algorithms

MDPI

Article

Transform a Simple Sketch to a Chinese Painting by a Multiscale Deep Neural Network

Daoyu Lin [1,2], Yang Wang [2], Guangluan Xu [2,*], Jun Li [1,2] and Kun Fu [1,2]

1 Department of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100190, China; lindaoyu15@mails.ucas.ac.cn (D.L.); lj120118@163.com (J.L.); fukun@mail.ie.ac.cn (K.F.)

2 The Key Laboratory of Technology in Geo-Spatial Information Processing and Application System, Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China; primular@163.com

* Correspondence: gluanxu@mail.ie.ac.cn; Tel.: +86-136-8107-8907

Received: 30 October 2017; Accepted: 8 January 2018; Published: 11 January 2018

Abstract: Recently, inspired by the power of deep learning, convolution neural networks can produce fantastic images at the pixel level. However, a significant limiting factor for previous approaches is that they focus on some simple datasets such as faces and bedrooms. In this paper, we propose a multiscale deep neural network to transform sketches into Chinese paintings. To synthesize more realistic imagery, we train the generative network by using both L1 loss and adversarial loss. Additionally, users can control the process of the synthesis since the generative network is feed-forward. This network can also be treated as neural style transfer by adding an edge detector. Furthermore, additional experiments on image colorization and image super-resolution demonstrate the universality of our proposed approach.

Keywords: deep neural network; sketch; arts synthesis; style transfer

1. Introduction

Chinese painting is one of the oldest artistic traditions in human history; Zhang Zeduan, one of the Song Dynasty artists, was amazed by the tremendous achievements of Along the River during the Qingming Festival (Figure 1a). It is hard for ordinary people to draw such a long painting without continuous training. However, through the neural network, we only need to draw out pure sketches that can produce impressive artworks.

An increasing number of researchers has already paid much attention to the synthesis of magnificent artworks, and it is a significant problem in computer graphics and vision [1–4]. Image synthesis [5,6] is a process of creating new images from some forms of image description, which can be sketches only or require users to draw the sketches according to the text labels. Recently, inspired by the power of Convolutional Neural Networks (CNN) [7,8], generative models based on deep neural networks can produce incredibly realistic images [9]. The gap between the image produced by the CNN and the image we want to produce is called the loss function. We train the CNN by minimizing the loss function so that the image it produces is similar to the real artwork. When we choose Euclidean distance as the loss function (as the Euclidean distance is to minimize the average of all pixels), the output of the CNN may be blurred. To solve this problem, the perceptual loss functions [10] were proposed, which are not based on differences between pixels, but between high-level image feature representations extracted from pretrained CNN. By using perceptual loss, the generative model can produce more high-quality images.

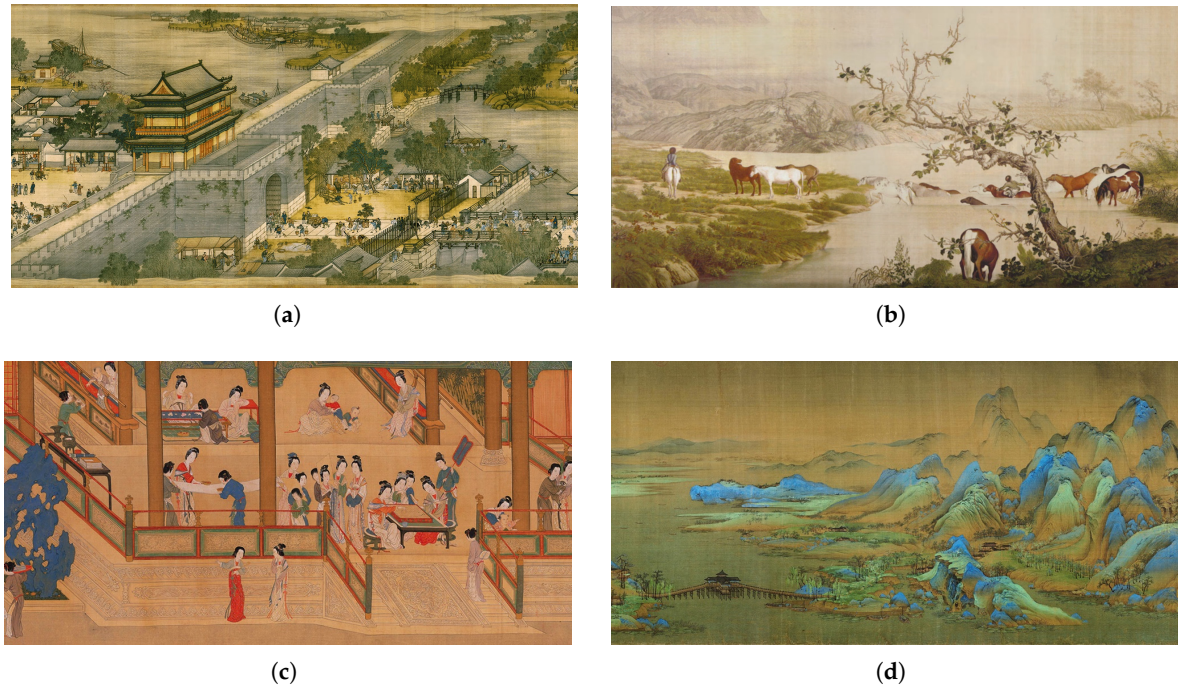


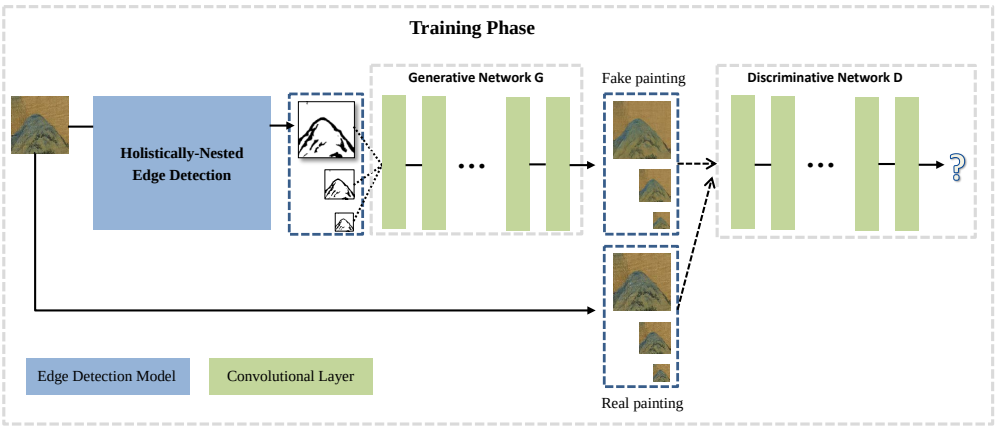
Figure 1. (**a**) Part of Along the River during the Qingming Festival (original size: 38,414 × 1800); (**b**) part of One Hundred Horses (original size: 16,560 × 2022); (**c**) part of Spring Morning in the Han Palace (original size: 102,955 × 5228); (**d**) part of A Thousand Li of Rivers and Mountains (original size: 150,000 × 6004).

In this paper, we present a deep Generative Adversarial Network (GAN) [11] by inputting simple sketches to synthesize surprising art paintings. By using both L1 loss and adversarial loss, the result is more realistic and closer to the human sensory vision. Unlike other works that generate images by sketches, they are usually generating single and straightforward scale images such as faces or bedrooms [12,13]. To produce multi-scale Chinese paintings, we propose a multiscale generative model that consists of a full convolution layer [14], and this generative model can create artworks that have not only local detail information, but also the global framework of the painting. The overall model is as Figure 2 shows.

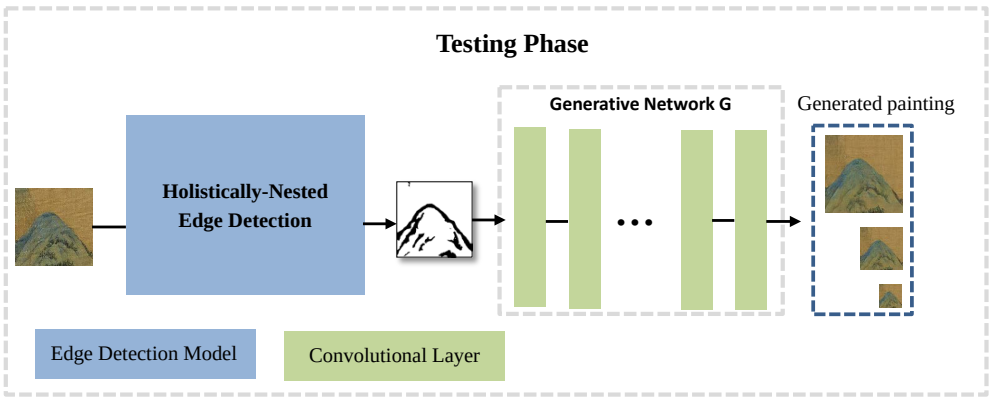
The contributions of this paper include:

- We propose a deep generative adversarial network to produce surprising Chinese paintings by inputting simple sketches.
- It can use multiscale images to train the generative model and the discriminative model by setting these two models as fully-convolution networks.
- By adding an edge detector, the generative model can also be treated as a neural style transfer method.
- The method we proposed is also effective in other image-to-image translation problems, such as image colorization and image super-resolution.

The remaining content is organized as follows. We provide a brief review of related work in Section 2. We present the methods and the architecture of the proposed network in Section 3. The training and testing process, as well as the experiment results, are shown in Section 4. In Section 5, we demonstrate the universality of our proposed approach. A summary containing a discussion and further thoughts is presented in Section 6.



(**a**)



(**b**)

Figure 2. Overview of the proposed approach. Training a multiscale deep neural network that can generate a Chinese painting from a sketch. We use an edge detection model to produce the input sketch. (**a**) By training, the discriminative network can make a distinction between the real and fake paintings, and the generative network aims at fooling the discriminative network by training; (**b**) after training, the generator can create realistic paintings.

2. Related Work

Learning from synthetic images from image datasets has always been a research interest in computer graphics and computer vision. Previously, the most successful method was often a nonparametric approach [5,15–18], which reuses existing image fragments of synthetic images. Over the past few years, parametric models [19–23] based on deep CNN have yielded promising results. Those image synthesis methods cannot create realistic high-resolution images, but they have implicit generalization capabilities, which are difficult for data-driven nonparametric methods.

A common approach to image synthesis is to learn a low-dimensional potential representation that can be used later to reconstruct an image, like a Variational Autoencoder (VAE) [19,20] or a GAN [21]. In general, image synthesis in this way can be conditioned on image attributes [24], grayscale images [25] and low-resolution input images.

2.1. Convolutional Networks

Convolutional networks [8], also known as convolutional neural networks or CNNs, are a specialized kind of neural network for processing images. Convolutional networks were inspired by biological processes in which the connectivity pattern between neurons is inspired by the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of

the visual field known as the receptive-field. The receptive-fields of different neurons partially overlap such that they cover the entire visual field. Convolutional networks can learn high-level semantic features automatically rather than requiring handcrafted features, which have been tremendously successful in practical applications such as image classification, object location and some low-level vision jobs. From LeNet [8] to AlexNet [7] and GoogleNet [26], learning deeper convolutional neural networks has become a tendency in recent years.

2.2. Generative Adversarial Nets

Generative adversarial networks [21] were first introduced in 2014. The goal of the adversarial approach is to learn deep generative models. The generator network directly produces samples $x = G(z; \theta_g)$. However, the discriminator network endeavors to make a distinction between samples drawn from the training data and samples drawn from the generator. The discriminator produces a probability value given by $D(x; \theta_g)$, demonstrating the probability that x is a real training example instead of a fake sample drawn from the model. In follow-up work, a Deep Convolutional GAN (DCGAN) [9] performed very well in image synthesis tasks and showed that its latent representation space captures important factors of variation. Improving GANs [27] provided a great variety of new architectural features and training procedures, including feature matching and mini batch discrimination, which can produce very clean and sharp images and learn codes that contain valuable information about these textures. InfoGAN [28] is an information-theoretic extension to the generative adversarial network that is able to learn disentangled representations in a completely unsupervised manner.

2.3. Sketch to Image and Style Transform

In the early age, image synthesis systems were usually built on sketch retrieval [29] and allow users to create novel, realistic images by using different sketch styles. Sketch2Photo [16] can compose a realistic picture from a simple freehand sketch annotated with text labels. The composed picture is generated by seamlessly stitching several photographs, which correspond to the sketch and text labels. Photosketcher [15] is an interactive system for the synthesis of novel images that use only user sketches as input. It is dedicated to image content and does not require keywords or other metadata associated with the image.

Recently, some sketch to image methods built on CNN have emerged. Sketch Inversion [13] is a deep neural network for inverting face sketches to synthesize photorealistic face images in the wild. Scribbler [12] is a deep adversarial image synthesis architecture that is conditioned on sketched boundaries and sparse color strokes to generate realistic cars, bedrooms or faces. Pix2pix [30] is a conditional adversarial network (cGAN) [31]; it learns the mapping from an input image to an output image.

The pioneering work of Gatys et al. [2,3] performed artistic style transfer, combining the content of one image with the style of another by jointly minimizing the feature reconstruction loss and a style reconstruction loss also based on features extracted from a pretrained convolutional network.

Johnson et al. [10] introduced a fast approach based on the algorithm proposed by Gatys et al. [2]. They train an equivalent feed-forward generator network for each specific style, which is three orders of magnitude faster than previous works.

3. Method

3.1. Generative Adversarial Nets

A generative adversarial model consists of two neural networks: generative model G and discriminative model D. Normally, the generative model G aims at mapping from a latent space to a particular data distribution of interest. In contrast, the discriminative model D learns to determine whether a sample is from the true data or the synthesized instances produced by the generator.

The training objective of generative model G is to increase the error rate of the discriminative model D (i.e., "fool" the discriminator by making new synthesized instances that seem to have come from the true data distribution).

To make this concept more precise, the following minimax game can be done with training adversarial networks.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)} \log D(x) + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \quad (1)$$

The discriminator $D(x)$ takes some image x as input and yields the possibility, and the image x is sampled from training data $p_{data}(x)$. The generator $G(z)$ aims at producing samples that fool D, where z follows a prior noise distribution $p_z(z)$.

3.2. Conditional Generative Adversarial Nets

In a simple generative model, there is no control on modes of the data being produced. However, it is probable to direct the data generating process by conditioning the model on extra information. With additional information being added to GANs, it can be developed into cGANs. The objective of cGANs is:

$$\mathcal{L}_{cGAN}(G,D) = \mathbb{E}_{x,y\sim p_{data}(x,y)}[\log D(x,y)] + \mathbb{E}_{y\sim p_{data}(y),z\sim p_z(z)}[\log(1 - D(G(y,z),y)]\,. \quad (2)$$

As noticed, y is a kind of auxiliary information. Just as shown in Figure 2, y means the input sketch.

3.3. Network Architecture

We designed a multiscale GAN to deal with the huge size, magnificent background and exquisite details of Chinese painting. To make the generative model and the discriminative model handle multiscale input images, they are all made up by convolutional layers.

In the generative model, to synthesize the output image whose resolution is the same as the input sketch, all convolutions are 3×3 spatial filters applied with $stride = 1$ and $pad = 1$. Using a small-sized filter can effectively reduce the number of parameters. Without changing the receptive-fields, we can make networks deeper, which can obtain a more non-linear fitting ability. Unlike the U-Net [32] structure used by Pix2pix [30] that uses several downsampling steps and then the same number of upsampling steps by using the deconvolutional layers [33], we used all stride one convolutional layer to avoid the using of deconvolutional layers, which will produce checkerboard artifacts [34] in the output images. Tables 1 and 2 show the details of the architecture including the generative model and discriminative model, respectively.

Table 1. Generative model network architectures. Conv means convolutional layer; BN means batch normalization; ReLU means rectified linear unit.

Generative Model					
Name	**Kernel Size**	**Stride**	**Pad**	**Norm**	**Activation**
Conv1	$3 \times 3 \times 3 \times 64$	1	1	-	ReLU
Conv2	$3 \times 3 \times 64 \times 64$	1	1	BN	ReLU
Conv3	$3 \times 3 \times 64 \times 64$	1	1	BN	ReLU
Conv4	$3 \times 3 \times 64 \times 64$	1	1	BN	ReLU
Conv5	$3 \times 3 \times 64 \times 64$	1	1	BN	ReLU
Conv6	$3 \times 3 \times 64 \times 64$	1	1	BN	ReLU
Conv7	$3 \times 3 \times 64 \times 64$	1	1	BN	ReLU
Conv8	$3 \times 3 \times 64 \times 64$	1	1	BN	ReLU
Conv9	$3 \times 3 \times 64 \times 64$	1	1	BN	ReLU
Conv10	$3 \times 3 \times 64 \times 3$	1	1	-	ReLU

Table 2. Discriminative model network architectures. Conv means Convolutional layer; BN means Batch Normalization; Leaky means leaky rectified linear unit.

Discriminative Model					
Name	**Kernel Size**	**Stride**	**Pad**	**Norm**	**Activation**
Conv1	$4 \times 4 \times 3 \times 64$	2	1	-	Leaky
Conv2	$4 \times 4 \times 64 \times 128$	2	1	BN	Leaky
Conv3	$4 \times 4 \times 128 \times 256$	2	1	BN	Leaky
Conv4	$4 \times 4 \times 256 \times 512$	2	1	BN	Leaky
Conv5	$4 \times 4 \times 512 \times 512$	2	1	BN	Leaky
Conv6	$4 \times 4 \times 512 \times 512$	2	1	BN	Leaky
Conv7	$4 \times 4 \times 512 \times 1$	2	1	BN	-

There are only convolutional layers in our networks. Let X be the input. The basic convolutional layers are expressed as:

$$F(X) = W_k * X + B_k \,. \tag{3}$$

In Equation (3), W_k represents the weights of the filters and B_k represents the biases, while $*$ denotes convolution operation. The Rectified Linear Unit (ReLU) is the activation function behind the convolutional layer, which is defined as:

$$f(x) = \max(0, x) \,. \tag{4}$$

where x is the input to a neuron. Combining Equations (3) and (4), a convolutional layer followed by a ReLU can be expressed as:

$$F(X) = \max(0, W_k * X + B_k) \,. \tag{5}$$

We also used Batch Normalization (BN) in our networks. BN is a normalization strategy that makes the distribution of layers' input consistent at the output of layers; it eliminates the effect of the internal covariant shift. Covariate shift would amplify permutation at the deeper layers. If it happens, the inputs of the activation function will stay in the saturated region. In that region, the gradient will be very small, and the phenomenon of the vanishing gradient would happen and stop neural network training. The BN layer can be applied to any input of layers. Our generative and discriminative model is the combination of convolutional layers, ReLU and BN.

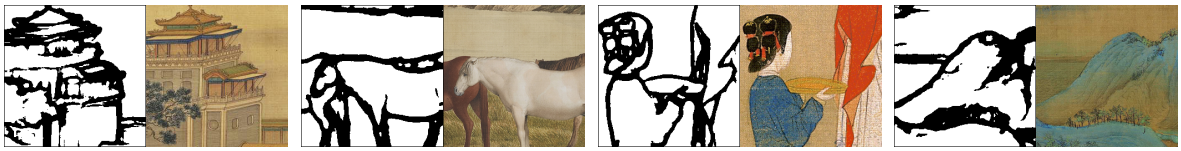
3.4. Loss Function

The training data are a pair of images composed of a sketch and corresponding paintings; as shown in Figure 3. We use the loss function to define the gap between the generated image and the real paintings, and the model is then trained by minimizing the loss function. We use both L1 loss and adversarial loss to train the generative model.

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y\sim p_{data}(x,y),z\sim p_z(z)}[\|y - G(x,z)\|_1] \tag{6}$$

L1 loss is a good way to reconstruct low-frequency information, while adversarial loss can reconstruct high-frequency information very well, so it can produce realistic images by combining L1 loss (Equation (6)) and adversarial loss (Equation (2)). Our final objective is below.

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G) \tag{7}$$



(**a**) (**b**) (**c**) (**d**)

Figure 3. Exemplary pairs from our training data, sketches (left) and paintings (right). (**a**) Along the River during the Qingming Festival; (**b**) One Hundred Horses; (**c**) Spring Morning in the Han Palace; (**d**) A Thousand Li of Rivers and Mountains.

4. Experiments and Results

We used four of the top ten paintings in ancient China as training data, including Along the River during the Qingming Festival (Figure 1a), One Hundred Horses (Figure 1b), Spring Morning in the Han Palace (Figure 1c) and A Thousand Li of Rivers and Mountains (Figure 1d). The dataset was generated by splitting images in Figure 1 into 256×256 sub-images with no overlap. To generate multiscale images, we resize the original images in Figure 1 to $1/2$ and $1/4$ of the original size by bilinear interpolation and then split into 256×256 sub-images with no overlap. Therefore, we obtained 23,520 small-level images, 5880 middle-level images and 1470 high-level images. We obtained 30,870 sub-images, and we randomly divided the training set and test set according to the ratio of 8:2. We use HED [35] to get the sketch corresponding to the painting, as shown in Figure 3. HED performs image-to-image prediction using a deep learning model that leverages fully-convolutional neural networks and deeply-supervised nets, which can automatically learn rich hierarchical representations to edge and object boundary detection. This edge detection algorithm addresses two important issues in this long-standing vision problem: (1) holistic image training and prediction; and (2) multi-scale and multi-level feature learning. Therefore, HED is suitable for extracting edges of Chinese paintings with complex textures.

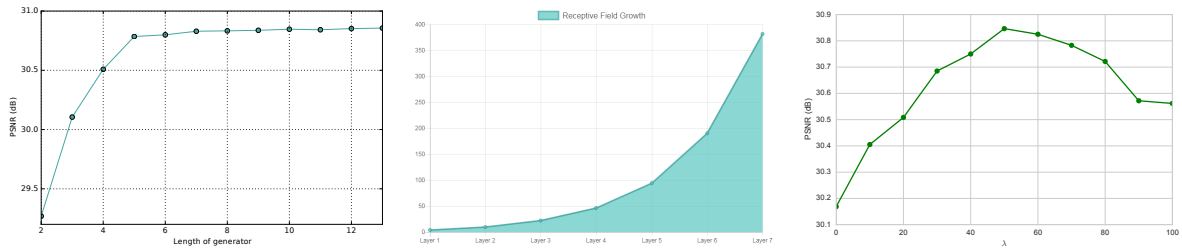
4.1. Training Details

We use Google's TensorFlow framework [36] to implement this multi-scale deep neural network, and we train the generative model and discriminative model on NVIDIA graphics cards, GTX 1080. In addition to scaling the images to the range [0,1], no other pre-processing was done to train images. We trained all models with mini-batch stochastic gradient descent, and the batch size is 1. We set the slope of the leak to 0.2 in the leaky ReLU. To accelerate training, we used the Adam optimizer with tuned hyperparameters, and the learning rate was set to 0.0002. Additionally, we set the momentum term $\beta_1 = 0.5$ to help stabilize training. We also set the BatchNormLayer [37] decay factor 0.9 for the exponential moving average. Our network can produce high quality and diverse results as shown in Figure 4.

Figure 4. Our deep network takes user sketches to synthesize Chinese paintings. The input sketch can be any size, and our deep network can synthesize the Chinese painting with fine details in real time. Despite the diverse sketch styles, our network can produce high quality and diverse results. (**Left**) Input sketch; (**right**) generated Chinese painting.

4.2. Network Architecture Analysis

To investigate the tradeoffs between the depth of network and PSNR, we change the length of the generator from two to 13. The result is shown in Figure 5a. It can be seen from Figure 5a that when the network length is greater than 7, the PSNR increases slowly with the increase of the network length. The trade-off between performance and speed of generator is shown in Table 3. In general, the deep network performs better than the shallow one at the expense of computational cost. We set 10 layers in the generative network (shown in Table 1) in the following experiments, to strike a balance between performance and speed.



(**a**) (**b**) (**c**)

Figure 5. (**a**) Length of the generator vs. the PSNR of generated testing painting; (**b**) length of the discriminator vs. the size of the recptive field; (**c**) λ vs. the PSNR of the generated testing painting.

Table 3. We evaluate the trade-off between performance and speed in networks with different depths.

Length of Generator	7	8	9	10	11	12	13
PSNR (dB)	30.829	30.832	30.837	30.846	30.842	30.852	30.856
Times (512×512)	0.023 s	0.027 s	0.029 s	0.032 s	0.035 s	0.039 s	0.044 s

The receptive-field is defined as the region in the input space at which a particular CNN's feature is looking. A large receptive-field can provide more context for predicting image details. The size of the receptive-field is determined by the kernel size, stride size and the length of the network. The receptive-field growth of our discriminator. Since the size of the input image is 256, we chose the network structure shown in Table 2, with a perceived field size of 382. In Equation (7), we analyze the

effect of λ on the generated painting, as shown in Figure 5c, when $\lambda = 50$ obtains the best PSNR in testing images.

4.3. Compare to Pix2pix

To compare with Pix2pix, we trained the Pix2pix framework using the same data, as well as the same training parameters. Pix2pix uses several downsampling steps. The same number of upsampling steps achieved by deconvolutional layers is used. Different from Pix2Pix, we used the all stride one convolutional layer to avoid producing checkerboard artifacts in the output images. Additionally, our generator is deeper than Pix2pix, which obtains a more non-linear fitting ability. Figure 6 shows the contrast of the results between our work and Pix2pix; the first row is the input sketch, and the second row shows the results of our proposed networks. The third row shows the results produced by Pix2pix, and the fourth row is the real image corresponding to the sketch. Table 4 shows quantitative evaluations of Pix2pix and our method. Compared to the results produced by Pix2pix, our multi-scale deep network can produce clearer paintings, no matter whether for detailed information or overall images. For example, our method can produce the facial details of the characters and the details of the trees. As is shown in our training of facial images in Figure 7, some input facial images do not have a nose sketch, but the ground truth images include a nose. Thus, our generative model has the ability to determine the location of the nose by sketches of the eye and mouth, shown in Figure 6.

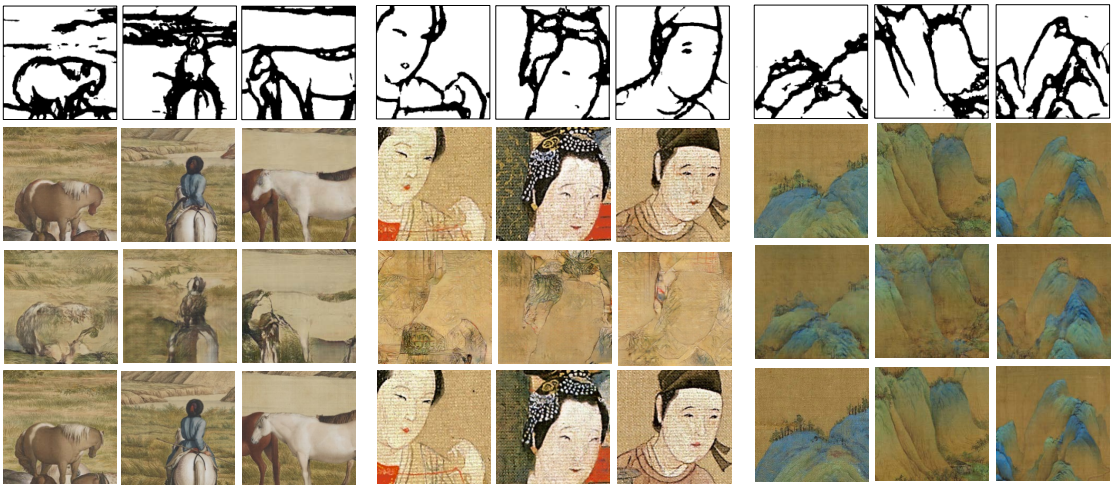


Figure 6. Comparison of experimental results to Pix2pix. From top to bottom: input sketch, our deep network, Pix2pix and the ground truth.

Figure 7. Some facial images in the training data.

Table 4. Quantitative evaluation of Pix2pix and our work. We evaluated average PSNR/SSIM for images in Figure 6.

Methods \ Images	Column 1 to 3	Column 4–6	Column 7–9
Pix2Pix (PSNR/SSIM)	28.38/0.4534	28.08/0.3421	28.85/0.5312
Ours (PSNR/SSIM)	30.95/0.7282	28.76/0.5854	31.84/0.6863

4.4. Validity of Multiscale

To test the validity of multiscale inputs, we input the multiscale sketch. The results are shown in Figure 8, indicating that our generative model works perfectly in multiscale inputs. From the first line, we can see that the proposed method not only can produce the detail information of the face, including eyes, mouth, and so on, but also can vividly depict the shape of the whole body. The resulting pictures in the second line show that our model can generate both small objects like trees and huge objects like distinct peaks and mountains.

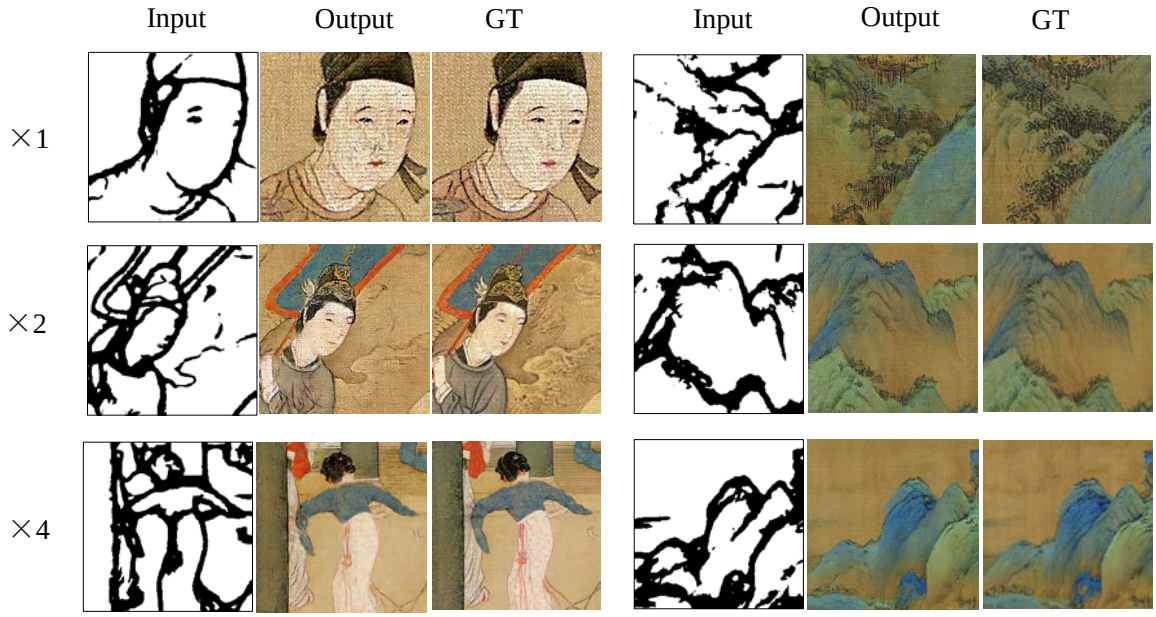


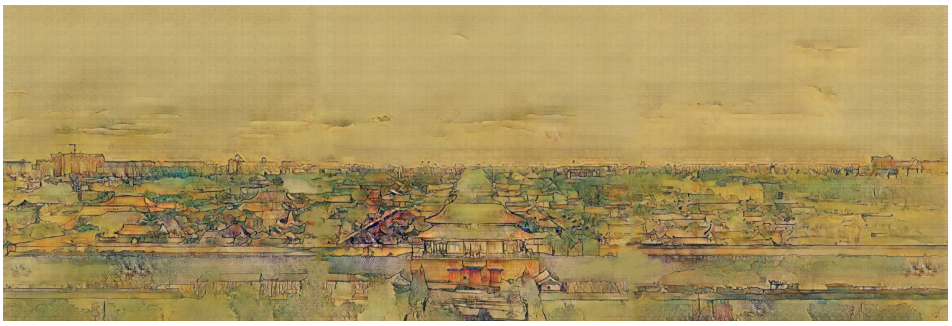
Figure 8. Multi-scale input and its corresponding output. We reduce the original image to 1/2 and 1/4 of the original size by bilinear interpolation to generate multiscale images. (**Top**) The small level scale input sketch; (**middle**) the middle-level scale input sketch; (**bottom**) the large level scale input sketch.

4.5. Style Transfer

At the same time, our models can also be treated as a neural style transfer: input the original image; extract the edge of the original image as a sketch; and input the sketch into the model. Then, we can get the image with the Chinese painting style. The generative model can handle any size of input image. As shown in Figure 9, we have style transitions for large 2300 × 768 images. Because of the graphics memory constraints, we divided this huge image into three parts to process. Different from other style transfer algorithms [2,38], our method can reduce distortions better.

(**a**)

Figure 9. *Cont.*



(**b**)

Figure 9. Transforming a real image into a Chinese painting. Our network can handle any size of images. (**a**) Input image; (**b**) Chinese painting produced by our network. The input image has an enormous size of (2300, 768); because of the graphics memory constraints, we divided this huge images into three parts to process.

Because our network uses a feed-forward approach, it performs significantly faster than previous deconvolution-based approaches. Therefore, it is possible to run style transfer in real-time or on video. As shown in Figure 10, we transform an aerial photograph to the Chinese painting style video; please see the additional files.

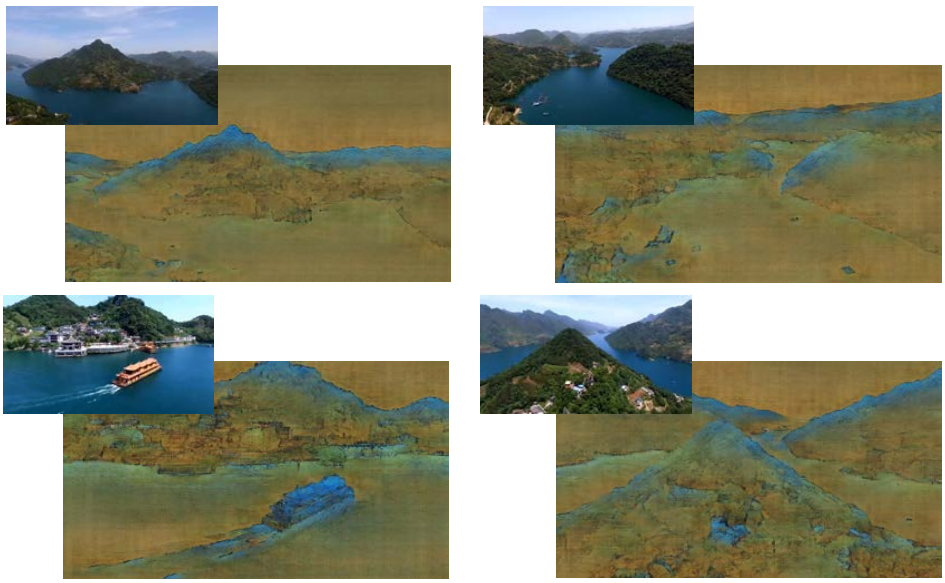
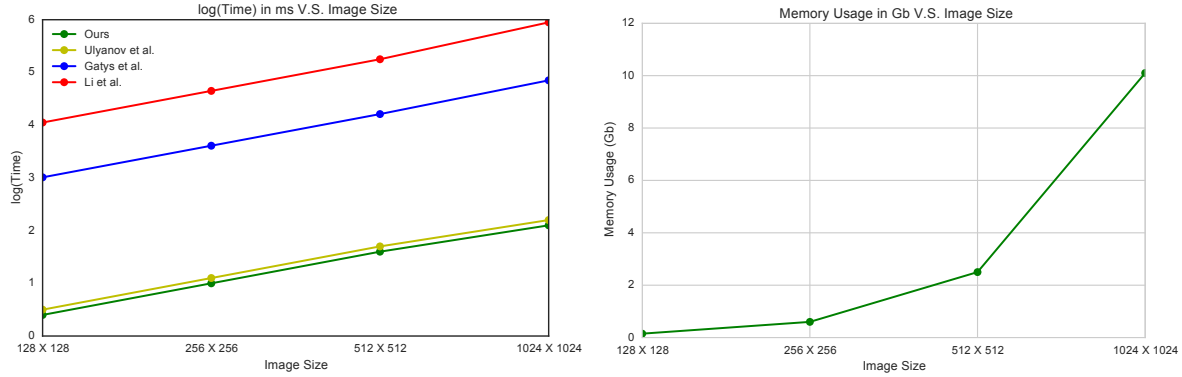


Figure 10. Transforming an aerial photograph into the Chinese painting style video. For the whole video, please see the additional files.

4.6. Time and Memory Usage

Finally, we compared the time complexity of the proposed methods and other style transfer algorithms, as shown in Figure 11a. Although the computation time between Ulyanov et al.'s [39] and our method is similar (the comparative results are shown in Figure 12), our result can keep the structure of the input image. Table 5 compares the runtime of our method and other methods for several image sizes. Compared to Gatys et al. [2], our method is three orders of magnitude faster. Our method processes 512×512 images at about 30 FPS, making it feasible to run in real time or on video.



(**a**) (**b**)

Figure 11. (**a**) Speed comparison (in log space) between our method and Gatys et al. [2], Li et al. [38] and Ulyanov et al. [39]. The feed-forward methods (ours and Ulyanov et al. [39]) are significantly faster than Gatys et al. [2] (500-times speed up) and Li et al. [38] (5000-times speed up). (**b**) Memory usage vs. image size, the required memory is linearly related to the size of input image.

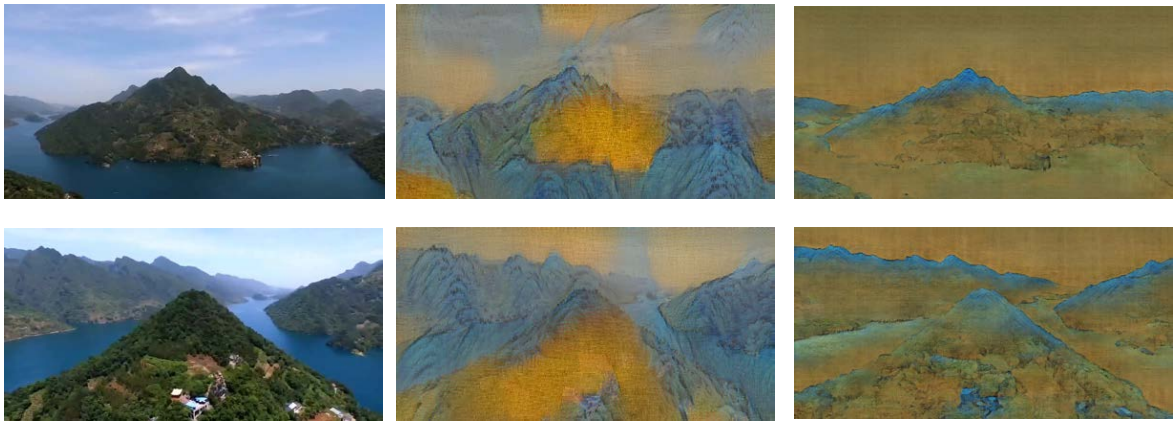


Figure 12. Example results of style transfer using our image transformation networks. (left) Input image; (middle) results of Ulyanov et al. [39]; (right) ours.

Table 5. Speed (in seconds) for our style transfer network vs. the optimization-based baseline for varying numbers of iterations and image resolutions. Both methods are benchmarked on a GTX 1080 GPU.

Methods / Image Size	Gatys et al. [2]	Ulyanov et al. [39]	Ours
128 × 128	1.542 s	0.004 s	0.002 s
256 × 256	6.483 s	0.015 s	0.008 s
512 × 512	25.23 s	0.051 s	0.032 s
1024 × 1024	106.2 s	0.212 s	0.122 s

Regarding memory, our generative model requires 2 MB to save its parameters. In the process of generating Chinese paintings, the required memory is linearly related to the size of input sketch, as shown in Figure 11b: for 256 × 256 pictures, it takes about 600 Mb; for 512 × 512 pictures, it consumes 2.5 Gb memory.

5. Other Applications

Besides inputting simple sketches to synthesize surprising art paintings, the method we proposed also can be treated as a general-purpose solution to image-to-image translation problems, such as image colorization, image super-resolution, and so on. To verify the universality of the proposed method, we apply it in colorization and super resolution.

5.1. Image Colorization

The goal of colorization is not to recover the actual ground truth color, but rather, to produce a plausible colorization that the user finds useful even if the colorization differs from the ground truth color. Colorization can seem like a daunting task because so much information is lost (two out of three color dimensions) in converting a color image to its underlying grayscale representation. Because grass is usually green and the sky is usually blue, the semantics of an image scene provide many clues for a plausible colorization.

To verify the effectiveness of the proposed method, we artificially colorize grayscale satellite imagery. The dataset used is the UC Merced Land Use dataset [40]; this dataset consists of images of 21 land use classes (100 256 $\times$ 256-pixel RGB images for each class) selected from aerial optical images acquired by the U.S. Geological Survey. Given the lightness channel L, our system predicts the corresponding a and b color channels of the image in the CIE Lab color space. Lab color space is an alternative system for representing pixel colors versus the standard RGB values. It is useful because the L channel is statistically independent of the pure color a, b channels. We divide the training set and test set according to the ratio of 8:2. After training, we display the resulting colorizations of our method in Figure 13, the grayscale image on the left and the artificial colorization of a grayscale image on the right. The result demonstrates that our method can produce a realistic colorization.

Figure 13. Example artificial colorization of satellite images: grayscale images (**left**), colorful images (**middle**) and ground truth (**right**).

5.1.1. Comparisons with Other Approaches

We compare against other methods in Figure 14. Our approach accurately colorized the ocean and beach, while the other two approaches biased tones towards either the ocean or the beach.

Following the previous work [25], we conducted a user study by asking "Does this image look natural?" to evaluate the naturalness of the ground truth, as well as the results of our model and other approaches. The images are randomly selected and displayed one by one to the user. This research was done by five different users. For each type, we show approximately 100 images from 400 images in total. The users are instructed to use their intuition and feelings rather than spending too much time on the detail of the image. We can see the result in Table 6. For this, 93.80% of our method's images look natural, 96.60% of the ground truth. This strongly suggests that our model can create realistic colorizations.

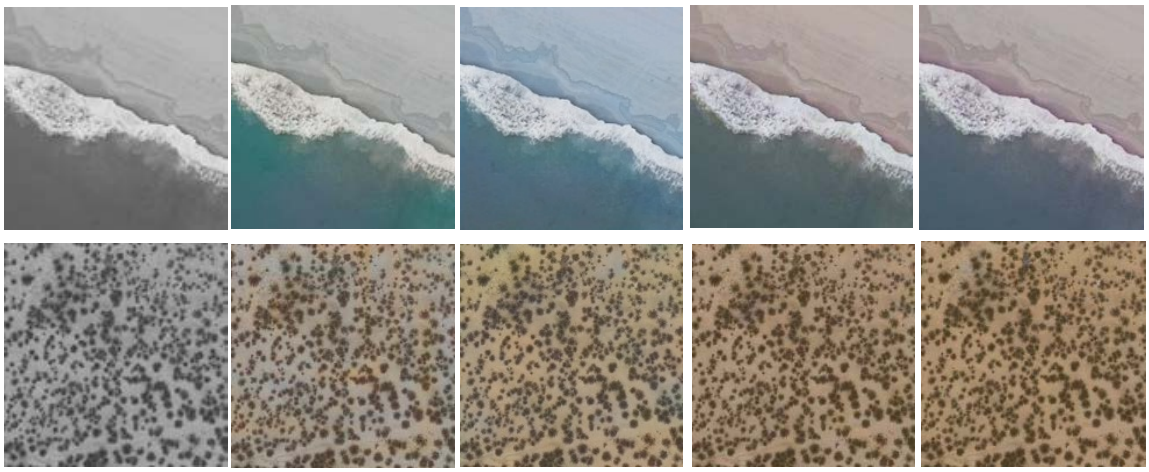


Figure 14. We compare against the other methods. The first column contains the input image; the second column is the result of [25]; the third column is the result of [41]; the fourth column is our result; and the last row is the ground truth.

Table 6. Results of our user study evaluating the naturalness of the colorizations.

Approach	Iizuka et al. [25]	Zhang et al. [41]	Ours	Ground Truth
Naturalness	78.20%	80.80%	93.80%	96.6%

5.2. Image Super-Resolution

Single Image Super-Resolution (SR), which aims to recover a high-resolution image from a low-resolution image, is a widespread problem in image processing. A high resolution with higher pixel density contains more details, which play an essential part in some applications. To train the network we proposed, we use 91 images from [42] and 200 images from the training set of [43]. Following previous works [44–46], we only consider the luminance channel in YCbCr color space, because humans are more sensitive to luminance changes. For benchmark, we use two datasets: Set5 [47], Set14 [46].

In Figure 15, we showed the super-resolution results of our method, and it can reconstruct detailed textures and edges in the high-resolution images.

Figure 15. Super-resolution results of images in Set5 [47] and Set14 [46]. Our method takes the bicubic interpolation of low-resolution images (**left**) as input and high-resolution images (**right**, 4×) as output.

5.2.1. Comparisons with Other Approaches

Our proposed work is compared with other SR algorithms: A+ [48], SelfExSR [49] and SRCNNs [50]. Table 7 summarizes quantitative results on the four testing datasets. Our method outperforms all previous methods in these datasets. In Figure 16, we compare our method with some state-of-the-art methods. Our method reconstructs detailed textures and sharper edges in the HR images and provides noticeable improvement compared to other works.

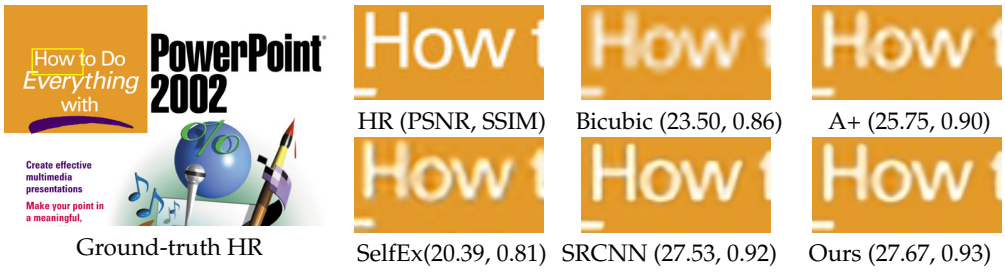


Figure 16. Image "ppt3" (Set14, ×3): there are sharper edges between "H", "o" and "w" in our result; while in other works, the text is blurry. SR, Super-Resolution.

Table 7. Quantitative evaluation of state-of-the-art SR methods. We evaluated average PSNR/SSIM for scale factors ×2, ×3 and ×4 on datasets Set5, Set14.

Dataset	Scale	Bicubic PSNR/SSIM	A+ PSNR/SSIM	SelfEx PSNR/SSIM	SRCNN PSNR/SSIM	Ours PSNR/SSIM
Set5	×2	33.66/0.9299	36.54/0.9544	36.54/0.9537	36.49/0.9537	36.66/0.9542
	×3	30.39/0.8682	32.58/0.9088	32.43/0.9057	32.58/0.9093	32.75/0.9090
	×4	28.42/0.8104	30.28/0.8603	30.14/0.8548	30.31/0.8619	30.48/0.8628
Set14	×2	30.24/0.8688	32.28/0.9056	32.26/0.9040	32.22/0.9034	32.42/0.9063
	×3	27.55/0.7742	29.13/0.8188	29.05/0.8164	29.16/0.8196	29.28/0.8209
	×4	26.00/0.7027	27.32/0.7491	27.24/0.7451	27.40/0.7518	27.49/0.7503

6. Conclusions and Future Work

In this paper, we propose a multiscale deep neural network to transform sketches into Chinese paintings that can also be treated as neural style transfer. To synthesize more realistic imagery, we train the generative network by using both L1 loss and adversarial loss. Furthermore, additional experiments on remote sensing images and street images demonstrate the universality of our proposed approach. The models we proposed can produce surprising artworks, but there is still a big difference between the works produced by artists. In future work, we may focus on some abstract artworks, which is still a difficult problem for artificial intelligence.

Acknowledgments: This work was supported in part by the National Natural Science Foundation of China under Grant No. 41501485 and No. 61331017.

Author Contributions: Yang Wang, Guangluan Xu, Jun Li and Kun Fu conceived and designed the experiments; Daoyu Lin performed the experiments and wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jing, Y.; Yang, Y.; Feng, Z.; Ye, J.; Song, M. Neural Style Transfer: A Review. *arXiv* **2017**, arXiv:1705.04058.
2. Gatys, L.A.; Ecker, A.S.; Bethge, M. A neural algorithm of artistic style. *arXiv* **2015**, arXiv:1508.06576.
3. Gatys, L.A.; Ecker, A.S.; Bethge, M. Image style transfer using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2414–2423.

4. Shih, Y.; Paris, S.; Barnes, C.; Freeman, W.T.; Durand, F. *Style Transfer for Headshot Portraits*; Association for Computing Machinery (ACM): New York, NY, USA, 2014.
5. Efros, A.A.; Freeman, W.T. Image quilting for texture synthesis and transfer. In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 12–17 August 2001; ACM: New York, NY, USA, 2001; pp. 341–346.
6. Wei, L.Y.; Levoy, M. Fast texture synthesis using tree-structured vector quantization. In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, New Orleans, LA, USA , 23–28 July 2000; ACM Press/Addison-Wesley Publishing Co.: Boston, MA, USA, 2000; pp. 479–488.
7. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems, Proceedings of the Neural Information Processing Systems, Lake Tahoe, Nevada, 3–6 December 2012*; Curran Associates, Inc.: Red Hook, NY, USA, 2012; pp. 1097–1105.
8. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.
9. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv* **2015**, 1–15, arXiv:1511.06434.
10. Johnson, J.; Alahi, A.; Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Berlin, Germany, 2016; pp. 694–711.
11. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *arXiv* **2014**, 1–9, arXiv:1406.2661.
12. Sangkloy, P.; Lu, J.; Fang, C.; Yu, F.; Hays, J. Scribbler: Controlling Deep Image Synthesis with Sketch and Color. *arXiv* **2016**, arXiv:1612.00835.
13. Güçlütürk, Y.; Güçlü, U.; van Lier, R.; van Gerven, M.A. Convolutional sketch inversion. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Berlin, Germany, 2016; pp. 810–824.
14. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
15. Eitz, M.; Richter, R.; Hildebrand, K.; Boubekeur, T.; Alexa, M. Photosketcher: Interactive sketch-based image synthesis. *IEEE Comput. Graph. Appl.* **2011**, *31*, 56–66.
16. Chen, T.; Cheng, M.M.; Tan, P.; Shamir, A.; Hu, S.M. Sketch2photo: Internet image montage. *ACM Trans. Graph.* **2009**, *28*, doi:10.1145/1618452.1618470.
17. Shih, Y.; Paris, S.; Durand, F.; Freeman, W.T. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Trans. Graph.* **2013**, *32*, doi:10.1145/2508363.2508419.
18. Kwatra, V.; Schödl, A.; Essa, I.; Turk, G.; Bobick, A. Graphcut textures: Image and video synthesis using graph cuts. *ACM Trans. Graph.* **2003**, *22*, 277–286.
19. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
20. Rezende, D.J.; Mohamed, S.; Wierstra, D. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 1278–1286.
21. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems, Proceedings of the Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014*; Curran Associates, Inc.: Red Hook, NY, USA, 2014; pp. 2672–2680.
22. Van den Oord, A.; Kalchbrenner, N.; Espeholt, L.; Vinyals, O.; Graves, A. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems, Proceedings of the Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016*; Curran Associates, Inc.: Red Hook, NY, USA, 2016; pp. 4790–4798.
23. Hertzmann, A.; Jacobs, C.E.; Oliver, N.; Curless, B.; Salesin, D.H. Image analogies. In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 12–17 August 2001; ACM: New York, NY, USA, 2001; pp. 327–340.
24. Yan, X.; Yang, J.; Sohn, K.; Lee, H. Attribute2image: Conditional image generation from visual attributes. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Berlin, Germany, 2016; pp. 776–791.

25. Iizuka, S.; Simo-Serra, E.; Ishikawa, H. Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. *ACM Trans. Graph.* **2016**, *35*, 110:1–110:11.
26. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
27. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved Techniques for Training GANs. In *Advances in Neural Information Processing Systems, Proceedings of the Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016*; Curran Associates, Inc.: Red Hook, NY, USA, 2016; pp. 1–10.
28. Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. *arXiv* **2016**, arXiv:1606.03657.
29. Dixon, D.; Prasad, M.; Hammond, T. iCanDraw: Using sketch recognition and corrective feedback to assist a user in drawing human faces. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Denver, CO, USA, 7–11 May 2010; ACM: New York, NY, USA, 2010; pp. 897–906.
30. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. *arXiv* **2016**, arXiv:1611.07004.
31. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784.
32. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Berlin, Germany, 2015; pp. 234–241.
33. Zeiler, M.D.; Krishnan, D.; Taylor, G.W.; Fergus, R. Deconvolutional networks. In Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 2528–2535.
34. Odena, A.; Dumoulin, V.; Olah, C. Deconvolution and Checkerboard Artifacts. 2016. Available online: http://distill.pub/2016/deconv-checkerboard/ (accessed on 15 July 2017).
35. Xie, S.; Tu, Z. Holistically-nested edge detection. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1395–1403.
36. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv* **2016**, arXiv:1603.04467.
37. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of The 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
38. Li, C.; Wand, M. Combining markov random fields and convolutional neural networks for image synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2479–2486.
39. Ulyanov, D.; Lebedev, V.; Vedaldi, A.; Lempitsky, V. Texture networks: Feed-forward synthesis of textures and stylized images. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016.
40. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; ACM: New York, NY, USA, 2010; pp. 270–279.
41. Zhang, R.; Zhu, J.Y.; Isola, P.; Geng, X.; Lin, A.S.; Yu, T.; Efros, A.A. Real-Time User-Guided Image Colorization with Learned Deep Priors. *ACM Trans. Graph.* **2017**, *36*, doi:10.1145/3072959.3073703.
42. Yang, J.; Wright, J.; Huang, T.S.; Ma, Y. Image super-resolution via sparse representation. *IEEE Trans. Image Process.* **2010**, *19*, 2861–2873.
43. Martin, D.; Fowlkes, C.; Tal, D.; Malik, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In Proceedings of the Eighth IEEE International Conference on Computer Vision, Vancouver, BC, Canada, 7–14 July 2001; pp. 416–423.

44. Chang, H.; Yeung, D.Y.; Xiong, Y. Super-resolution through neighbor embedding. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 27 June–2 July 2004.
45. Glasner, D.; Bagon, S.; Irani, M. Super-resolution from a single image. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 349–356.
46. Zeyde, R.; Elad, M.; Protter, M. On single image scale-up using sparse-representations. In Proceedings of the International Conference on Curves and Surfaces, Avignon, France, 24–30 June 2010; pp. 711–730.
47. Bevilacqua, M.; Roumy, A.; Guillemot, C.; Morel, A. Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding. In Proceedings of the 23rd British Machine Vision Conference (BMVC), Guildford, UK, 3–7 September 2012.
48. Timofte, R.; De Smet, V.; Van Gool, L. A+: Adjusted anchored neighborhood regression for fast super-resolution. In Proceedings of the Asian Conference on Computer Vision, Singapore, 1–5 November 2014; Springer: Berlin, Germany, 2014; pp. 111–126.
49. Huang, J.B.; Singh, A.; Ahuja, N. Single image super-resolution from transformed self-exemplars. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5197–5206.
50. Dong, C.; Loy, C.C.; He, K.; Tang, X. Learning a deep convolutional network for image super-resolution. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin, Germany, 2014; pp. 184–199.