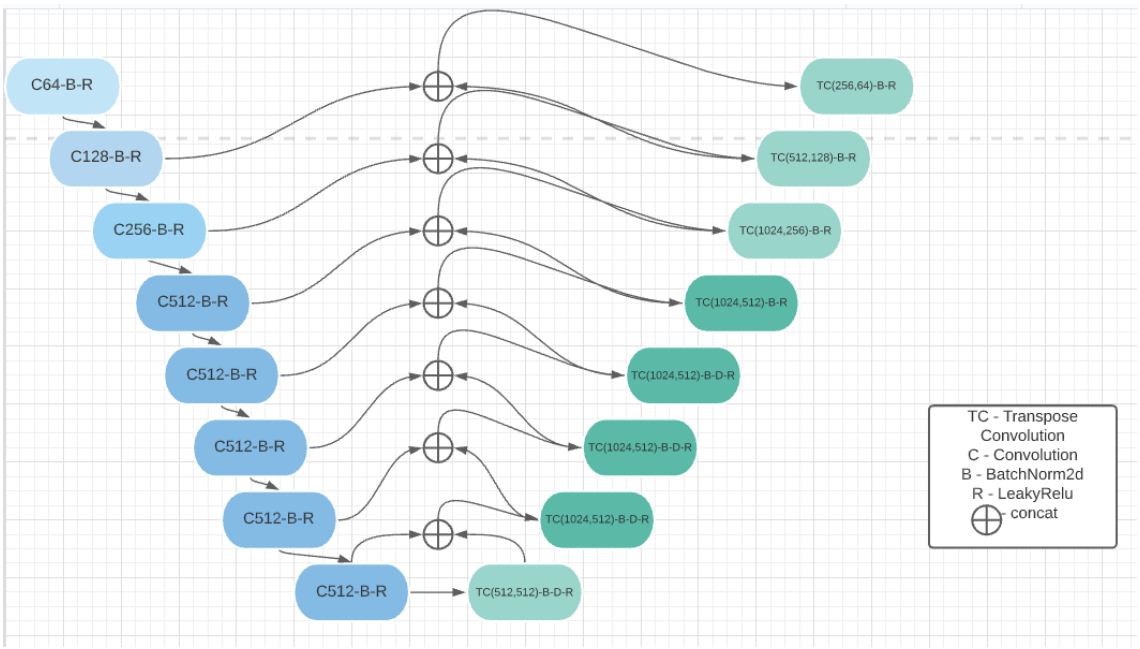

Figure 1: U-Net Generator Architecture

The Discriminator takes in a generated image and identify it between real and fake image. It produces the output $D(G(x))$ represents the probaility of whether $G(x)$ is real. The benefit from the discriminator is that we could learn a more flexible goal from structured loss instead of manually forcing it to match the original image, which ignores the variety of the same objects in different images. Our discriminator is designed in the same structure as PatchGAN, which is a sequential structure of convolution layers as illustrated in Fig. [3].

The goal of the conditional GAN is defined as

$$L_{cGAN}(G, D) = E_{x,y}[logD(x, y)] + E_{x,z}[log(1 - D(x, G(x, z)))] \quad (1)$$

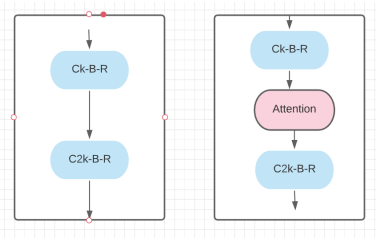

Figure 2: Adapt Attention

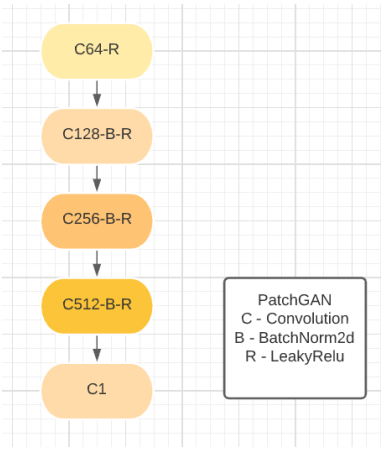

Figure 3: Discriminator

. The discriminator is trying to maximize the loss between the produced images and the ground truth RGB images while the generator is trying to minimize the loss such that the produced image as close as to the RGB images. In [1] also showed that it would be beneficial to also using an L1 loss for the generator optimization to encourage less blurring. Thus, in detailed, the Discriminator would be a sum binary cross entropy loss (BCEWithLogitsLoss) between the target (0 for fake and 1 for real) and the output ($D(G(x))$) and ($D(y)$). The Generator would be a sum of binary cross entropy loss that tries to learn $G(x)$ as real as possible ($D(G(x))$ matches with target real) and the L1 loss between $G(x)$ and y. Both the network would be updated separately with Adam optimizer.

4 Experiment

4.1 Dataset

We used the street scene dataset from Synscapes [7]. The dataset provides annotation with semantic label and instance label. We also combine the semantic label and instance label follow the Cityscapes convention for thing and stuff classes to generate panoptic label for further experimenting on the effect of different types of labels on image translation. In order to train within a reasonable time with single GPU, We select 500 images for training and 100 images for testing. An example of the dataset is provided below.

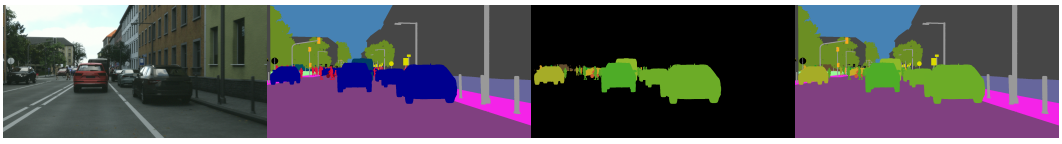

Figure 4: Realistic RGB Figure 5: Semantic Figure 6: Instance Figure 7: Panoptic

The difference between different type of segmentation is that

- Semantic Segmentation: label different object with the same class with the same color
- Instance Segmentation: only label the countable object and label the same instance with the same color
- Panoptic Segmentation: label 'countable' object in the same way as instance segmentation and label 'amorphous' object in the same way as semantic segmentation

4.2 Result

We have trained the network for 100 epochs and compared the generated result from (semantic -> photo), (instance -> photo) and (panoptic -> photo). Since the normal evaluation metrics like per-pixel mean squared error do not serve to measure such task, we evalute the result from the test set with perceptual studies:

Here's an example of the generated photo from each experiment:

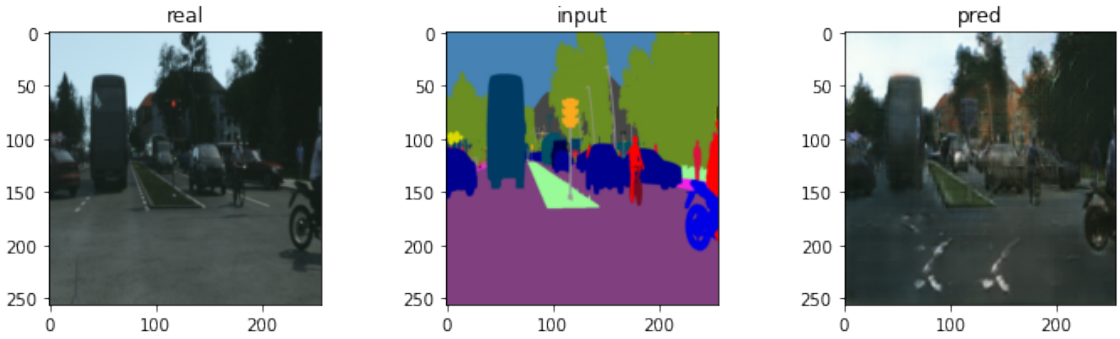

Figure 8: Sematic Label -> Generated Photo Result

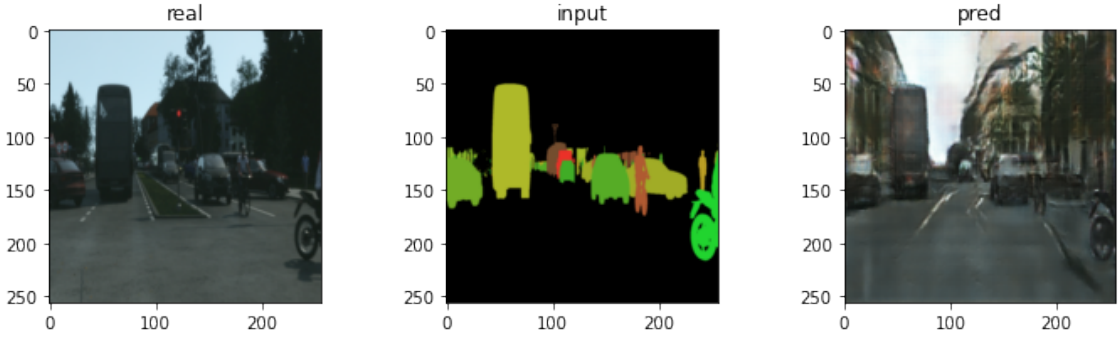

Figure 9: Instance Label -> Generated Photo Result

We can see that for the same image, semantic label (Fig. [8]) cannot well distinguish object belongs to the same class. While using the instance label (Fig. [9]), we provide zero information about its background (infrastructure, sky, road, etc), the generator can learn the general structure for these background for example the upper part of the image is usually the sky, the car and the pedestrian are usually walking on the road. Although when we focusing on the detail of the background, such as the building, we can see random windows and doors popped out as texture. As for panoptic, we observe that panoptic require more epochs to learn the texture as each image has more category than the other two type of segmentation.

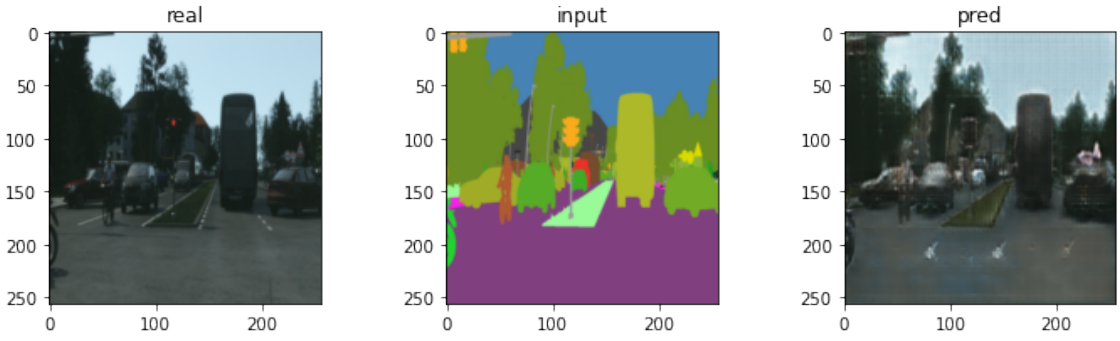

Figure 10: Panoptic Label -> Generated Photo Result

4.3 Ablation Study

We also experimented on modifying different parts of the architecture of the network. We constructed two network described in the cGan paper.The two network have different generator and same discriminator. The first network is the usual CNN model (The result is in Fig. [11]) which has an encoder-and-decoder structure. The second network incorporated UNet for the structure (The result is in Fig. [8]).

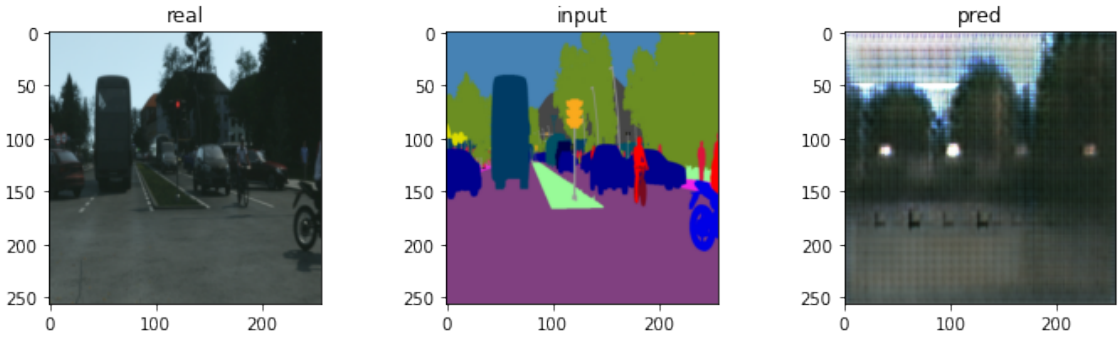

Figure 11: CNN model

In comparison, we found that CNN model without proper skip-connection model can not inpaint segmentation images with expected details. During training, we discovered the model tends to print repeated features over the entire image as seen in Fig. [11]. In contrary, the UNet is about to incorporate feature from difference classes as seen in FIG. The comparison reveals that the skip-connection served a critical role in information flow.

In addition, we also implemented an new model using the self-attention module and UNet (Fig. [2]) . The inspiration is that we discovered some repeated pattern painted over the flat area in the image (i.e. rode, building surface). The improvement is as obvious as the comparison between traditional CNN model with UNet. As we can see in Fig. [12].By comparing the result,the self-attention module helped the model to capture more detail texture among different classes.

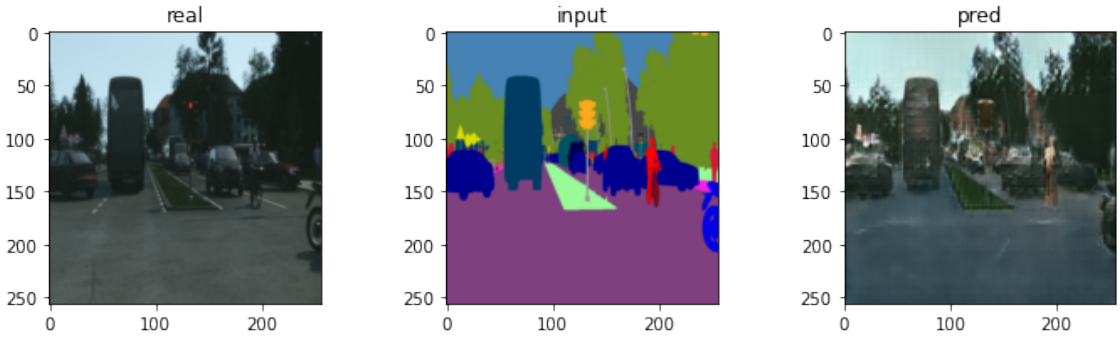

Figure 12: UNet with Attention

References

[1] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. *CoRR*, abs/1604.07379, 2016.

[2] 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.

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

[4] Yipin Zhou and Tamara L. Berg. Learning temporal transformations from time-lapse videos. *CoRR*, abs/1608.07724, 2016.

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

[6] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018.

[7] Magnus Wrenninge and Jonas Unger. Synscapes: A photorealistic synthetic dataset for street scene parsing. *CoRR*, abs/1810.08705, 2018.