Depth-aware Neural Style Transfer

Xiao-Chang Liu
Ming-Ming Cheng*
CCCE, Nankai University

Yu-Kun Lai
Paul L. Rosin
Cardiff University

(a) Style: The Muse by Pablo Picasso (b) Content (from Pixabay) (c) [Johnson et al. 2016] (d) Ours

Figure 1: Image style transfer results. (a) style image, (b) content image, (c) result of [Johnson et al. 2016] and (d) our depth-aware style transfer result. We can see that when stylizing an image with rich relative depth and spatial distance information, compared to [Johnson et al. 2016], our results can better keep the original layout and relative depth relationships.

ABSTRACT

Neural style transfer has recently received significant attention and demonstrated amazing results. An efficient solution proposed by Johnson et al. trains feed-forward convolutional neural networks by defining and optimizing perceptual loss functions. Such methods are typically based on high-level features extracted from pre-trained neural networks, where the loss functions contain two components: style loss and content loss. However, such pre-trained networks are originally designed for object recognition, and hence the high-level features often focus on the primary target and neglect other details. As a result, when input images contain multiple objects potentially at different depths, the resulting images are often unsatisfactory because image layout is destroyed and the boundary between the foreground and background as well as different objects becomes obscured. We observe that the depth map effectively reflects the spatial distribution in an image and preserving the depth map of the content image after stylization helps produce an image that preserves its semantic content. In this paper, we introduce a novel approach for neural style transfer that integrates depth preservation as additional loss, preserving overall image layout while performing style transfer.

CCS CONCEPTS

•**Computing methodologies** → **Image manipulation;** *Computational photography;* **Non-photorealistic rendering;**

KEYWORDS

Non-photorealistic rendering, deep learning, depth

ACM Reference format:
Xiao-Chang Liu, Ming-Ming Cheng, Yu-Kun Lai, and Paul L. Rosin. 2017. Depth-aware Neural Style Transfer. In *Proceedings of NPAR'17, Los Angeles, CA, USA, July 28-29, 2017,* 10 pages.
DOI: 10.1145/3092919.3092924

1 INTRODUCTION

The goal of non-photorealistic rendering (NPR) is to render a scene in a stylized or artistic manner. Broadly speaking, there are two categories of approaches: image based NPR [Rosin and Collomosse 2013] and 3D model based NPR [Strothotte and Schlechtweg 2002]. The former has wide applicability, but the difficulties in parsing the contents of images without prior semantic knowledge limits the quality of the outputs. In contrast, the latter has the advantage of being able to use the available 3D information to directly compute depth discontinuities, surface normals, etc. These are invaluable when performing rendering, *e.g.* in determining placement of strokes. This paper tackles the problem of image stylization – in particular style transfer – and aims to leverage some of the benefits of 3D based approaches, by inferring 3D information from

*Corresponding author's email: cmm@nankai.edu.cn

NPAR'17, Los Angeles, CA, USA

978-1-4503-5081-5/17/07…$15.00
DOI: 10.1145/3092919.3092924

2D images, in order to improve the quality of results compared to existing methods that operate exclusively on 2D information.

Specifically, this paper investigates image-based style transfer. Given a content image and a style image as input, the aim is to synthesize an output image with the same content as the content image but following the style given in the style image. This is an interesting problem in non-photorealistic rendering. Many methods have been developed for automatic style transfer. With the fast development in deep learning, neural networks have shown increasing power in many areas, especially in the field of computer vision. From image detection [Cheng et al. 2014; Girshick et al. 2014; Ren et al. 2015] to semantic segmentation [Long et al. 2015; Wei et al. 2017, 2016], deep neural networks have made breakthroughs in nearly all the areas. Recent progress has demonstrated deep learning is not only effective at solving those well defined problems with ground truth, but also problems with no ground truth available for training, a typical example being image style transfer (e.g., [Gatys et al. 2016; Johnson et al. 2016; Ulyanov et al. 2016]). For such problems, ground truth results are not well defined and extremely time consuming to obtain. The success of such methods builds on the fact that deep neural networks have an extremely strong feature extraction capability. As we know, features are essential in determining the style of images. Traditional methods usually use handcrafted features typically in the form of mathematical representations, determining e.g. the opacity and shape of a brush stroke. Although some of these algorithms achieve remarkable results, they have major disadvantages: low-level features may fail to capture essential semantic styles or content, and the effectiveness of features is often problem dependent. Deep neural networks provide an alternative where the features are effectively learned, which is generic and achieves significant performance gain compared to traditional methods, thanks to their strong nonlinear representation capability.

Existing deep neural network based image style transfer methods often produce impressive results. However, for challenging input images with multiple objects and complex spatial layout, the synthesized image tends to distribute style elements evenly across the whole image, and make objects in the scene become unrecognizable. This is particularly true for images of scenes covering a wide range of depths. The results are not entirely satisfactory (see an example in Figure 1c). This is probably due to the fact that the pre-trained networks used to define perceptual loss functions were originally designed to perform object recognition, and so their feature extraction ability for style transfer is limited. To supplement this, we propose to add depth reconstruction loss to help train the image transformation network. The depth map captures the structure and overall layout of the scene. Experimental results show that our results achieve the desired style transfer and retain the essential layout of the content image (see Figure 1d).

2 RELATED WORK

Image style transfer and depth prediction are two fundamental problems in computer vision and computer graphics. Earlier research uses traditional methods. With the prevalence of deep neural networks recently, researchers began to consider how to apply them to style transfer and depth prediction. New approaches for style transfer (e.g., [Gatys et al. 2016; Johnson et al. 2016]) and depth prediction (e.g., [Chen et al. 2016; Liu et al. 2015]) with novel strategies are constantly emerging. These methods achieve fairly good results, and provide the basis for this work.

2.1 Image Style Transfer

Traditional Methods. When performing style transfer, obtaining suitable feature representations is essential. Efros and Leung [Efros and Leung 1999] propose a non-parametric method for texture synthesis, which tries to preserve most of the local structure, and produces good results for both synthetic and real-world textures. Efros and Freemann [Efros and Freeman 2001] present a simple image-based method to change the appearance of an image by stitching together small patches of existing images. Hertzmann et al. [Hertzmann et al. 2001] describe a framework for processing images by example, named "image analogies", which transfers styles represented using a pair of non-styled and styled images to novel input images. However, the common limitation of these non-parametric methods is that they only use low-level features of images and may not be able to capture content and style effectively.

Deep Learning based Methods. With the development of both theory and hardware capability, deep learning offers a novel alternative for style transfer. As ground truth is generally unavailable for style transfer, training a model that extracts features dedicated for style transfer is challenging. As a first attempt, Gatys et al. [Gatys et al. 2016] use Gram matrices of the neural activations from different layers of a Convolutional Neural Network (CNN) to represent the artistic style of an image, and generate a new image from a white noise initialization followed by an iterative optimization process. This novel method attracted many follow-up works aimed at improving different aspects of their approach. To reduce the computational burden, Johnson et al. [Johnson et al. 2016] and Ulyanoy et al. [Ulyanov et al. 2016] train a feed-forward network to quickly approximate solutions to the optimization problem. To improve the transfer results, researchers have developed different complementary schemes, e.g. by incorporating novel spatial constraints through gain maps [Selim et al. 2016] and semantic maps [Gatys et al. 2017], and by combining deep convolutional neural networks with a Markov random field (MRF) prior [Li and Wand 2016]. To expand the field of application, Ruder et al. [Ruder et al. 2016] present an approach that transfers the style from one image to a whole video sequence. Selim et al. [Selim et al. 2016] propose an approach for head portrait painting which works for different painting styles. Some works concentrate on theoretical studies, exploring why the Gram matrices can represent artistic styles. Li et al. [Li et al. 2017] demonstrate that matching the feature maps of the style image and the generated image can be seen as minimizing the Maximum Mean Discrepancy (MMD) with the second order polynomial kernel. McCaig et al. [McCaig et al. 2016] investigate the value of such neural style transfer algorithms when carrying out creative computational research, explaining and schematizing the essential aspects of the algorithm's operation.

Among all the works, Johnson's method [Johnson et al. 2016] stands out by way of its fast speed whilst achieving results with satisfactory quality. By pre-training a feed-forward network rather than directly optimizing the loss functions as in [Gatys et al. 2016],

Johnson's method is orders of magnitude more efficient for stylizing new input images. We thus build our depth-aware style transfer based on [Johnson et al. 2016] although the idea can as well be incorporated into other image style transfer frameworks.

2.2 Single-Image Depth Perception

Traditional Methods. Image-to-depth conversion is a long-standing problem with a large body of literature. Prior traditional methods (e.g., [Liu et al. 2010; Saxena et al. 2005]) typically formulate the depth estimation as a Markov Random Field learning problem. However, it is in general intractable to learn and infer MRFs, so they usually employ some approximation methods. In addition, many methods rely on upright orientation of images and hence the flexibility is limited. Moreover, traditional methods usually use hand-crafted features (e.g., texton, GIST, SIFT), so their representational power is also limited.

Deep Learning based Methods. The recent convergence of deep neural networks and RGB-D datasets [Geiger et al. 2013; Silberman et al. 2012] has accelerated the development in this area. Liu et al. [Liu et al. 2015] utilize the continuous characteristic of depth values, and treat the depth estimation problem as a continuous conditional random field (CRF) learning problem. They propose a deep structural learning scheme, which learns potentials of continuous CRF in a unified deep CNN framework to estimate depths from a single image. Eigen and Fergus [Eigen and Fergus 2015] address three different computer vision tasks, including depth prediction, using a single multiscale convolutional network architecture. The method progressively refines predictions using a sequence of scales without the help of any superpixels or low-level segmentation. Li et al. [Li et al. 2015] tackle this problem by regression on deep CNN features, which is combined with a post-processing refining step using a CRF. Zhang et al. [Zhang et al. 2015] develop a Markov random field to provide a coherent single explanation of an image. Wang et al. [Wang et al. 2015] utilize both global prediction and local prediction, and formulate the problem in a two-layer Hierarchical Conditional Random Field (HCRF) to produce the final depth map. Although all of these methods can produce good results, it is noteworthy that the networks in these works are all trained on ground-truth metric depth.

In practice however, research shows that humans are better at judging relative depth [Todd and Norman 2003]. Zoran et al. [Zoran et al. 2015] propose a framework that infers mid-level visual properties of an image by learning about ordinal relationships. This work shows that it is feasible to estimate metric depth using only annotations of relative depth. Inspired by the previous work, Chen et al. [Chen et al. 2016] propose a new algorithm that learns to estimate metric depth using annotations of relative depth. The algorithm uses an "hourglass" network, which has been used to achieve state-of-the-art results on human pose estimation, and the training data are RGB images with relative depth annotations. They demonstrate that this algorithm significantly improves single-image depth perception in the wild.

3 METHOD

As shown in Figure 2, our system is composed of three main parts: an image transformation network f_W, and two loss networks ϕ_0 and ϕ_1. The two loss networks are used to define three loss functions: l_1, l_2 and l_3, where l_1 and l_2 are based on ϕ_0, and correspond to the style loss and content loss, also denoted as $l^{\phi_0}_{style}$ and $l^{\phi_0}_{content}$ respectively. l_3 is the depth loss $l^{\phi_1}_{depth}$, based on ϕ_1. The image transformation network is a deep residual convolutional neural network parameterized by weights W; it transforms an input image x into an output image $\hat{y}$ via the mapping $\hat{y} = f_W(x)$. Each loss function computes a scalar value $l_i(\hat{y}, y_i)$ measuring the difference between the output image $\hat{y}$ and a target image y_i ($i = 1, 2, 3$ corresponding to content, style and depth images).

The image transformation network is trained using stochastic gradient descent to minimize a weighted combination of loss functions:

$$W^* = \arg\min_W \mathbf{E}_{x,\{y_i\}}[\sum_{i=1}^{3} \lambda_i l_i(f_W(x), y_i)] \quad (1)$$

The three loss functions fall into two categories: perceptual loss function ($l^{\phi_0}_{style}$ and $l^{\phi_0}_{content}$) and per-pixel loss function ($l^{\phi_1}_{depth}$). Perceptual loss functions, based on high-level features extracted from pre-trained networks, are used to measure high-level perceptual and semantic differences between images. Compared with per-pixel losses, perceptual losses measure image similarities more robustly. This works because according to some recent work (e.g., [Mahendran and Vedaldi 2015; Simonyan et al. 2013]), the convolutional neural networks pre-trained for image classification have already learned to encode the perceptual and semantic information. In contrast, per-pixel loss is more suitable when we have a ground-truth target that the network is expected to match. This is suitable for the depth loss, as relative depth can be estimated from the content and synthesized images. In our method, ϕ_0 is a pre-trained image classification network, and ϕ_1 is a single-image depth perception network [Chen et al. 2016].

In the training phase, we pass each input image x through the image transform network f_W and obtain synthesized image $\hat{y}$. To measure the total loss, the input image x also serves as the content target y_c. The user supplied style image is treated as the style target y_s. The style reconstruction loss $l^{\phi_0}_{style}$ is produced by comparing each $\hat{y}$ with y_s in the loss network ϕ_0, and the content reconstruction loss $l^{\phi_0}_{content}$ is produced by comparing each $\hat{y}$ with y_c in the same loss network ϕ_0. The depth reconstruction loss $l^{\phi_1}_{depth}$ is produced by an additional depth prediction network ϕ_1 through comparing the output of $\hat{y}$ and y_c in ϕ_1, with the aim of making the stylized image retain a depth output consistent with the content.

3.1 Image Transformation Networks

Inspired by the architectural guidelines set forth by [Radford et al. 2015], we replace the pooling layers of the image transformation networks with strided and fractionally strided convolutions, which achieve the same goal of sampling. The network body consists of five residual blocks [He et al. 2016]. All non-residual convolutional layers are followed by spatial batch normalization [Ioffe and Szegedy 2015] and ReLU nonlinearities with the exception of the output layer, which instead uses a scaled tanh to range the output pixels from 0 to 255. Generally, each layer in the network is equivalent to a non-linear filter bank. With the increase of the

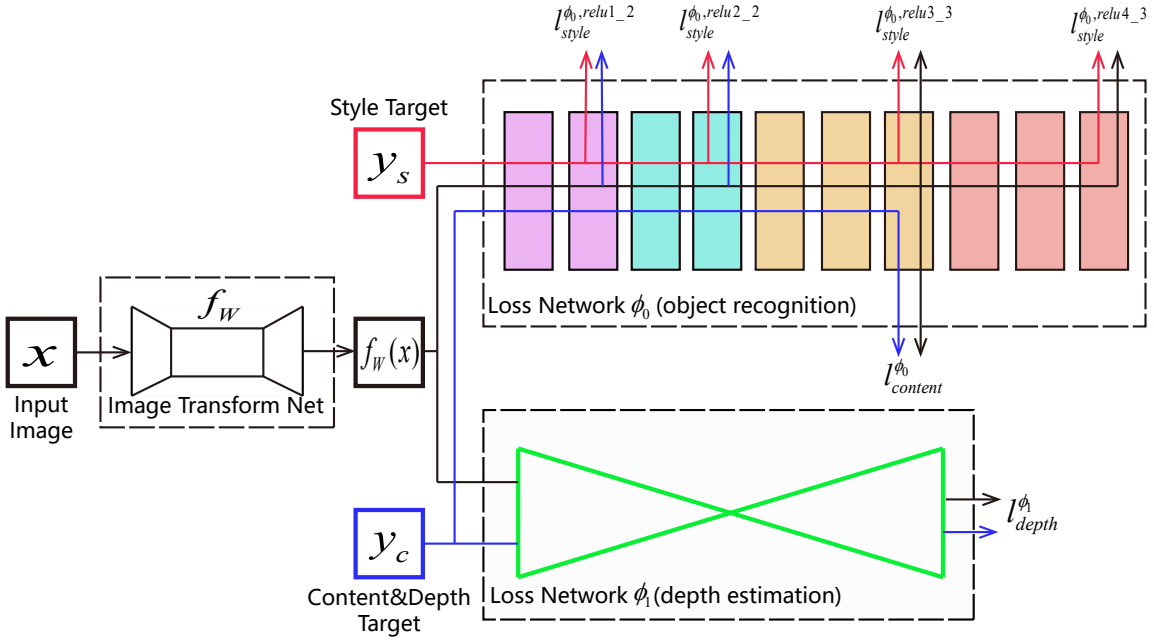

Figure 2: System Overview. We train an image transformation network to transform the input images. We use a loss network pre-trained for object recognition to define style and content loss, and an additional depth estimation network to define depth loss. In the training stage, for a specific style, we obtain the corresponding style transfer model through optimizing the total loss.

layer's position, the complexity of the filter bank increases. Hence the input image x is encoded in each layer of the network by the filter responses to that image.

Inputs and Outputs. In the training phase, the input and output are both color images of size 256×256 with 3 color channels. Since the image transformation networks are fully-convolutional, there is no limit to the size of test images.

Downsampling and Upsampling. We first use two downsampling layers and then two upsampling layers, each of stride 2, to process the input. Between the sampling layers are several residual blocks. After these processing steps, the size of the image is preserved, but this procedure comes with two advantages: On the one hand, after downsampling, we can use a larger network for the same computational cost. For instance, the computational cost of a 3×3 convolution with C filters on an input of size $H \times W \times C$ is equal to a 3×3 convolution with DC filters on an input of shape $\frac{H}{D} \times \frac{W}{D} \times DC$, where D is the downsampling factor. On the other hand, downsampling gives a larger effective receptive fields with the same number of layers. For instance, without downsampling, each additional 3×3 convolutional layer increases the effective receptive field size by 2. After downsampling by a factor of D, the effective receptive field size increases to $2D$. In general, the larger the receptive fields, the better the style transfer results are.

Residual Connections. He et al. [He et al. 2016] point out that residual connections make it easy for the network to learn the identity function. It can be observed that when performing style transfer, in many cases, the output image should share structure with the input image, so we include several residual blocks in our network to enhance this ability.

3.2 Depth Loss Function

The depth loss function is used to measure the depth differences between the transformed image $\hat{y}$ and the content target image y_c. In order to preserve maximum depth information and potential structural features, we take the outputs of the depth estimation network and compute the distances as the depth loss.

Let x and $\hat{y}$ be the original image and the transformed image, $\phi_1(x)$ and $\phi_1(\hat{y})$ are their respective depth estimation with shape $H \times W$. The depth loss function is the (squared, normalized) Euclidean distance between feature representations:

$$l^{\phi_1}_{depth}(\hat{y}, x) = \frac{1}{C \times H \times W} \|\phi_1(\hat{y}) - \phi_1(x)\|_2^2 \tag{2}$$

Let ϕ_1^{ijk} be the activation of the i^{th} filter at position (j, k) of the outputs. The derivative of this loss with respect to the outputs is:

$$\frac{\partial l^{\phi_1}_{depth}}{\partial \phi_1^{ijk}(\hat{y})} = \begin{cases} \frac{2(\phi_1^{ijk}(\hat{y}) - \phi_1^{ijk}(x))}{C \times H \times W} & \text{if } \phi_1^{ijk}(\hat{y}) > 0 \\ 0 & \text{if } \phi_1^{ijk}(\hat{y}) < 0 \end{cases} \tag{3}$$

Figure 3: Qualitative result comparison with [Johnson et al. 2016]. We can see that the style of the original method tends to be distributed fairly evenly, which thus obscures the image layout. In contrast, our image stylization method better retains the structure of the content image. Content images are from Pixabay. Style1: The Starry Night by Vincent van Gogh. Style2: The Great Wave by Hokusai.

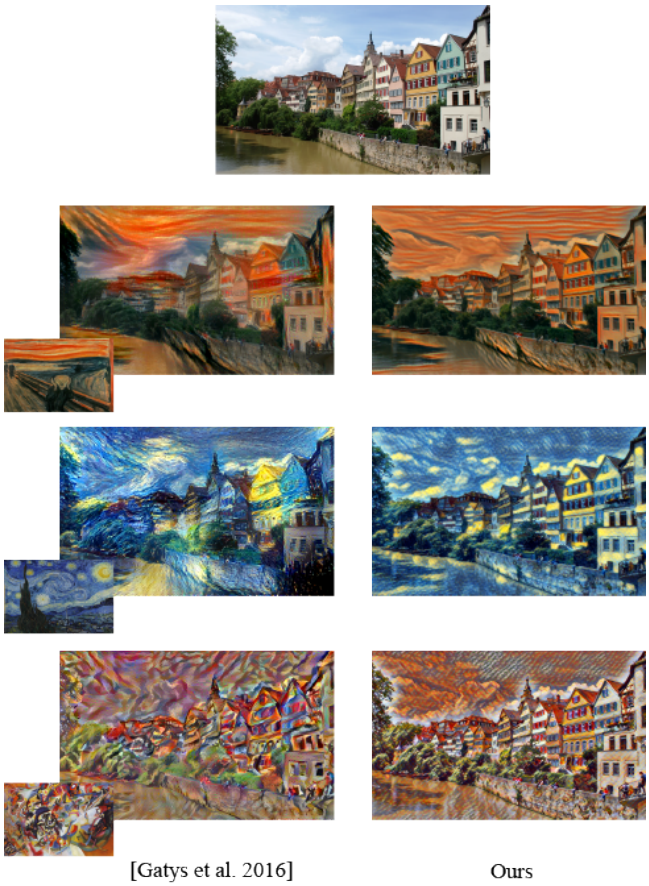

Figure 4: Qualitative result comparison with [Gatys et al. 2016]. We can see that the content of style images also appears in the stylized results of [Gatys et al. 2016], and the original content is messed up. Our results apply the styles of the style images without inserting their content, and the original content is well preserved. Content image is from Pixabay. Style images (from top to down): The Scream by Edvard Munch; The Starry Night by Vincent van Gogh; Composition by Wassily Kandinsky.

3.3 Content Loss Function

Similar with [Johnson et al. 2016], the content loss is defined as the (squared, normalized) Euclidean distance of high-level features in the specific layer of image classification network ϕ_0. Let ϕ_0^j be the activations of the j^{th} layer of ϕ_0 when processing the image x with shape $H_j \times W_j \times C_j$. The content loss function is:

$$l_{content}^{\phi_0}(\hat{y}, x) = \frac{1}{C_j \times H_j \times W_j} \|\phi_0^j(\hat{y}) - \phi_0^j(x)\|_2^2 \qquad (4)$$

The derivative of this loss with respect to the outputs has the same form as Eq. 3.

The selected layer j is chosen from early layers, because doing so tends to ensure the transferred image $\hat{y}$ is visually indistinguishable from x. Unlike the depth loss function, we use perceptual loss here, because as mentioned in [Todd and Norman 2003], compared with per-pixel differences of the feed-forward outputs, perceptual losses are more robust and stable.

3.4 Style Loss Function

The style loss is to help ensure the output image $\hat{y}$ reproduces the style target y_s. We thus wish to penalize differences in style: colors, textures, common patterns, etc. To achieve this effect, we select a set of layers and compute the sum of individual losses as the final style reconstruction loss, similar to [Gatys et al. 2016; Johnson et al. 2016].

Figure 5: Comparison of depth maps. The style is the same as style 2 in Figure 3. The first row contains the content image and the corresponding depth map using [Chen et al. 2016], the second row shows results using [Johnson et al. 2016] and the depth map using [Chen et al. 2016], the last row is our method and the depth map. Content image is from Pixabay.

As above, let ϕ_0^j be the filter responses with a shape of $H_j \times W_j \times C_j$ at the j^{th} layer of the network ϕ_0 for the input x. The Gram matrix $G_j^{\phi_0}$ is defined as the inner product of every two filter responses. So it is a symmetric matrix of $C_j \times C_j$ whose elements are given by

$$G_j^{\phi_0}(x)_{c,c'} = \frac{1}{C_j \times H_j \times W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi_0^j(x)_{h,w,c} \phi_0^j(x)_{h,w,c'} \quad (5)$$

The contribution of layer j to the total style loss is then defined as the squared Frobenius norm of the difference between the Gram matrices of the output and target images:

$$l_{style}^{\phi_0,j}(\hat{y}, y_s) = \|G_j^{\phi_0}(\hat{y}) - G_j^{\phi_0}(\hat{y_s})\|_F^2 \quad (6)$$

Assuming J stands for the set of selected layers, the total style loss is defined as:

$$l_{style}^{\phi_0}(\hat{y}, y_s) = \sum_{j \in J} l_{style}^{\phi_1,j}(\hat{y}, y_s) \quad (7)$$

More details about specific selected layers J can be found in Section 4.1.

It is noteworthy that the form of $l_{style}^{\phi_0}$ is different from $l_{depth}^{\phi_1}$ and $l_{content}^{\phi_0}$. The form of style representation is different. Also, we select a set of layers instead of one. There are two reasons why we use this representation. First, due to the characteristics of the loss network ϕ_0, the responses of its intermediate layers cannot be directly used to represent the style of an image. Instead, the style of an image can be intrinsically represented by feature correlations in different layers of a CNN, so we calculate the distance between Gram matrices to measure the style similarity. Secondly, by including feature correlations of multiple layers, we obtain a multi-scale representation of the style. So, by finding an image $\hat{y}$ that minimizes the style reconstruction loss for multiple layers, we tend to preserve the stylistic features but do not preserve the spatial structure.

4 EXPERIMENTS

In this section, we provide the training details and perform experiments on some pictures with several styles. Compared to [Johnson et al. 2016], our results are more effective: providing stylization whilst preserving scene structure. Unfortunately, performing evaluation of NPR algorithms is well known to be a difficult problem due to the wide variety of NPR styles, the lack of ground truth and objective evaluation measures [Hertzmann 2010; Isenberg 2013]. One solution is to use a proxy measure to gain at least an indication of the quality of an algorithm [Hertzmann 2010]. We take this approach, and carry out comparative experiments which consider the depth, edge and salience of the stylized images. Although they do not provide a direct measurement for the quality of results, they reflect in part the capability of the method for maintaining fundamental features of images.

4.1 Training Details

We choose Microsoft COCO [Lin et al. 2014], including $80k$ images, as the training dataset. All the training images are resized to 256×256 and then trained with a batch size of 4 for 40,000 iterations. On the choice of optimization method, we use Adam [Kingma and Ba 2014] with a learning rate of 1×10^{-3}, because this method is straightforward, is computationally efficient, has few requirements and is well suited for problems that involve large amounts of data. Based on the cross-validation per style target, the output images are regularized with a total variation regulation strength ranging from 1×10^{-6} to 1×10^{-4}. Weight decay and dropout are not used in our model, as the model does not overfit within two epochs. We compute feature reconstruction loss at layer $relu2\_2$ and style reconstruction loss at layers $relu1\_2$, $relu2\_2$, $relu3\_3$, and $relu4\_3$ of the VGG-16 loss network. The depth reconstruction loss is computed at the output layer of the model in [Chen et al. 2016]. The weights of the three losses are 1 (content), 5 (style) and 5 (depth), respectively. Our implementation uses Torch [Collobert et al. 2011] and cuDNN [Chetlur et al. 2014]; the training process takes approximately 4 hours on a single GTX Titan X GPU.

4.2 Qualitative Results

In Figure 3, we show qualitative examples comparing our results with the method of [Johnson et al. 2016] for two style and content images. Except for the extra depth reconstruction loss, in all cases

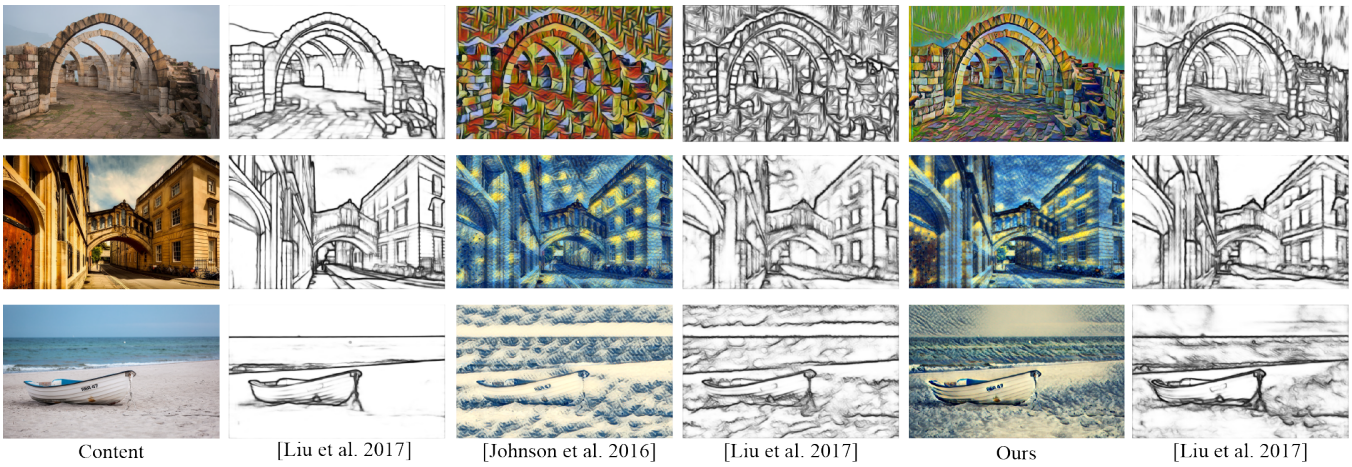

Figure 6: Example results of edge detection using [Liu et al. 2017]. The three style images are respectively the same as the styles in Figure 1 and Figure 3. The first two columns are the content and the edge detection results using [Liu et al. 2017]. The middle two columns are the results of [Johnson et al. 2016] and the edge detection results using [Liu et al. 2017]. The last two columns are our results. It can be seen that the results of [Johnson et al. 2016] introduce extensive spurious edges, which spread over the whole image. Content images are from Pixabay.

the hyperparameters are exactly the same between the two methods. We see that for pictures containing a rich 3D spatial layout with obvious distance relationships, our method can better retain the content's general layout. In addition, in the original method, the style features are evenly distributed so that it is difficult to tell apart the foreground and background elements of the scene. In contrast, while our results show a variety in the styles within an image, structures with continuous change in depth tend to have the same style, i.e. they are rendered in a consistent manner. Moreover, our results can effectively maintain the basic properties of the original images. For more comparative results, see Figure 8.

We also compare our method with [Gatys et al. 2016] (see Figure 4). Our method avoids the typical artifacts of [Gatys et al. 2016] which tends to insert some content of the style images into the synthesized images, and produces stylized images with well preserved image layout. For instance, like the second row of Figure 4, when rendering a photograph of the Thbingen in the style of the painting *The Scream*, in the result of [Gatys et al. 2016] we can see the barrier, which belongs to the content of *The Scream*. However, our approach has a few disadvantages, such as the deficiency of expressing abstract lines.

4.3 Depth Map Comparison

The depth map is an important characteristic of an image, since it contains 3D feature information about the objects. In our view, the purpose of style transfer is to change the image style whilst retaining other fundamental characteristics as much as possible. For certain types of images in particular, depth is critical to the perception. When rendering these images, a good result should not make a huge change to the depth map.

In Figure 5, we compare the depth maps of the original image and the depth maps produced from the stylized results of two methods. The results indicate that we recover the overall subjective depth structure of the scene quite well, often with crisp edges at depth discontinuities.

It is not surprising that our results preserve the depth well, because we minimize the depth loss in the training stage. Although not totally the same, our results well preserve the relative relationship between positions, and it is enough to make people aware of the spatial distribution and positional relation. Moreover, this extra loss not only substantially alters the transferred results, but also leads to changes in other basic properties.

4.4 Edge Comparison

When transferring the style of an image, what we want to change is the brushwork and color scheme, and these changes have little effect on edges, which are usually the boundaries and outlines of the major objects. So whatever the style is, the major objects' boundaries and shapes should not change largely. Therefore, measuring the preservation of detected edges provides an indication of the effectiveness of the style transfer. A good result should maintain the original edges well without introducing other clutter edges. We choose the recent richer convolutional features edge detector (RCF) [Liu et al. 2017] as the edge detection method. As a CNN-based method, HED has a distinct advantage over the traditional methods in that it can better capture semantic boundaries and produces fewer responses to purely low-level features (e.g., gradient, contrast). By using CNN, RCF tends to respond significantly to semantic boundaries. We first select images to apply style transform to, and then use RCF to detect their edges.

As shown in Figure 6, in the result of [Johnson et al. 2016], the stylistic elements are scattered over the image. So it is inevitable that this has introduced unnecessary edges. In the worst case, as in the first row, we cannot differentiate between the foreground and background, the overall layout was disrupted and the content was obscured. To a lesser degree, we see these phenomena in the other

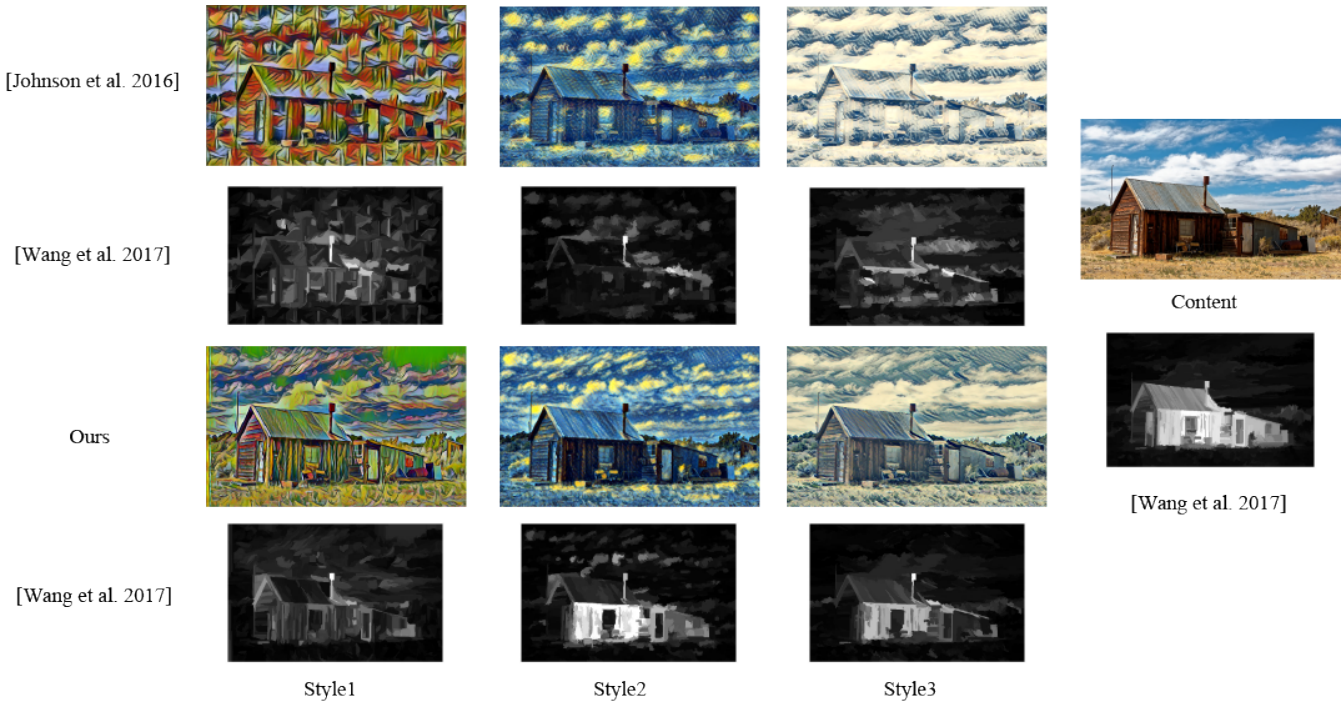

Figure 7: Example results of saliency detection using [Wang et al. 2017]. The three style images are respectively the same as the styles in Figure 1 and Figure 3. The last column are the content image and the groundtruth. The first two lines are the transferred results of [Johnson et al. 2016] and the saliency detection results using [Wang et al. 2017]. The last two lines are the results of our method and the corresponding saliency map. It can be seen that the saliency maps of [Johnson et al. 2016] are out of accordance with the groundtruth, while our method preserves largely the groundtruth. Content image is from Pixabay.

two cases as well. In contrast, our method provides a strong sense of object and depth layers, is artistically more attraction, and very dramatic.

4.5 Saliency Comparison

In computer vision, a saliency map is useful as it points out the visually dominant locations. During the style transformation, a good result should not cause large changes in the saliency map under the premise of retaining the original content. After stylization, it is acceptable to weaken or enhance the original saliency map, but its integrity should be retained. We can still identify the content from the new saliency map. So we apply saliency detection as a supplementary evaluation method. On the choice of evaluation method, we choose discriminative regional feature integration method (DRFI) [Wang et al. 2017] as the detection method. DRFI is based on performing multi-level image segmentation. It maps the regional feature vector to a saliency score, and finally fuses the saliency scores to generate the saliency map. Prior to deep-learning methods, DRFI is the best one among all the traditional methods.

As shown in Figure 7, the saliency detection result shows that our method not only substantially enhances spatial detail information of the result image, but also effectively preserves the saliency of the original image. In the saliency maps of [Johnson et al. 2016], the salient parts appear as tiny spots and the whole map loses the sense of coherence, so that we cannot extract saliency object from it. In contrast, our results make it easy for an observer to determine the objects. The root cause of these can be explained by the correlation between the depth map and saliency map. Generally speaking, depth information can affect identification of visually salient regions in images. This phenomenon was studied in [Lang et al. 2012], who concluded that humans fixate preferentially at closer depth ranges and determined that the relation between depth and saliency is non-linear. Our results also demonstrate that there is an association between depth and saliency, since by strengthening depth in the training phase, saliency was also enhanced.

4.6 Trade-off between perceptual loss and depth loss

When synthesizing an image optimizing a combination of the perceptual loss and depth loss, an image that perfectly matches both constraints at the same time does not usually exist. However, since the loss function we minimize is a linear combination of the perceptual and depth loss functions, we can freely adjust the relative weighting between the two factors. A strong emphasis on depth will result in images that match the spatial distribution of the content. When placing strong emphasis on perceptual loss, one can better capture the image style, but the overall layout tends to be less well preserved.

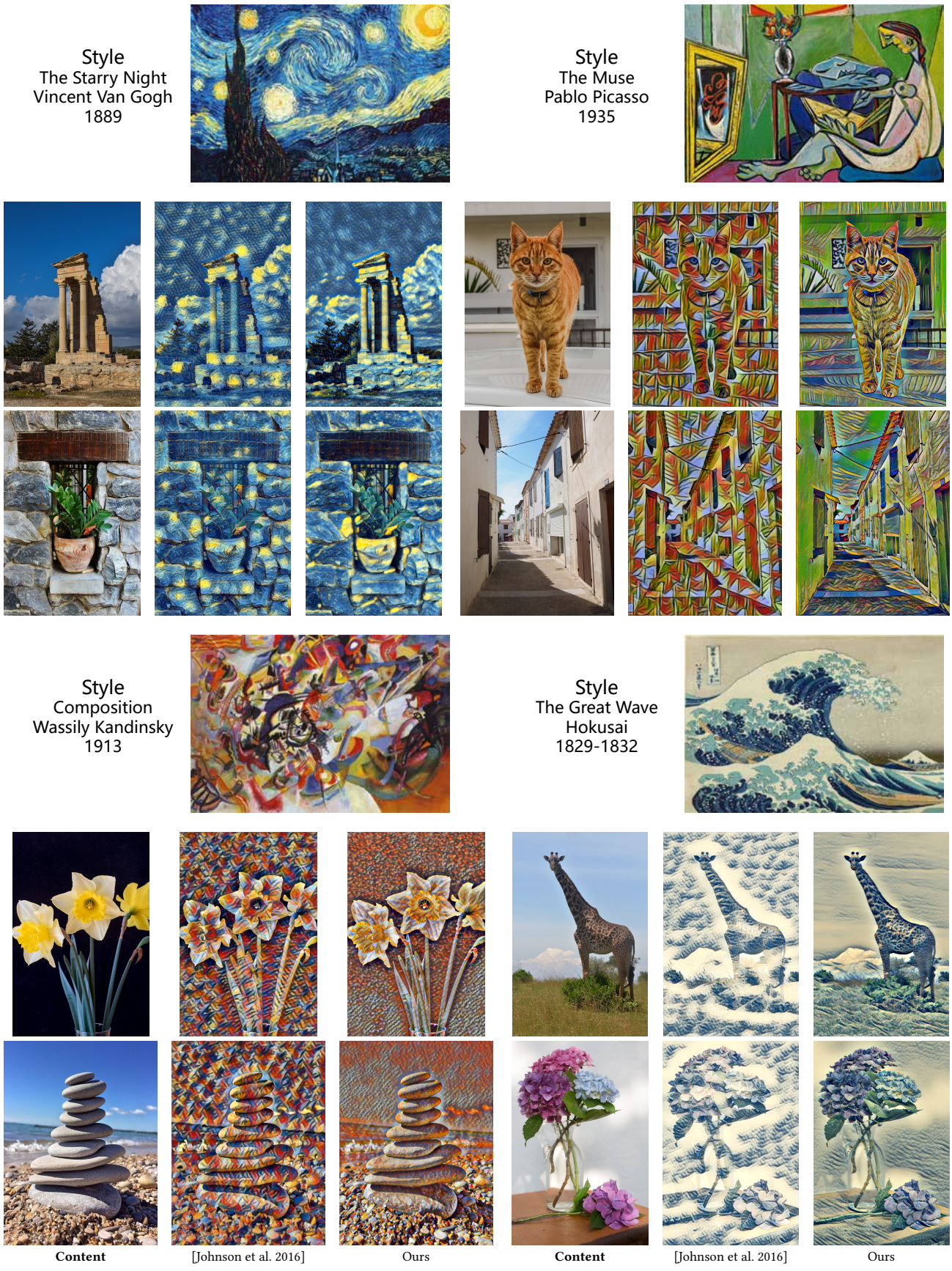

Figure 8: Example results of using our method. Compared with [Johnson et al. 2016], our results better capture major structures of the image, and have crisper outlines. Content images are from Pixabay.

5 CONCLUSION

In this paper we have combined the benefits of feed-forward image transformation methods and optimization-based methods for image generation by training feed-forward transformation networks with extra depth loss functions. Compared to existing methods, our method is advantageous. For a wide range of images, we achieve aesthetically appealing results which better preserve the semantic content and layout of the content images.

At the same time, we perform evaluation to compare the results of different methods. As we have stated before, changing the absolute value of depth is acceptable, but relative depths should be retained in order to retain distinct rendering of objects as well as foreground and background, especially for images that cover a large range of depth. Finally, style transfer should preserve the coherence and spatial layout of the original content image, and we evaluate this by checking the amount of change in the saliency map caused by style transfer.

In the future, we will investigate incorporating and combining other information such as intrinsic images (e.g. shading, albedo), which can also be extracted by CNNs, to improve style transfer.

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their useful feedback. All the content images used in this paper were taken from the website Pixabay. This research was supported by NSFC (No. 61572264, 61620106008), and YESS Program by CAST.

REFERENCES

Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. 2016. Single-image depth perception in the wild. In *NIPS*. 730–738.

Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. 2014. BING: Binarized Normed Gradients for Objectness Estimation at 300fps. In *CVPR*.

Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer. 2014. cudnn: Efficient primitives for deep learning. *arXiv preprint arXiv:1410.0759* (2014).

Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. 2011. Torch7: A Matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*.

Alexei A Efros and William T Freeman. 2001. Image quilting for texture synthesis and transfer. In *ACM SIGGRAPH*. 341–346.

Alexei A Efros and Thomas K Leung. 1999. Texture synthesis by non-parametric sampling. In *ICCV*, Vol. 2. 1033–1038.

David Eigen and Rob Fergus. 2015. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*. 2650–2658.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. In *CVPR*. 2414–2423.

Leon A Gatys, Alexander S Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. 2017. Controlling Perceptual Factors in Neural Style Transfer. In *CVPR*.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. 2013. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research* 32, 11 (2013), 1231–1237.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*. 580–587.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

Aaron Hertzmann. 2010. Non-Photorealistic Rendering and the Science of Art. In *NPAR*. 147–157.

Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. 2001. Image analogies. In *ACM SIGGRAPH*. 327–340.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

Tobias Isenberg. 2013. Evaluating and Validating Non-photorealistic and Illustrative Rendering. In *Image and Video-Based Artistic Stylisation*, Paul L. Rosin and John P. Collomosse (Eds.). Springer, 311–331.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*. Springer, 694–711.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Congyan Lang, Tam V Nguyen, Harish Katti, Karthik Yadati, Mohan Kankanhalli, and Shuicheng Yan. 2012. Depth matters: Influence of depth cues on visual saliency. In *ECCV*. 101–115.

Bo Li, Chunhua Shen, Yuchao Dai, Anton van den Hengel, and Mingyi He. 2015. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs. In *CVPR*. 1119–1127.

Chuan Li and Michael Wand. 2016. Combining Markov Random Fields and convolutional neural networks for image synthesis. In *CVPR*. 2479–2486.

Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. 2017. Demystifying Neural Style Transfer. *CoRR* abs/1701.01036 (2017).

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *ECCV*. 740–755.

Beyang Liu, Stephen Gould, and Daphne Koller. 2010. Single image depth estimation from predicted semantic labels. In *CVPR*. 1253–1260.

Fayao Liu, Chunhua Shen, and Guosheng Lin. 2015. Deep convolutional neural fields for depth estimation from a single image. In *CVPR*. 5162–5170.

Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Kai Wang, and Xiang Bai. 2017. Richer Convolutional Features for Edge Detection. In *CVPR*.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*. 3431–3440.

Aravindh Mahendran and Andrea Vedaldi. 2015. Understanding deep image representations by inverting them. In *CVPR*. 5188–5196.

Graeme McCaig, Steve DiPaola, and Liane Gabora. 2016. Deep Convolutional Networks as Models of Generalization and Blending Within Visual Creativity. *CoRR* abs/1610.02478 (2016).

Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NIPS*.

Paul L. Rosin and J. Collomosse. 2013. *Image and Video-based Artistic Stylisation*. Springer.

Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. 2016. Artistic style transfer for videos. In *GCPR*. 26–36.

Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. 2005. Learning depth from single monocular images. In *NIPS*, Vol. 18. 1–8.

Ahmed Selim, Mohamed Elgharib, and Linda Doyle. 2016. Painting style transfer for head portraits using convolutional neural networks. *ACM TOG* 35, 4 (2016), 129.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. 2012. Indoor segmentation and support inference from RGBD images. In *ECCV*. 746–760.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv:1312.6034* (2013).

Thomas Strothotte and Stefan Schlechtweg. 2002. *Non-photorealistic computer graphics: modeling, rendering, and animation*. Morgan Kaufmann.

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

Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. 2016. Texture networks: Feed-forward synthesis of textures and stylized images.

Jingdong Wang, Huaizu Jiang, Zejian Yuan, Ming-Ming Cheng, Xiaowei Hu, and Nanning Zheng. 2017. Salient Object Detection: A Discriminative Regional Feature Integration Approach. *IJCV* 123, 2 (2017), 251–268.

Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. 2015. Towards unified depth and semantic prediction from a single image. In *CVPR*. 2800–2809.

Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. 2017. Object Region Mining with Adversarial Erasing: A Simple Classification to Semantic Segmentation Approach. In *CVPR*.

Yunchao Wei, Xiaodan Liang, Yunpeng Chen, Xiaohui Shen, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. 2016. STC: A Simple to Complex Framework for Weakly-supervised Semantic Segmentation. (2016). DOI:https://doi.org/10.1109/TPAMI.2016.2636150

Ziyu Zhang, Alexander G Schwing, Sanja Fidler, and Raquel Urtasun. 2015. Monocular object instance segmentation and depth ordering with CNNs. In *ICCV*. 2614–2622.

Daniel Zoran, Phillip Isola, Dilip Krishnan, and William T Freeman. 2015. Learning ordinal relationships for mid-level vision. In *ICCV*. 388–396.