Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros

https://arxiv.org/abs/1703.10593
https://arxiv.org/pdf/1703.10593.pdf

TL;DR

The authors present an approach to translate an image from a source domain X to a target domain Y in the absence of paired examples. In other words, the authors propose a new unsupervised method of transferring an image's style to another.

Introduction

A new method is presented to capture the special characteristics of one image collection and figure out how they could be translated into another image collection, all in the absence of any paired examples. In simple terms, the authors propose a new method of unsupervised image-to-image translation. Since obtaining paired data is extremely difficult as it requires artistic authoring, we seek an algorithm that can learn to translate between two domains without paired input-output examples.

To strengthen the relationship between the input and the output, the authors exploit the property that translation is bidirectional. Translating a sentence to french then back to english, should result in the original sentence. This is applied by training the translator $G: X \rightarrow Y$ and $F: Y \rightarrow X$ simultaneously, and adding a cycle consistency loss that encourages $F(G(x)) \approx x$ and $G(F(y)) \approx y$. Combining this loss with adversarial losses on domains X and Y result in unpaired image-to-image translation.

Formulation

The goal is to learn mapping functions between two domains X and Y. The model includes two mappings $G: X \rightarrow Y$ and $F: Y \rightarrow X$. It also contains two adversarial discriminators D_x and D_Y. D_x aims to distinguish between images $\{x\}$ and translated images $\{F(y)\}$. Similarly D_Y aims to discriminate between $\{y\}$ and $\{G(x)\}$. Adversarial losses are used for matching the distribution of generated images and cycle consistency losses are to prevent the learned mapings G and F from contradicting each other.

Adversarial loss is applied to both mapping functions. Objective is:

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))],$$

A similar loss function is introduced for D_X and $F: Y \rightarrow X$

$$\min_F \max_{D_X} \mathcal{L}_{\text{GAN}}(F, D_X, Y, X).$$

Cycle consistency loss

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1].$$

All Together(lambda controls the relative importance of the two objectives)

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F),$$

We aim to Solve:

$$G^*, F^* = \arg\min_{G,F} \max_{D_x, D_Y} \mathcal{L}(G, F, D_X, D_Y)$$

Implementation

The architecture contains three convolutions, several residual blocks, two convolutions with stride ½, and one convolution that maps features to RGB. The discriminator networks use 70x70 Patch GANs.

Results

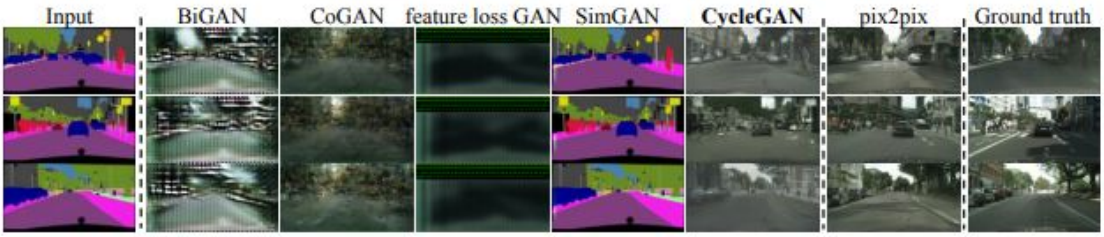
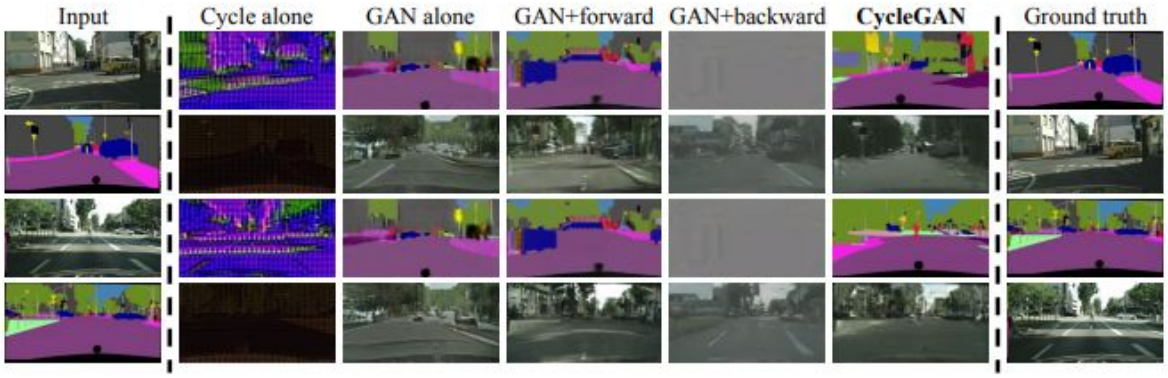


Figure 5: Different methods for mapping labels↔photos trained on Cityscapes images. From left to right: input, Bi-GAN/ALI [7, 9], CoGAN [32], feature loss + GAN, SimGAN [46], CycleGAN (ours), pix2pix [22] trained on paired data, and ground truth.



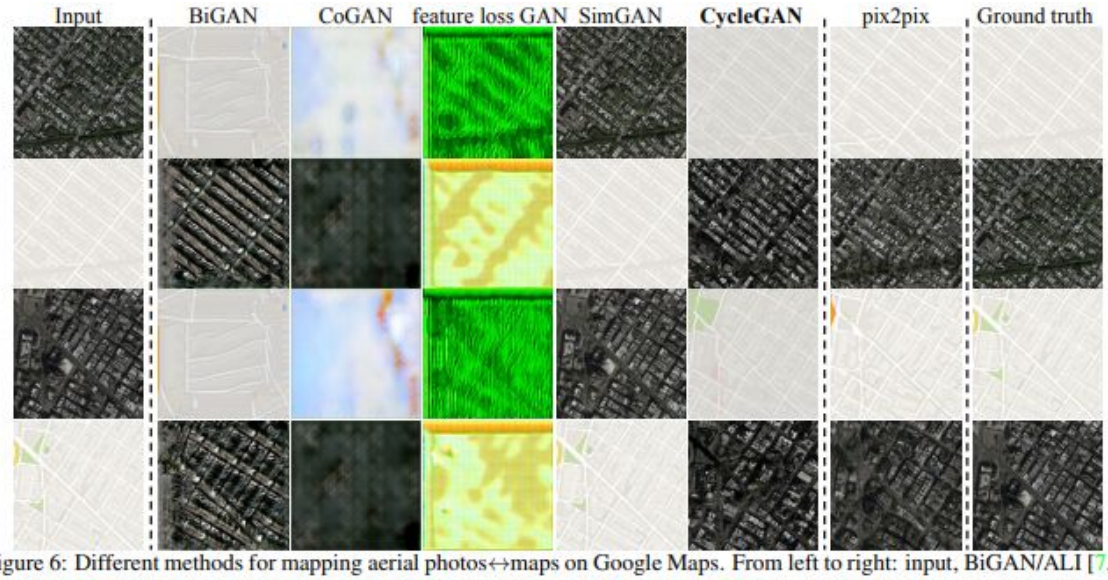
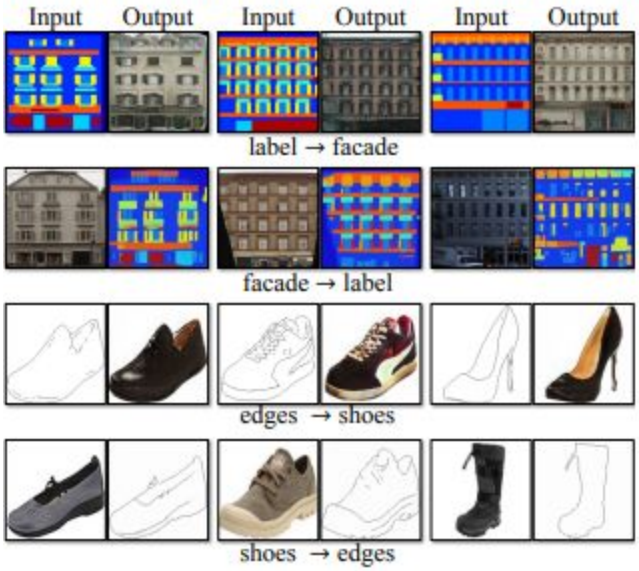


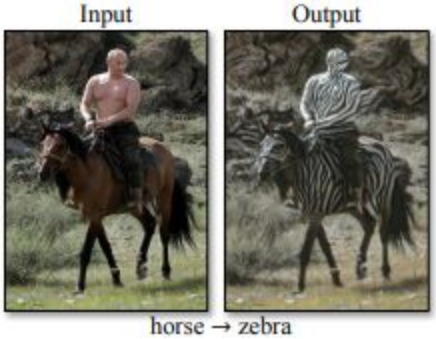
Figure 6: Different methods for mapping aerial photos↔maps on Google Maps. From left to right: input, BiGAN/ALI [7, 9], CoGAN [32], feature loss + GAN, SimGAN [46], CycleGAN (ours), pix2pix [22] trained on paired data, and ground truth.

producing identical label maps regardless of the input photo.

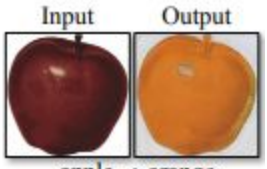


Limitations and Discussions

Although this method can achieve compelling results in many cases, results are far from uniformly positive. This method often succeeds in translation tasks involving color and texture changes. Tasks with geometric changes often fail.



horse → zebra



apple → orange