

Image to Image Translation using Deep Learning Techniques

S. Ramya

Assistant Professor
Department of Computer
Science and Engineering
Kongu Engineering
College, Erode, India

S. Anchana

Department of Computer
Science and Engineering
Kongu Engineering
College
Erode, India

A.M. Bavidhrra

Shrrei
Department of Computer
Science and Engineering
Kongu Engineering
College
Erode, India

R. Devanand

Department of Computer
Science and Engineering
Kongu Engineering
College
Erode, India

ABSTRACT

This project aims to make image-to-image translation for different tasks using deep learning techniques. Using dataset generated by a combination of online images. We have generated images for different tasks. Our innovation is we train and test the cycle-consistent adversarial networks using dataset. We have tried to fine-tune hyper parameters such as batch size, learning rate, lambda in loss function and trying to add dropout to our network. The experiment outcomes show that our method can successfully transfer for disparate tasks while conserving the original content.

Keywords

CycleGAN, CNN, Adversarial losses

1. INTRODUCTION

Image style transfer has been welcomed these days for creating samples for visualizing interior design, computer games items and create drawings. Lots of people use filters on camera to get funny pictures (ex:snapchat,Instagram). Given both content image and style image, photographic style transfer will regenerate the content image in the style of the style image, which are used for transfer effects like time of day, season, and illumination. Existing techniques are not sufficient to produce accurate transferred images, so we want to find more and try to improve the existing algorithms. For example, if we transfer the style of a painting to a photograph, the final output shows us a very generalized style, because a painting commonly has a consistent style, but a photographs tend to have a more localized style. In addition, for many tasks, paired training data will not be available. We present a method to translate an image from a source domain to a target domain with unpaired examples using CycleGAN (Cycle- consistent Generative Adversarial Networks). The project is mainly about transferring the look of one photo onto another, while still looking like a photo, and especially pointing on implementing CycleGAN for unpaired examples. The adversarial losses and cycle consistency losses make CycleGAN more powerful than traditional methods. With this technique, we can not only change input images into artistic styles of paintings, but also make object transfiguration such as changing between a horse and a zebra, an apple and orange and even changing the background environment into a snow scene. Our techniques can be applied to a wide range of applications, including style transfer, object transfiguration and photo enhancement. These applications can be adapted by companies like Instagram or Snapchat to create interesting filters, or even by autonomous driving companies to improve their performance based on the images we generated.

2. RELATED WORK

CNN has been proved to have powerful aptitude for image style transformation. Our work with CNN is based on the paper Deep Photo Style Transfer by Yanghao Li [1]. This paper builds upon the work Neural Style Transfer by Fujun Luan. [2]. Throughout this paper we will give an in depth report of image style transfer as described in [1] and discuss specifics of how to enhance the original algorithm to induce better to two unpaired images. The largest in this method are gained through CycleGAN. This newly developed algorithm is more powerful than the traditional CNN to do image style transfer. Our work with CycleGAN is based on the paper Unpaired Image-to-Image Translation using Cycle- Consistent Adversarial Networks by Zhu et al [3]. Building on the model introduced in the paper, we use combined datasets to do different image style transferring tasks.

2.1 Generative Adversarial Networks

Generative Adversarial Networks (GANs) [4] have achieved good results in image generation and image editing. The GANs' success is because of an adversarial loss that impacts the generated images to be, in principle, indeterminate from real photos. This loss is particularly forceful for image generation tasks, that most of computer graphics aims to optimize the result so that they achieve the target.

2.2 Cycle-Consistency

Cycle-Consistency uses transitivity to normalize structured data. In visual tracking, imposing simple forward backward consistency was a standard trick. Recently, higher-order cycle consistency has been used in structure from motion, 3D shape matching and depth estimation. Our work is very similar to that of Zhu et al [3], Zhou et al. [6] and Go- dard et al. [7].

3. DATASET AND FEATURES

3.1 Dataset Preparation

We gather our dataset for both neural style transfer and cycle-consistent adversarial networks from google internet. For Neural style transfer task, we choose to use images directly from the internet, and use some painting images as our style image. We then scale our images to 300*400 pixels. For cycle-consistent adversarial networks, depending on different applications, we choose to use different dataset. Our dataset images are scaled to 256*256 pixels. And our training set contains original images and snowy images. For test set, we randomly download some images from the internet to gather the result. Horse to Zebra and Apple to Orange task. We choose to use dataset provided by google internet that contains horse, zebra, apple and orange as

our training set. We scale the images to 256*256 pixels. For test set, we randomly download some images from the internet and gather the result.

4. METHODS

4.1 Neural Style Transfer

First we used Neural Style Transfer to perform image-to-image translation, which fit an image by combining the content of one image with the style of another image. We used VGG19 convolutional neural network and we defined a cost function for both content and style pictures and then use gradient descent to minimize the combined cost. Rather than mapping one specific image to another one, we trained the network to learn the mapping from one collection to another collection. This learning can be applied to other tasks and change all the new input images to have the same feature.

4.2 Cycle-Consistent Adversarial Networks

The second method is combining adversarial losses and cycle consistency losses to learn mapping functions between two domains.

4.3 Adversarial Loss

We used the following formula as the adversarial loss

$$GAN = [\log D_Y + - D_Y(G)] \quad (1)$$

where D_Y is a discriminator that encourages G which is a mapping function that maps domain X to Y to translate image X into output that is indistinguishable from domain Y . The first E is the expected value taken from the data distribution and the second E is the expected value taken from the data distribution $p(y)$. In the above formula, the purpose of the mapping function G is to generate images $G(x)$ that look similar to images from domain Y . We used a similar function for the mapping function F which maps from domain Y to X and the adversarial loss is as follows:

$$LGAN = E [\log D_X(x)] + E [\log(1 -)] \quad (2)$$

with the first E taken from the data distribution and the second E taken from the data distribution. Using adversarial loss alone cannot guarantee every input x mapped desire output y , because with large capacity, a the data distribution and the second E taken from the data distribution.

4.4 Cycle Consistency Loss

Network can map the same set of given images to any random images of the target (unpaired images). To reduce the mapping function's dimensions, we need to make sure that the image translation cycle will bring the x back to the original image. So we used the following Cycle Consistency Loss formula:

$$\mathcal{L}_{Cyc}(G, F) = E_{x \sim p_{data}(X)} [\| F(G(X)) - X \|] + E_{y \sim p_{data}(Y)} [\| F(G(Y)) - Y \|] \quad (3)$$

where F is a mapping from Y to X and G is a mapping from domain X to Y . In order to be cycle-consistent, $F(G(x))$ should map back to x and $G(F(y))$ should map back to y .

4.5 Network Architecture

We use the architecture from Johnson et al. As the starting point of our network, because it showed results for neural style transfer. The network contains double stride-2 convolutions, many residual blocks [9], and a double fractionally strided convolutions with stride 1/2. Here 6 blocks are used for 128 x 128 images and nine blocks for 256 x 256 images and higher resolution training images. Similar to Johnson et al. [8], here instance normalization is also used [10]. We use 70 x 70

PatchGANs [11] for the discriminator networks, which classifies whether 70 x 70 overlapping image patches are fake or real. Such a patch-level discriminator architecture has numbered parameters than a full-image discriminator and can function on arbitrarily-sized images in a fully convolutional fashion [11].

4.6 Training Details Training Environment

For neural style transfer, we simply use MacOS CPU to generate the result. The training environment is shown in table 1.

Table 1. Training Environment

Language	Python3.5
Framework	Google TensorFlow 1.4.0

For cycle-consistent adversarial networks, we choose to use a GPU to train our network from scratch. Our training environment is shown in table 2.

Table 2. Training Environment

Language	Python3.5
Framework	PyTorch 0.4
GPU	Nvidia GeForce GTX 960M
GPU Memory	4044MB

4.7 Initialization

Weights are initialized randomly with zero mean as truncated normal distribution and 0.001 as standard deviation. With proper data normalization, it is assumed that approximately half weights will be positive and half will be negative. Hence, we want the weights to be very close to zero, but not equal to zero.

4.8 Learning Rate

The learning rate is 0.00025 for the first 200 epochs and we use linear decay every 10 epochs till the end. The learning rate for the latter is quite smaller, since we are fine tuning the model to achieve the result. Out of our several poolings we find this one can obtain both good improvement and high efficiency in the process.

4.9 Batch Size

Unlike the original paper, we use batch size for gradient descent equal to 5. We choose batch size not same as I to elude overfitting. And we observe a better trade-off between performance and hardware limitation.

4.10 Update Method

We applied Adam update, which in practice works better than standard momentum.

4.11 Regularization

We have also used dropout method as regularization technique, which can decrease overfitting in neural networks by blocking complex network, co-adaptations on training data. In this experiment, drop probability is equal to 0.4, since it can maximize number of randomly-generated network structures [12].

4.12 Loss Function

Our full objective is:

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

where the first two are adversarial losses aiming to minimize $G(x)$ and y ; $F(y)$ and x . The last part is the cycle consistency loss, aiming to reconstruct images $F(G(x))$ to match the input images x . We have tried to fine-tune λ by a number of combinations and find the original 10 to be the best choice.

5. RESULT

Our tasks include: Neural style transfer task, Making snow on the image task, Horse to Zebra task, Apple to Orange task and Monet painting to photo task. For neural style transfer task, we perform 200 iterations for a given content and style image, and used a pre-trained VGG model to generate the artistic image. For other tasks we perform 250 epochs for training, and after training we also performed a single generation on our test-set to generate images.

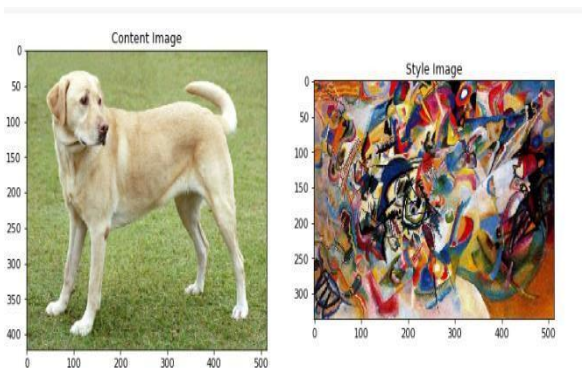


Fig1: Input images



Fig 2: Output Image

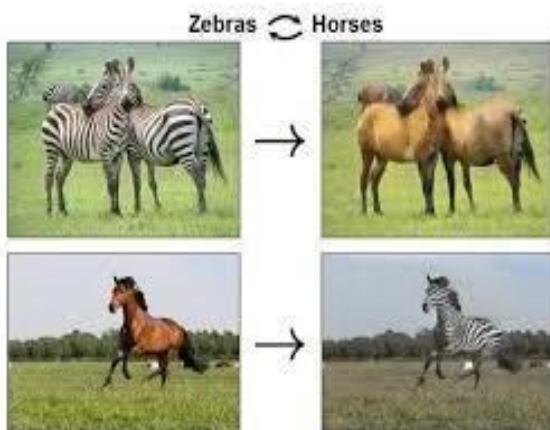


Fig 3: Object Configuration

6. REFERENCES

- [1] L. Gatys, A. Ecker, and M. Bethge. A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576, 2015.
- [2] J. Zhu, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, 2017.
- [3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In NIPS, 2014.
- [4] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-backward error: Automatic detection of tracking failures. In ICPR, 2010.
- [5] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Aug. 2015, pp. 3431–3440.
- [6] P. Isola, J. Y. Zhu, T. Zhou, and A. Efros, “Image-to-image translation with conditional adversarial networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Aug. 2017, pp. 1125–1134.
- [7] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. (2017). “Learning to discover cross-domain relations with generative adversarial networks.” [Online]. Available: <https://arxiv.org/abs/1703.05192>
- [8] Z. Yi, R. Hao, P. Tan, and M. Gong, “DualGAN: Unsupervised dual learning for image-to-image translation,” in Proc. ICCV, Aug. 2017, pp. 2868–2876.
- [9] j. Y. Zhu, T. Park, P. Isola, and A. A. Efros. (2017). “Unpaired image-to-image translation using cycle-consistent adversarial networks.” [Online]. Available: <https://arxiv.org/abs/1703.10593>.
- [10] M. Y. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-image translation networks,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 700–708
- [11] X. Huang, M. Y. Liu, S. Belongie, and J. Kautz. (2018). “Multimodal unsupervised image-to-image translation.” [Online]. Available: <https://arxiv.org/abs/1804.04732>.
- [12] Y. Choi, M. Choi, and M. Kim, “Stargan: Unified generative adversarial networks for multi-domain image-to-image translation,” Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2018, pp. 3662–3670.
- [13] H. Y. Lee, H. Y. Tseng, J. B. Huang, M. Singh, and M. H. Yang, “Diverse image-to-image translation via disentangled representations,” in Proc. ECCV, Sep. 2018, pp. 35–51.
- [14] N. Liu, J. Han, and M. H. Yang, “PiCANet: Learning pixel-wise contextual attention for saliency detection,” Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Aug. 2018, pp. 3089–3098.