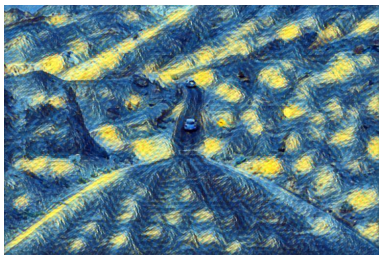
Depth-Preserving Style Transfer

Xiuming Zhang
Massachusetts Institute of Technology
xiuming@mit.edu

Teammates: Ruizhi Liao & Yu Xia

Style
Starry Night

(A) Scene with large variations in depth



(B) Johnson *et al.*, 2016

(C) Our depth-preserving results

When the input scene exhibits large variations in depth (A), the current state of the art tends to destroy the layering and lose the depth variations, producing a "flat" stylization result (B). This paper aims to address this issue by incorporating depth preservation into the loss function such that variations in depth and layering are preserved in the stylized image (C).

Abstract

Style transfer is defined as the process that given a content image and a style image it tries to migrate the style from the style image to the content image. Though it is not clear what the exact definition of style is, pattern transforming and matching are generally accepted.

In this work we present a novel method which preserve the depth information of the content image while migrating the style.

1. Introduction

convolutional neural network (CNN)

2. Related Work

The core of our method is incorporating depth preservation losses into the image transformation neural network. Therefore, we review related literature on both neural network-based image style transfer and single-image depth estimation.

2.1. Image Style Transfer with Neural Networks

Style transfer can be considered as a more general form of texture transfer, where one transfers texture from one image (style image) to another image (content image). Ideally, semantics of the content image should not be altered in this process. In texture transfer, it is usually the low-level features that are utilized, *e.g.*, in [4].

With the recent prevalence of deep neural networks, researchers started exploring how high-level features extracted by neural networks can be utilized for the task of style transfer. For instance, Gatys *et al*. perform image style transfer by synthesizing a new image that matches both contents of the content image and styles of the style image [8]. In particular, they extract content representations from the content image and style representations from the style image using the VGG network [15]. Since the VGG network is trained to perform object recognition and localization tasks, the layers deep down the network hierarchy capture object information (i.e., the contents) of the content image and are insensitive to the exact pixel values. Therefore, outputs from these deep layers serve as good content targets that the synthesized image tries to achieve at varying levels of resolution. As for style, they adopt a feature space built on filter responses in any layer of the network [6]. By design, the feature space captures texture information without global arrangement. Finally, they minimize a weighted sum of the content and style loss under a CNN framework, where forward and backward passes are iteratively performed. Build-

ing upon this work, the authors recently devised a way of preserve the original colors in the content image [7]. However, the high computational cost still remains as a drawback in [8].

To reduce the computational burden and generate visually similar-quality results, Johnson *et al*. [9] train a feed-forward image transform network to approximate solutions to the optimization problem posed in [8]. In particular, their system consists of a deep residual CNN as the image transform network and the pretrained VGG network [15] as the fixed loss network. For each style image, the image transform network is trained to apply this style to a content image while minimizing the style and content losses as measured by the loss network. This method produces reasonably good results with low computational cost, but tends to lose the depth variations and destroy layering in the content image as illustrated in the teaser figure. This issue can be addressed by incorporating depth preservation losses into the loss function, as shown later in this paper.

2.2. Single-Image Depth Estimation

Deep neural networks trained on ground-truth metric depth data have demonstrated promises in the task of single-image depth estimation [13, 5, 11, 17]. Collecting such ground truth requires specialized cameras, such as Kinect, posing a challenge to large-scale data collections. Although crowdsourcing may seem to be a solution, humans are known bad at estimating absolute depths (which are inherently ambiguous from a single monocular image), but better at at judging relative depths [16]. Inspired by this fact, Zoran *et al*. train a neural network to repeatedly judge relative depths of point pairs and interpolate out per-pixel metric depth by solving an optimization problem [18].

Building on [18], a recent work by Chen *et al*. proposes an end-to-end neural network that takes in a single RGB image in the wild (*i.e*., taken in unconstrained settings) and outputs pixel-wise depth estimations [1]. Specifically, the deep network follows the "hourglass architecture" recently proposed in [14], which is essentially a series of convolutions and downsampling followed by a series of convolutions and upsampling. Similar to [18], RGB images with relative depth annotations are used as training data. The loss function penalizes large differences in metric depth when the ground-truth relative depth is annotated equal.

3. Methods

The overview of our network structure is shown in Figure 1. Compared to Johnson et al. [9]'s work, our structure is featured in having a depth estimation network as part of our loss function. In all, our network is composed of 3 subnets: an image transformation network f_W, a perceptual loss network ϕ and a depth estimation network δ. Similar to Johnson et al., the image transformation f_W is a convolutional network which produce the output image $\hat{y}$ given the input image x by $\hat{y} = f_W(x)$ (where W is the weights of the network).

To keep track of the depth information, our loss function is composed of 2 neural networks: the perceptual loss network and the depth estimation network. As mentioned in [9], pretrained convolutional neural networks are able to extract perceptual information and encode semantics which are useful for the loss function. Similarly, a pretrained depth estimation network has already learned to estimate the depth information from the single input image. Therefore we utilize a pre-trained image classification network for the perceptual loss part and a pretrained depth estimation network for the depth loss part. Specifically, our loss function is defined as a weighted linear combination of the content loss l_{content}, the style loss l_{style} and the depth loss l_{depth}.

$$L(\hat{y}, y) = \lambda_1 l_{\text{content}}(\hat{y}, y) + \lambda_2 l_{\text{style}}(\hat{y}, y) + \lambda_3 l_{\text{depth}}(\hat{y}, y)$$

Therefore the training goal is to minimize the expected loss.

$$W^* \leftarrow \arg\min_W \mathbb{E}_{\{x,y\}}[L(f_W(x), y)],$$

where $\mathbb{E}_{\{x,y\}}$ is the estimation of the expectation via the (training) set $\{x, y\}$.

3.1. Depth Loss Function

We make use of depth loss function to measure the amount of depth differences between the input image x and the output image $\hat{y}$. Ideally, the output image should have similar depth features with that of the input. Rather than capture the per-pixel differences of the feed-forward outputs we capture the high level features from the depth estimation network. More specifically, we define the depth loss function l_{depth} as the 2-norm of the feature vectors (from selected layers)

$$l_{\text{depth}}(\hat{y}, y) = \sum_{i \in I_\delta} \frac{1}{N_i(\delta)} \|\delta_i(\hat{y}) - \delta_i(y)\|_2^2$$

, where $N_i(\delta)$ is the normalizing factor for the i-th layer in δ and $\delta_i(y)$ indicates the feature vector on the i-th layer if y is feeded as the input to the network δ. The layer set I_δ means the set of (high-level) layers we want to extract features from. The motivation for a high level depth loss function is that we want to encourage the output from f_W to be similar to the content image from the depth pespective but we don't want their depth estimation to be exactly the same. There are several reasons for such a motivation: firstly, the estimations of depth from the network ϕ are not necessarily accurate which make it meaningless to pursue a (per-pixel) exact match on the depth estimation. Secondly, we

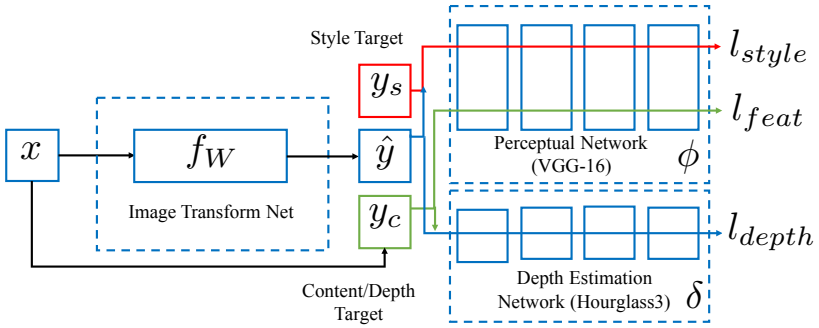


Figure 1: Network Structure Overview

need to allow the image transformatin network f_W to perceptually transform the image which might involve changes of shapes, places and lines. Again it is not promising to propose a per-pixel loss, which reduce the chances of such transformations. Thirdly, as argued in [9], perceptual losses are more robust and stable than the per-pixel losses.

3.2. Content Feature Loss Function and Style Loss Function

For content feature loss function l_{feat} and the style loss function l_{style}, we briefly recall the explanation from [9]. As one of the main contributions in Johnson et al.'s paper, l_{feat} and l_{style} are both measures of differences of high-level features. l_{feat} captures the distances in respect of perceptual features between the content target y_c(i.e., the input image x) and the output image $\hat{y}$. Similarly, as proposed from Gatys et al. [8], l_{style} captures the distances between the style image y_s and the output image $\hat{y}$. Therefore

$$l_{\text{feat}}(\hat{y}, y) = \sum_{i \in I_\phi} \frac{1}{N_i(\phi)} \|\phi_i(\hat{y}) - \phi_i(y)\|_2^2,$$

and for style loss we use the Frobenius norm of differences of the Gram matrices of $\hat{y}$ and y_s.

$$l_{\text{style}}(\hat{y}, y_s) = \sum_{i \in I_\phi} \frac{1}{N_i(\phi)} \|G_i^\phi(\hat{y}) - G_i^\phi(y_s)\|_F^2.$$

4. Experiments

4.1. Training Details

Microsoft COCO dataset [12] (contiaing around 80K images) was used for training our depth-preserving style transfer networks. Each training image was resized to 256×256. Maximum iterations were set to be 40000, and a batch size of 3 was applied. These settings gave roughly 1.5 epochs over all the training data. The optimization was based on Adam [10] with a learning rate of 1×10^{-3}. No weight decay or dropout was used. The training was implemented using Torch [3] and cuDNN [2]. Each style training took around 7 hours on a single GTX Titan X GPU. As for the target losses, the content target loss is computed at VGG network layer $relu2\_2$, the style target loss is computed at VGG network layers $relu1\_2$, $relu2\_2$, $relu3\_3$ and $relu4\_3$, and the depth target loss is computed at the output layer of Chen *et al*. network [1].

4.2. Qualitative Results

5. Discussion and Conlucsions

Some might argue that if the weight for the style target loss in the Johnson *et al*. [9] network is decreased, we could get similar results. But this is not the case. Note that the content target loss term is computed as the distance of the feature representations in the VGG network [15], which was designed and trained for object recognition. The backgrounds in the content images, for instance, skies, roads, and lawns, are hard to be fully represented in those features. Also, the depth of an "object" (for example, the road in the teaser image) may change a lot in an image, and some pixel-level (or superpixel-level) loss should be included to preserve stereopsis of the transferred images. Therefore, an extra depth loss is necessitated in our problem setting.

References

[1] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. *arXiv preprint arXiv:1604.03901*, 2016.

[2] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer. cudnn: Efficient primitives for deep learning. *arXiv preprint arXiv:1410.0759*, 2014.

[3] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, number EPFL-CONF-192376, 2011.

[4] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM, 2001.

[5] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolu-

tional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015.

[6] L. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 262–270, 2015.

[7] L. A. Gatys, M. Bethge, A. Hertzmann, and E. Shechtman. Preserving color in neural artistic style transfer. *arXiv preprint arXiv:1606.05897*, 2016.

[8] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016.

[9] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *arXiv preprint arXiv:1603.08155*, 2016.

[10] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[11] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1119–1127, 2015.

[12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[13] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5162–5170, 2015.

[14] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. *arXiv preprint arXiv:1603.06937*, 2016.

[15] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[16] J. T. Todd and J. F. Norman. The visual perception of 3-d shape from multiple cues: Are observers capable of perceiving metric structure? *Perception & Psychophysics*, 65(1):31–47, 2003.

[17] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. Yuille. Towards unified depth and semantic prediction from a single image. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2800–2809. IEEE, 2015.

[18] D. Zoran, P. Isola, D. Krishnan, and W. T. Freeman. Learning ordinal relationships for mid-level vision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 388–396, 2015.