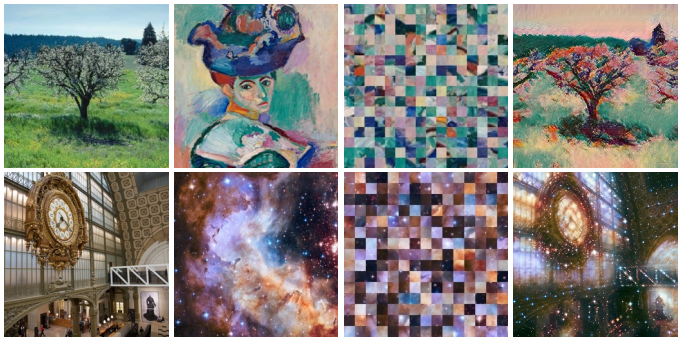
P^2-GAN: Efficient Stroke Style Transfer using Single Style Image

Zhentan Zheng, Jianyi Liu*, and Nanning Zheng, *Fellow, IEEE*

***Abstract*—Style transfer is a useful image synthesis technique that can re-render given image into another artistic style while preserving its content information. Generative Adversarial Network (GAN) is a widely adopted framework toward this task for its better representation ability on local style patterns than the traditional Gram-matrix based methods. However, most previous methods rely on sufficient amount of pre-collected style images to train the model. In this paper, a novel Patch Permutation GAN (P^2-GAN) network that can efficiently learn the stroke style from a single style image is proposed. We use patch permutation to generate multiple training samples from the given style image. A patch discriminator that can simultaneously process patch-wise images and natural images seamlessly is designed. We also propose a local texture descriptor based criterion to quantitatively evaluate the style transfer quality. Experimental results showed that our method can produce finer quality re-renderings from single style image with improved computational efficiency compared with many state-of-the-arts methods.**

***Index Terms*—Style Transfer, Generative Adversarial Network, Stroke Style, Patch Permutation**



(a) Content images (b) Style images (c) Patch Permutation (d) Our results

Fig. 1. Style transfer from single style image by our method. The benefits of using patch permutation include: (1) only one style image can generate multiple training images, (2) from the style image, the local stroke style can be better learnt.

I. Introduction

STYLE transfer aims at redrawing a content image into another artistic style while preserving its semantic content. It has a rich history in terms of texture transfer [1][2]. Style transfer technique enables amateurs to produce fantastic and versatile images in seconds, so that it has many practical applications in cartoon generation, oil painting re-rendering and fashion design etc [3][4][5].

Most early works on texture transfer were pixel-level approaches that relied on low-level visual features. Since Gatys *et al.* [6] for the first time incorporated convolutional neural networks (CNNs) into the computations of style loss and content loss, CNNs based approaches have become the mainstream of style transfer and achieved convincing results. These methods can be divided into three categories: a) online learning methods [7] [8], b) patch-swap methods [9][10][11] , c) offline learning methods [12][13][14].

The former two kinds of methods usually suffer from expensive online computational time (20s˜300s) since no style prior information has been taken into consideration. On the contrary, offline learning methods always use the feed-forward networks, mostly auto-encoder, trained from a set of pre-collected style images to memorize the style information, so that the target image can be obtained in real time through feed-forward computation[12][13][15] . These methods can be regarded as a trade of computational time from online stage to offline stage.

The offline learning methods mainly include two types: a) Gram matrix based methods[12][16], b) Generative Adversarial Network (GAN) based methods[17][18]. In GAN, the generative network implicitly learns the target style, aiming to fool the discriminator, while the discriminative network tries to distinguish the real/fake images by two-player minimax game. Compared with the Gram matrix based style loss, the adversarial loss in GAN can better learn the local style patterns. However, collecting sufficient amount of style images is sometimes difficult in practice and always makes the output style hard to control. That is, multiple training images are apt to generate unexpected texture effect since no two style images are exactly the same.

In this paper, we are interested in the offline learning based real-time style transfer method, and propose to use only one style image for the training of GANs. It will make it possible to precisely simulate the expected stroke style from the customer-designated image without interfered by other sources. A novel *Patch Permutation* GAN (P^2-GAN) network is proposed. Our idea is to randomly break the unique style image into multiple patches which will be used as the training set. The structure of content images is preserved through an encoder-decoder generator. Some results by our method are shown in Figure.1. To the best of our knowledge, only one GAN-based previous work namely MGAN[8] has taken into consideration of the single image training problem in style transfer. The main difference between MGAN and our method

Manuscript received April 19, 2005; revised August 26, 2015.

The authors are with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Shaanxi 710049, China.(e-mail: xjtu_evi@stu.xjtu.edu.cn; jyliu@xjtu.edu.cn; nnzheng@xjtu.edu.cn). *Corresponding author: Jianyi Liu.

is that their patches locate in the feature maps while ours are segmented in the original image space for a better control of local stroke style. In addition, we have also designed a novel *patch discriminator* to simultaneously process patch-wise images and natural images, which makes our network training more efficient.

The rest of paper is organized as follows: In Section II, we briefly review the related works and their relationship with out method. In section III, our proposed P^2-GAN method will be described in detail. Then, we will discuss the experimental results in Section IV, and conclude in Section V.

II. Related Work

In this section, we briefly review techniques of style transfer. Three types of mainstream methods and their relationship with our work are summarized. Specially, the single style image based transfer methods are discussed.

A. Gram matrix based methods

Gatys *et al.* [7] for the first time applied CNN into the style transfer task. A pre-trained VGG network is used to extract the feature maps, based on which the content loss is measured. Together with the style loss computed by the Gram matrix descriptor, their method achieved impressive re-rendering effect. However, since the transferred images are optimized from random initialized pixels by iteration manner, this method suffers from expensive online time (several minutes per image).

After that, several works were proposed toward improving its computational efficiency. By moving the style learning process from online stage to offline stage, the style transfer can be realized by the feed-forward network, which makes the online computational time much less than early methods. Johnson *et al.*[12] proposed a real-time style transfer method by using a Gram matrix based perceptual loss in an encoder-decoder network. Ulyanov *et al.*[16] proposed the TextureNet in which multiple scales on feature maps are considered to refine the style loss. They have further improved the TextureNet by using Instance Normalization (IN) to replace Batch Normalization(BN) with significant improvements to the quality of image stylization[15].

The common feature of these methods is that the pre-trained CNN is used to extract the feature maps, and the iteration manner is adopted to learn the transferred images. In addition, the reliance of Gram matrix in these methods has also limited the ability of grasping the "true" styles. Wilmot and Risser *et al.*[19] found that two images with totally distinct distributions may produce similar Gram matrices, since Gram matrix can only measure the global covariance of the image while ignoring the local diversities.

B. Patch-swap based methods

This type of methods usually use patch-wise search technique such as Patchmatch [20] to find an appropriate correspondence relationship of all patches between the content image and style image, then the style transfer can be accomplished by patch-swap on corresponding patches. Liao *et al.* [10] proposed the DIA algorithm which implements the coarse-to-fine patch correspondence on a five layer feature map pyramid extracted from the pre-trained CNN. A patch aggregation strategy is then used to fuse all patches coming from the style image. Similar approaches also include DFR[11] and CNNMRF [21].

Since all the patches in the re-rendered image are coming from the style image, the texture qualities here are always more realistic than the pixel-level synthesis in the online learning based methods. However, the patch search algorithm is also very time consuming which hinders the above methods from real-time transferring. In addition, the patch correspondence always requires the content/style image pairs to have similar semantic structure, otherwise the patch search will make no sense. This also restricts the applicability of these methods.

C. GAN based methods

GAN is a widely adopted model for offline style learning. The adversarial loss instead of the Gram-matrix based style loss can better learn the local style patterns. Cycle-GAN proposed by Zhu *et al.*[17] constructed a pair of GANs according to the principle of cycle-consistent. One pair of generator and discriminator is in charge of stylization, and the other pair is in charge of de-stylization. Chen *et al.*[18] proposed the gated-GAN to enable multiple-style transfer within a uniform network by means of a gate control strategy. The GAN based methods always rely on large amount of training images with the same style. Meanwhile, multiple training images may always induce the final re-rendered results toward unexpected direction, since each image has different visual characteristics on different scales.

In recent years, the concept of style transfer has been further extended to image-to-image translation task, which means converting one possible representation of a scene into another. In DualGAN[22], image translations such as daytime image to night scene, map images to aerial view photos, and plastic surface material to metal surface are implemented. cGAN[23] shows abilities of hand drawn sketches to photorealistic images. In starGAN [24], facial attributes such as age/gender/expression are altered with each other. Auto-Embedding GAN [25] can synthesize high resolution images from low resolution ones, and FuseGAN[26] can generate single fusion image from multi-focus images. The conditional GAN (cGAN) [27] is widely adopted in these methods to enhance control on data being generated by adding class labels or other additional information onto the generator and discriminator.

The technique of image translation is also used to generate synthetic data that is difficult to collect in reality. Wang *et al.* [28] proposed a SSIM Embedding CycleGAN to translate the synthetic images to the photo-realistic images for the task of crowd understanding. The author also designed an adversarial domain adaptation architecture for semantic segmentation in urban scenes[29].

D. Single image style transfer

If we limit the training set with only one sample, most existing GAN networks can't be possibly trained. The mo-

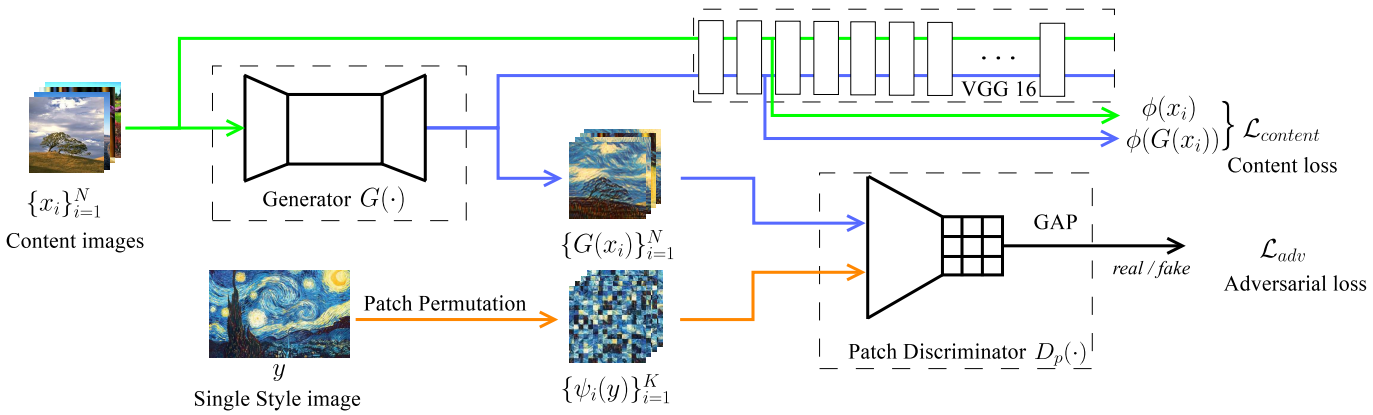


Fig. 2. Network architecture of the proposed P^2-GAN. A Patch Permutation operator is designed to generate training set from single style image. The proposed Patch Discriminator is compatible to both patch-wise images and natural images. The style transferred results can be obtained from the offline trained encoder-decoder generator $G(\cdot)$.

tivation of single style image transferring is that precise style exactly same with the training image can be expected, and the collection of training set is also avoided. The patch-swap based methods can be used for single image style transferring. However, none of these methods can run in real-time (typically cost 20sec to 3min). Some offline learning methods such as JohnsonNet[12], TextureNet[15] can also implement stylization using single style image, but their transferring qualities are still unsatisfactory due to the limitation of Gram matrix based global descriptor.

To the best of our knowledge, MGAN[8] is the only method that addressed the single style image training problem in GAN. Multiple patches are generated in this method by regular sampling from the feature map for the training of discriminative network. However, the feature map sampling always produces blurred re-renderings. In this paper, we will develop a novel GAN based single image transfer network with excellent performances both on image quality and computational efficiency.

III. Proposed Method

A. Problem Statement

In most GAN-based style transfer methods [14][18], a generator $G(\cdot)$ and a discriminator $D(\cdot)$ are usually learned from a set of content images $\mathcal{X} = \{x_i\}_{i=1}^N$ and style images $\mathcal{Y} = \{y_i\}_{i=1}^M$ simultaneously, where N and M denote the numbers of content/style images respectively. Given a new content image x_t, the re-rendered image $G(x_t)$ will preserve the content information of x_t and the style information of $\mathcal{Y}$.

In this paper, we focus on the transfer of "stroke" style. We name the strokes as the movements or marks of a pen or brush that artists make with it when painting. The red boxes in the Figure.3. illustrate some examples of artistic strokes. Compared with the "structure" features which usually imply global information, stroke style can only be better described locally. Obviously, multiple style image ($M > 1$) is against the goal of stroke transfer, since no two images are exactly the same with their strokes.

In many global descriptor based style transfer methods, some unwanted secondary texture like the signature of painter

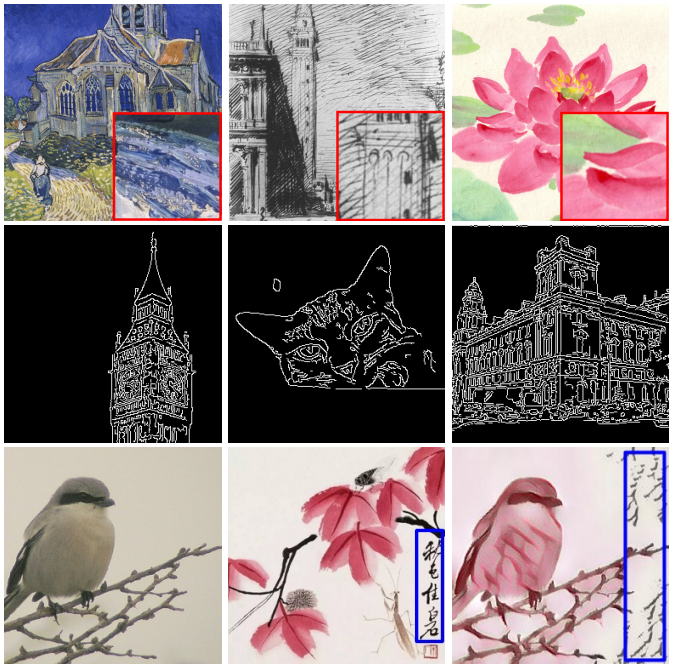


Fig. 3. Illustration of the "stroke" style, "structure" information, and "leakage" phenomenon discussed in this paper. The enlarged "stroke" styles are showed in the 1^{st} row. In the 2^{nd} row, we use the edge map to represent the "structure" of the images. In the 3^{rd} row, some unwanted structure (blue boxes) in the style image (middle) has been wrongly "leaked" into the synthetic image (right).

on the style image will be wrongly transferred into the final results. This will produce apparent artifacts and impair the quality of synthetic image. An example is shown in the third row of Figure.3. We name it as the "leakage" phenomenon, since some unwanted structure information besides stroke in the style image has been wrongly leaked into the results. In this paper, we will also analyze this problem and show our solution.

When $M = 1$, the training stage of standard GANs will become instable since single sample can not represent a distribution, and the network may even collapse while training. To overcome this problem, we have designed a novel network architecture as shown in Figure.2. A patch permutation module

$\psi(\cdot)$ and a patch discriminator $D_p(\cdot)$ are proposed. Together with an improved encoder-decoder generator $G(\cdot)$ for more efficient computations on both offline and online stages, high quality stroke style transfer using single style image can be accomplished in real time.

B. Patch Permutation

Since we only have one style image, we consider to break the style image into multiple patches, and learn stroke style from the patches. Therefore, a permutation method for the style image on the image is proposed.

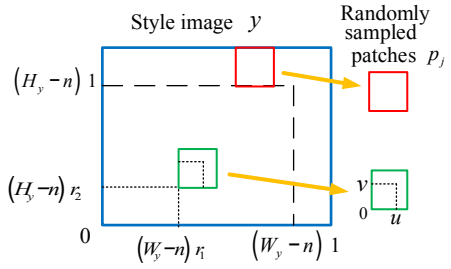


Fig. 4. The random sampling process from style image to any a patch for the patch permutation. The coordinate relationship of (1) is illustrated.

Firstly, patches with size $n \times n$ are cropped from style image y at its random position, n should be selected according to the scale of texture element in y. For any patch p_j, we have:

$$p_j(u, v) = y[(W_y - n)r_1 + u, (H_y - n)r_2 + v] \quad (1)$$

where $u, v \in \{1, \ldots, n\}$ are the horizontal and vertical coordinates respectively, and r_1, r_2 are random variables that obey the uniform distribute $\mathbf{U}(0, 1)$. W_y and H_y denote the width and height of image y respectively. The coordinate relationship of (1) is illustrated in Fig.4. After T^2 times random cropping, all the patches $p_j, j = 1, ..., T^2$, can be reorganized into a new image $\psi(y)$ that yields to the following block-matrix form:

$$\psi(y) = \begin{bmatrix} p_1 & p_2 & \cdots & p_T \\ p_{T+1} & p_{T+2} & \cdots & p_{2T} \\ \vdots & \vdots & \ddots & \vdots \\ p_{(T-1)T+1} & p_{(T-1)T+2} & \cdots & p_{T^2} \end{bmatrix} \quad (2)$$

where $\psi(y)$ has a size of $nT \times nT$ and can be regarded as a *permutation* of the original style image y.

Repeating such permutation K times, the so obtained image set $\mathcal{Y}_\psi = \{\psi_i(y)\}_{i=1}^K$ will be used as the training set for our model instead of the single image y. In $\mathcal{Y}_\psi$, the stroke style information of y will be preserved by selecting a suitable patch size n. Meanwhile, the structure information of y has been discarded, which is benefit to avoid content interference from y.

C. Patch Discriminator

The operation of patch permutation in previous section will inevitably generate discontinuous textures on junction pixels between adjacent patches in $\psi_i(y)$. This will incur severe artifacts in the transferred image by traditional discriminators. To address this problem, we propose a discriminator D_p which will be described in detail as below.

We also adopt the L-layer CNN structure similar to PatchGAN[30][17]. However, different from PatchGAN's setup where a relatively small stride (1 or 2) is applied, we restrict a special relationship between the stride and other network parameters. Assuming each patch is with a size of $n \times n$, the convolution kernel at each convolutional layer l has a size of $k_l \times k_l$, and the stride is set to s_l. The following rule will be imposed in our discrimination D_p:

$$\begin{cases} k_l = s_l \\ n = \prod_{l=1}^{L} k_l \end{cases} \quad (3)$$

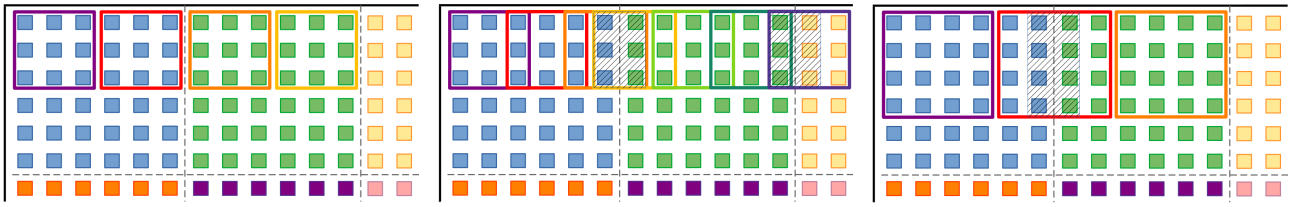
This rule will ensure that any convolution computation in each layer will be performed on one inner patch, and the sliding kernel will never have chance to stretch across different patches. This means our network will only feed forward *complete* textures from $\psi_i(y)$ although discontinuous textures do exist in it. Therefore, unlike PatchGAN which aims to classify whether overlapping image patches are real or fake, our discriminator classifies the independent patches. Since the network parameters in D_p are relevant to patch size, we name D_p the *patch discriminator* in this paper. The principle of D_p is illustrated in Figure.5.

Given an input $\psi_i(y)$ with size of $nT \times nT$, there will be T^2 elements on the output of the last convolutional layer, and finally, a global average pooling is used to vote up the final discriminative result. Besides the ability of compute patch permuted image, D_p can also process natural images such as $G(x_i)$ with no difference with any traditional discriminator since the CNN architecture hasn't been changed. The ability of simultaneously processing multiple patches in $\psi_i(y)$ and natural images $G(x_i)$ by single discriminator makes our network training very efficient.

D. Encoder-decoder Generator

For the purpose of style transfer, we design the generator with the encoder-decoder architecture similar to [31][30][32]. Depthwise separable convolutions (DSC) has been widely used to accelerate the feed-forward computation of CNN network[33][34]. The convolution computation in DSC is separated into two parts: a depth-wise convolution and a point-wise convolution. In our network, we also adopted the DSC architecture in the generator $G(\cdot)$. A 3×3 DSC module is implemented in both the encoder layer and the decoder layer (as shown in Fig.6(b), 6(c)). Between each depthwise convolution and pointwise convolution, we also adopted the Instance Normalization (IN) operation[15] to improve the quality of image stylization. Compared with the standard convolution operation, the above design of encoder-decoder generator has accounted for the 8 - 9 times decrease of parameter amount as shown in the experiments.

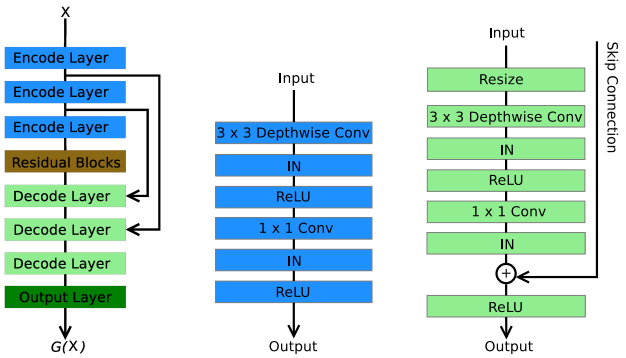
To address the problem of checkerboard artifacts incurred by the traditional deconvolution (usually as transposed convolution), we adopted the resize convolution operation[35][36]



(a) stride = kernel size = 3, patch size = 6 (b) stride = 2, kernel size = 3, patch size = 6 (c) stride = kernel size = 4, patch size = 6

Fig. 5. Principle of the proposed patch discriminator. We set the kernel size k_l and convolutional stride s_l to be a factor of the patch size n as in (a) to enable our discriminator on patch permuted images. In contrast, arbitrary setting in standard convolution as in (b) and (c) will produce severe artifacts due to discontinuous texture across adjacent patches.

before the DSC in each decoder layer. The feature maps will be upsampled into two times bigger than its original size using the nearest-neighbor interpolation. In all encoder layers and decoder layers, $relu$ is used as the activation function.



(a) Generator $G(\cdot)$ (b) Each encoder layer (c) Each decoder layer

Fig. 6. Structure of proposed generator. The generator is an encoder-decoder with skip connections between mirrored layers to improve the efficiency of training process. We adopted DSC to reduce computational consumption and resize convolution to improve the stylization quality.

E. Objective Function

Based on the patch discriminator proposed above, the adversarial loss function used in this paper is defined as follows:

$$\mathcal{L}_{adv}(G, D_p) = \mathbb{E}_{\psi(y)\in\mathcal{Y}_\psi}[\log D_p(\psi(y))] + \mathbb{E}_{x\in\mathcal{X}}[\log(1 - D_p(G(x)))] \quad (4)$$

The encoder-decoder generator $G(\cdot)$ will learn the specific stroke style coming from the unique style image y by optimizing the adversarial loss. Meanwhile, to preserve the structure information coming from the content image we compute the content loss in a same way as in [12]. Let $\phi_l(x)$ denote the feature map of x on the output of l-th layer in a $VGG16$ pretrained on ImageNet [37], the content loss is defined as the normalized squared Euclidean distance between the corresponding feature maps of original images and transferred images:

$$\mathcal{L}_{content}(G) = \frac{1}{C_l H_l W_l}\mathbb{E}_{x\in\mathcal{X}} \parallel \phi_l(x) - \phi_l(G(x)) \parallel_2^2 \quad (5)$$

where C_l, H_l and W_l denote the channel, height and weight of the l-th layer feature maps respectively. In our experiments, l is set to $relu1\_2$ to compute $\mathcal{L}_{content}$.

The final objective function is defined as follows:

$$\min_G \max_{D_p} \mathcal{L}(G, D_p) = \mathcal{L}_{adv}(G, D_p) + \lambda\mathcal{L}_{content}(G) \quad (6)$$

where λ is a factor to balance the adversarial loss and content loss. By solving the two-player optimization problem in (6), for any given image x_t, the transferred image $G(x_t)$ will be able to inherit the stroke style information coming from image y, as well as preserve its own content information.

F. The Leakage Problem

In the task of stroke style transfer, the dominant texture of painter's strokes usually occupies large portion in the style image. But in many paintings, some secondary textures like the signature or stamp also exists. These unwanted textures will be wrongly transferred in most global descriptor based methods. We show the reason here by taking Gram-matrix based style loss [7][12] as an example:

$$\mathcal{L}_{style}(a, b) = \parallel \sigma_l^\phi(a) - \sigma_l^\phi(b) \parallel_F^2 \quad (7)$$

Given two images a and b, let $\phi_l(\cdot)$ denotes the feature map on the l-th layer in a VGG16 network, and it has a shape of $C_l \times H_l \times W_l$. By reshaping $\phi_l(\cdot)$ into a matrix $\mathbf{B}$ of shape $C_l \times H_l W_l$, $\sigma_l^\phi(\cdot)$ denotes the Gram matrix of the feature map and it can be computed by $\sigma_l^\phi = \mathbf{B}\mathbf{B}^T / C_l H_l W_l$. The outer product of matrix $\mathbf{B}$ will inevitably constrain the synthetic image to maintain the same structure similarity with the style image, no matter the large area of dominant texture or the small area of secondary texture. Therefore, the secondary texture will be wrongly transferred into the synthetic images.

Our method can effectively eliminate the leakage phenomenon, because the operation of patch permutation will discard the structure information of the style image. The impact of secondary texture no longer takes effects since it only locates in very few patches and can be neglected in $D_p(\cdot)$. Meanwhile, the dominant texture that reflects stroke style will be retained in the final results since it locates nearly in every patches.

G. Discussions of Related Work

The work most closely related to P^2-GAN is the MGAN[8]. The major differences between MGAN and our method exist in three aspects: 1) We learn an encoder-decoder generator instead of the pre-trained VGG plus a fractional-strided convolution network in MGAN. Although two structures present the same image-feature-image dataflow, the pre-trained VGG in MGAN has limited the content image representation ability of the generator for its fixed weights once trained. 2) We cut the patches on the raw image rather than on the feature maps as in MGAN. Since the receptive field of each neuron on the $relu3\_1$ feature map is already big enough (over 30×30 pixels) and overlapping with each other, MGAN is apt to produce blurred images in style transfer. As contrast, our method directly learns from the patches in the original style image, therefore finer stroke effect can be transferred. 3) MGAN uses regular sampling, so that the patch number is inversely proportional to the patch size. Whereas, we use random sampling, and unlimited patch number can be expected theoretically. More patches means more training samples to solve the single image training problem.

Some other offline learning approaches like JohnsonNet[12], TextureNetIN[15] etc. can also be used for real time style transfer from single style image by exploiting the pre-trained feedforward network. However, our proposed P^2-GAN is different from these methods in at least two aspects: 1) We use the adversarial loss instead of the Gram-matrix based style loss as in these methods, so that the local style patterns such as "stroke" style can be better learned. 2) Our encoder-decoder network has been further optimized in terms of the computational efficiency and resource cost. In the experiments, comprehensive comparison with the above methods will be conducted.

H. Evaluation Criterion

Subjective quality evaluation in widely used in most style transfer methods[38][39]. In this paper, we propose a quantitative evaluation criterion based on the local texture similarity. Local Binary Pattern(LBP)[40] is used here as our texture descriptor. Given two images a and b, we firstly compute the LBP features $h_i \in \mathbb{R}^m$, $i = 1, ..., W$, $g_j \in \mathbb{R}^m$, $j = 1, ..., Z$ on randomly cropped patches from a and b respectively, where m denotes the dimension of feature vector on each patch, W and Z denote the number of patches in a and b. Our score function is so that defined as follows:

$$S(a, b) = \frac{1}{Z} \sum_{j=1}^{Z} \min_{i} \parallel h_i - g_j \parallel \tag{8}$$

In (8), the feature of each patch in b will try to find the most similar patch in a, in terms of the shortest LBP distance. By averaging all Z distances for each patch in b, the overall texture similarity between a and b can be measured. The LBP descriptor that performs on patch level rather than on whole image can effectively prevent the inference of content/structure information coming from the style image.

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

Fig. 7. Comparison of the texture evaluation criteria (7) and (8). (a) and (b) are synthetic images normalized to $[0, 1]$ with sharply different texture. Their style loss using (7) equals to 0, while the S score is 0.24. (c) and (d) are artistic images sharing the same texture. The $mean$ and std of (a) and (b) are $1/2$, $1/2$ and $1/\sqrt{2}$, 0 respectively. Their S score is 0.06, while the style loss is 491.3. All images are 256×256, with patch size 32×32, $W = 1000$ and $Z = 1000$.

To compare the proposed evaluation criterion (8) with the popular Gram matrix criterion, two typical pairs of images are illustrated in Fig.7. Firstly, a toy example is shown with two synthetic images (a) and (b). Although they possess sharply different visual appearances, their Gram matrices are exactly the same so that the style loss in (7) equals to 0. As contrast, our S score is 0.24 which has successfully distinguished their texture differences. The second example as in (c) and (d) is a pair of artistic images, which come from different portion of the same painting. In this case, our S score is 0.06 with large patch numbers. However, the style loss in (7) is 491.3 since their Gram matrices are different. The first example shows two irrelevant textures, while the second shows the same textures. The results manifest the advantage of our LBP-based criterion toward the texture measurement task. After all the Gram matrix doesn't equal to the texture itself in many cases.

IV. Experiments

A. Implementation details

We implement the generator following the architecture in Section III-D. It contains 3 sequential encoder layers, the output channels are configured as 32-64-128. The configuration of the generator is summarized in Table.I. Between the decoder layers and encoder layers, although more residual blocks may improve the synthesis quality for style image with complex stroke effect, it will also increase the computational cost. In our experiment, we only use one residual block in the generator and achieve comparable visual effect. After the residual blocks, 3 sequential decode layers are configured as 64-32-16. The output layer uses standard convolution to generate the rendered RGB image. Totally, our generator has about only 0.07M parameters, which is much less than [8] [12] and [15]. In addition, a $L = 3$ discriminator network is implemented in our experiments according to the rule in (3), as depicted in Table.II.

The P^2-GAN network proposed in this paper is trained on PASCAL VOC 2007 dataset[41]. N = 9.9k images are used as the content images to train our model. Each $\psi_i(y)$ consists of 24×24 patches with size of 9×9. We adopt the standard RMSPropOptimizer[42] to solve the objective function raised in (6). We set λ ranging from 1×10^{-6} to 1×10^{-5}. It takes about 20 minutes to train our model on the platform of single NVIDIA GTX 1080 GPU.

TABLE I
CONFIGURATIONS OF THE PROPOSED GENERATOR. WHERE *Conv dw* DENOTES DEPTHWISE CONVOLUTIONS, *Conv* DENOTES STANDARD CONVOLUTIONS.

Layer Type	Operator	Kernel Shape	Stride
Encode layer	Conv dw	$3 \times 3 \times 3$ dw	2
	Conv	$1 \times 1 \times 3 \times 32$	1
Encode layer	Conv dw	$3 \times 3 \times 32$ dw	2
	Conv	$1 \times 1 \times 32 \times 64$	1
Encode layer	Conv dw	$3 \times 3 \times 64$ dw	2
	Conv	$1 \times 1 \times 64 \times 128$	1
Residual Block	Conv dw	$3 \times 3 \times 128$ dw	1
	Conv	$1 \times 1 \times 128 \times 128$	1
Decode layer	Conv dw	$3 \times 3 \times 128$ dw	1
	Conv	$1 \times 1 \times 128 \times 128$	1
Skip Connection	Conv	$1 \times 1 \times 64 \times 128$	1
Decode layer	Conv dw	$3 \times 3 \times 128$ dw	1
	Conv	$1 \times 1 \times 128 \times 64$	1
Skip Connection	Conv	$1 \times 1 \times 64 \times 64$	1
Decode layer	Conv dw	$3 \times 3 \times 64$ dw	1
	Conv	$1 \times 1 \times 64 \times 32$	1
Output layer	Conv	$3 \times 3 \times 32 \times 3$	1

TABLE II
CONFIGURATION OF THE PROPOSED PATCH DISCRIMINATOR

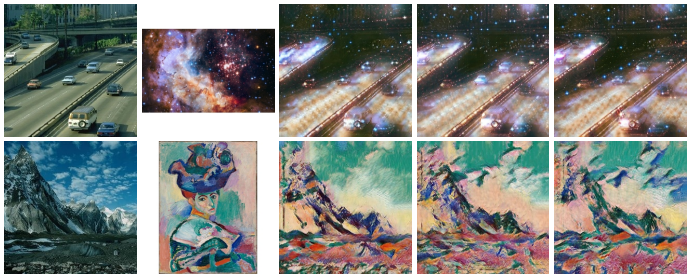
Operator	Kernel Shape	Stride	Activation
Conv	$3 \times 3 \times 3 \times 256$	3	LeakyReLU
Conv	$3 \times 3 \times 256 \times 512$	3	LeakyReLU
Conv	$1 \times 1 \times 512 \times 1$	1	Sigmoid
Pooling	Global Avg Pool	-	None

B. Parameters Configuration

In our objective function (6) , the hyper-parameter λ controls the balance between the content loss and the adversarial loss. We use about 9.9k content images to train our network by varying λ from 1×10^{-4} to 1×10^{-7}, and some typical results are shown in Fig.8. It can be found that bigger λ value can preserve more structure information of content images, while smaller λ tends to transfer more stroke texture from the style images. In the following experiments, we set λ between 1×10^{-5} and 1×10^{-6} for the tradeoff between stroke and structure.

(a) Content images (b) Style images (c) $\lambda = 1 \times 10^{-4}$ (d) $\lambda = 1 \times 10^{-5}$ (e) $\lambda = 1 \times 10^{-6}$ (f) $\lambda = 1 \times 10^{-7}$

Fig. 8. Impacts of the hyper-parameter λ in (6). Bigger λ value can preserve more structure information of content images, while smaller λ tends to transfer more stroke texture from the style images.



(a) Content images (b) Style images (c) 1 Residual Block (d) 2 Residual Blocks (e) 3 Residual Blocks

Fig. 9. Impact of changing the residual block numbers. We set the number of residual blocks in the generator network with 1,2 and 3. The corresponding style transfer results in (c), (d) and (e) show similar and satisfactory visual qualities.

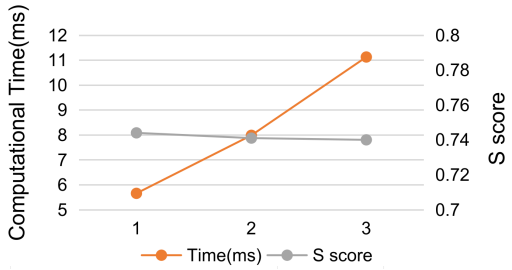
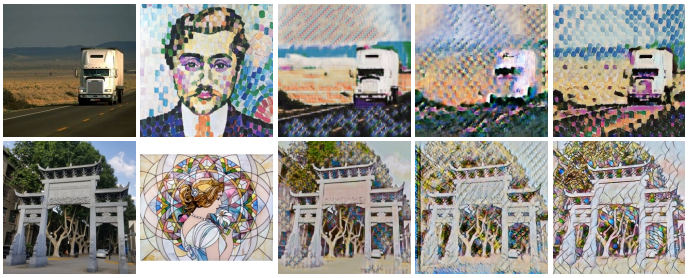


Fig. 10. Quantitative analysis of changing the residual block numbers. With 1, 2 and 3 residual blocks respectively, the computational time increases linearly, while the image qualities measured by the S score in (8) remain approximately the same.

In most network architectures that adopt ResNet as the backbone, usually several (i.e. eight in ResNet18) residual blocks are embedded to ensure the network precision. In our generator $G(\cdot)$, we have also tested the impact of using different number of residual blocks. 50 images with resolution of 256×256 are stylized using our generator with 1,2 and 3 residual blocks respectively. Some typical results are shown in Fig.9, and the averaged time and quality measurements are summarized in Fig.10. The quality of stylization is measured by criterion in (8). Both the subjective feeling and quantitative results of image qualities show negligible differences w.r.t. residual block number. Their visual qualities are all similar and satisfactory. However, the averaged computational time increases from 5.6ms, 7.9ms to 11.1ms respectively. Therefore, in all the following experiments, we set the block number to 1 as the best choice.

C. Ablation Experiments

To verify the effectiveness of the proposed network architecture, we conduct the ablation experiments in three ways: a) only remove the patch permutation module, b) remove both the patch permutation and the patch descriminator modules, c) permit arbitrary patch descriminator that doesn't comply with (3).



(a) Content images (b) Style images (c) Without any patch operation (d)Without patch Permutation (e) P^2-GAN

Fig. 11. Results of the ablation experiments. In (c), the modules of patch permutation and patch discriminator are all removed from our network. In (d), only the the patch permutation module is removed. Their qualities of generated images all degraded compared to the P^2-GAN.

(a) Content images (b) Style images (c) stride = 2, kernel size = 3, patch size = 9 (d) stride = kernel size = 3, patch size = 11 (e) stride = kernel size = 3, patch size = 9

Fig. 12. Ablation experimental results using arbitrary patch discriminators. Parameters in (e) strictly subject to our constraint, and seamless transferred image can be achieved. Even subtle changes on theses parameters will incur severe checkerboard artifacts on the transferred images as in (c) and (d).

For situation a) and b), since we have only one style image and plenty of content images to train the network, the adversarial learning doesn't take effects to generate the expected stroke texture due to the lacking of training sample problem. In situation b), the network degenerates to a standard GAN , and the generator can easily fool the discriminator even by the exact content image alike synthesis. Some style transfer results using these degenerated networks are shown in Fig.11 (c)(d). It can be seen that the stroke style in Fig.11 (b) has ignorable impacts on the final results, while the content images in Fig.11 (a) dominate the transferring effects.

For situation c), sufficient number of training samples (576 samples from single style image in our experiments) were generated by the patch permutation operation. The results are shown in Fig.12. In each row, three D_p with the same architecture but different parametric relationships are trained from the same single style images y as in (b). The patch size varies from 9×9 to 11×11 to generate patch permuted training images $\mathcal{Y}_\psi$, and small changes on the kernel size and the stride by 2 to 3 are also tested. It can be observed that when the kernel size is not equal to the value of stride (as in (c)), or the kernel size is not divisible by patch size (as in (d)), severe checkerboard artifacts will appear on the transferred images. The reason is that in these cases the convolutional operation will be performed inevitably across adjacent patches where discontinuous texture exists. On the contrary, D_p with parameters that strictly subject to (3) is invulnerable to patch permuted training images $\mathcal{Y}_\psi$, and natural style can be transferred seamlessly. Even subtle change (only one pixel) on the kernel size will yield remarkable difference on the transferred images. Therefore, the proposed constraint of parameters in D_p is essential to our patch permutation method.

D. Subjective Evaluation of Style Transfer Quality

Subjective evaluation is a commonly adopted manner in most style transfer works. In this section, two typical Gram-matrix based algorithms JohnsonNet[12], TextureNetIN[15] and a GAN based transfer algorithm MGAN[8] are compared with our method. For each method, 50 content images are transferred into 5 different styles, including *Mountain No.2* by Jay DeFeo, *Starry Night* by Van Gogh, *Portrait* by Jean Metzinger, *Helicopter* by Leonardo da Vinci and *Woman With a Hat* by Henri Matisse. Some typical results are shown in Figure.13 and 14.

It can be observed that MGAN is apt to produce blurred images where most edge and detail information have lost. This drawback can be attributed to the patch-wise operation conducted in the feature maps whose receptive fields are too big in the raw image. The other 3 algorithms can produce visual satisfactory and clean results. Especially, our method can generate most vivid stroke style effect, as shown in Fig.13 (g) enlarged from the red boxes in (f).

More "leakage" examples are demonstrated in Fig.14. In these results from JohnsonNet and TextureNetIN, some local structures in the style images, i.e. the calligraphy or signature have misled unexpected artifacts in the stylized results as shown in the blue boxes. The global descriptor in Gram matrix based approaches will inevitably transfer partial structure information from the style image. As contrary, this problem can be overcome by the patch-wise strategy in MGAN and our method.

E. Quantitative Evaluation of Style Transfer Quality

Given two images a and b, the content loss in (9) and the style loss in (7) are two popular quantitative evaluation criterion in some style transfer works[7][12]:

$$\mathcal{L}_{content}(a,b) = \frac{1}{C_l H_l W_l} \parallel \phi_l(a) - \phi_l(b) \parallel_2^2 \qquad (9)$$

In the experiment of this section, the above evaluation functions are computed using the feature maps on layer $relu3\_3$ in the VGG16 network. The similarities between the re-renderings and their original content/style images using the same experimental configuration as above are computed. The averaged content loss and style loss of totally 250 generated images from each of 4 different methods are shown in Figure.15. It can be found that JohnsonNet and TextureNetIN have higher similarities w.r.t. their original images compared to MGAN and our method, which means a better transfer quality in terms of the pre-defined loss functions. However, since the objective functions in JohnsonNet and TextureNetIN have the similar forms with (7) and (9), the optimization process will

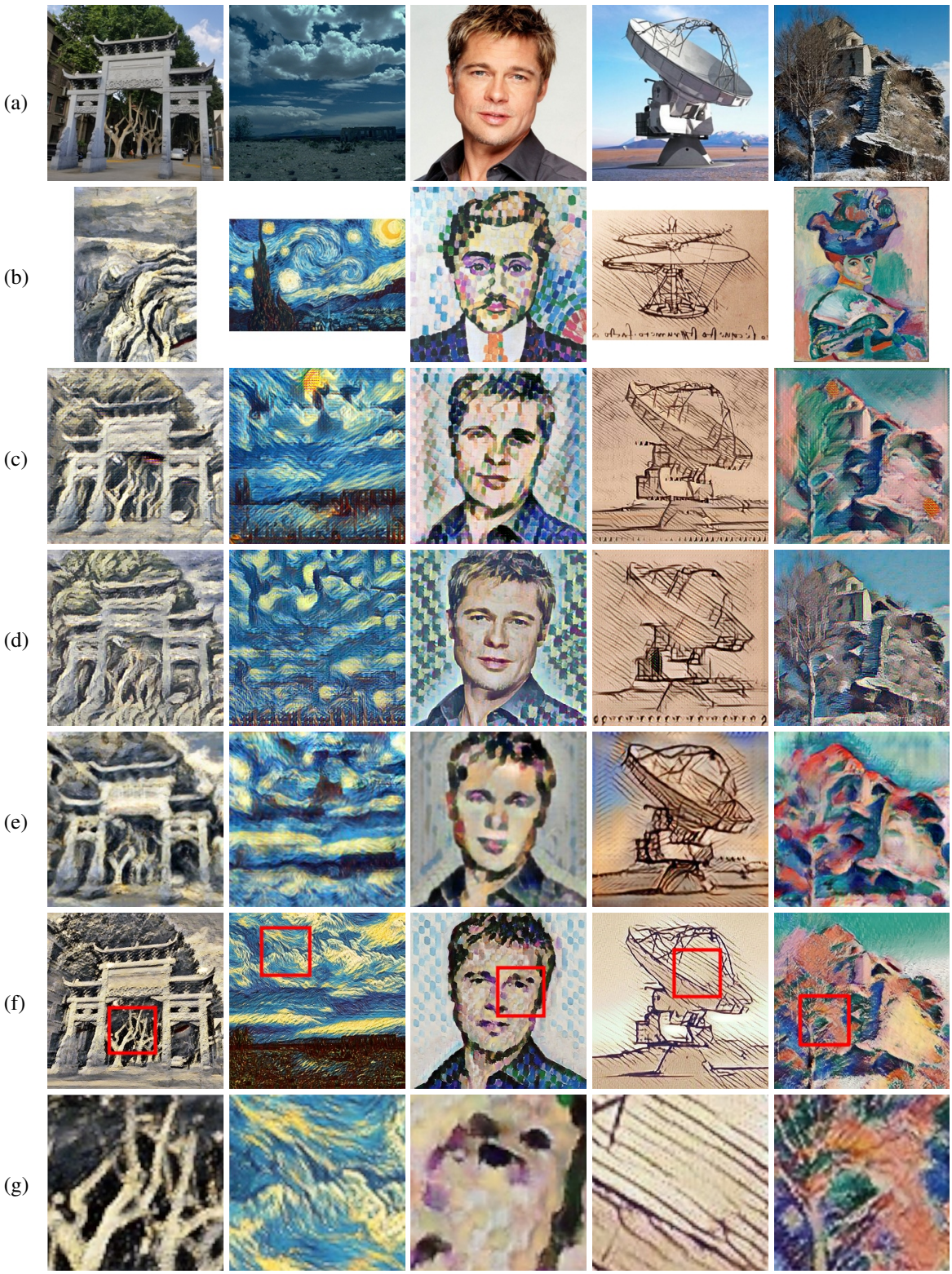
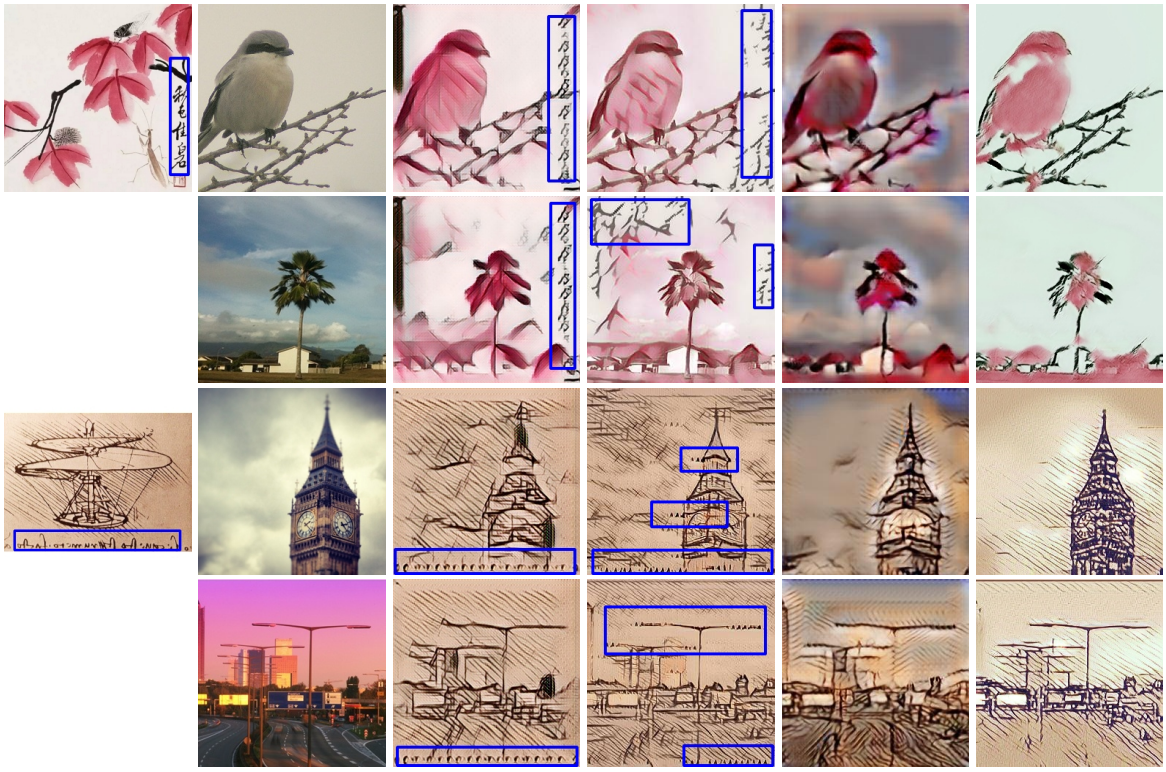


Fig. 13. Comparison of style transfer using single style image by 4 different methods. (a) and (b) denote the content and style images respectively. (c), (d), (e) and (f) denote the style transfer results by JohnsonNet[12], TextureNetIN[15], MGAN[8] and our method respectively. In (g), the vivid stroke transfer effects of our method can be observed from enlarged red boxes of (f).



(a) Style images (b) Content images (c) JohnsonNet[12] (d) TextureNetIN[15] (e) MGAN [8] (f) Ours

Fig. 14. Some examples of the structure "leakage" problem. Row 1 and 2 are results from the first style image, and row 3, 4 correspond to the second. The blue boxes indicate that the global descriptor of Gram matrix in JohnsonNet and TextureNetIN may incur apparent "leakage" effect of structure information from the style image, while this problem can be overcome by the patch-wise strategy in MGAN and our method.

guarantee their results to converge to the lower error scores. The Gram matrix based style loss and Euclidean distance based content loss are both global descriptors, therefore, the ability of this criterion on describing the local stroke style that our paper emphasized is very limited.

To compensate the limitation of the criterion above, we also quantitatively compared the transfer quality for all 4 methods using the criterion in (8). The proposed criterion is based on a patch-wise descriptor. Compared to the traditional evaluations in (7) and (9), it has stronger represent ability on local texture as well as better avoiding the inference of global structure information from the style image. The patch size is 32×32, $W = 1000$, and $Z = 1000$. The mean values and standard deviation of S score from each method are plotted in Figure.16.

The results showed that the MGAN has the highest S score (0.963 ± 0.37), which means the lowest style similarity between the transferred image and the given one. JohnsonNet and TextureNetIN present the better similarities, that is, 0.790 ± 0.33 and 0.776 ± 0.35 respectively. In contrast, our method has the lowest S score (0.756 ± 0.22), which indicates the highest texture similarity compared with the given style image. Due to the patch-wise convolution, our method emphasizes more on transferring local texture (stroke style) rather than the global description as done in JohnsonNet and TextureNetIN. Therefore, our method can better avoid the inference of structure information from the training image.

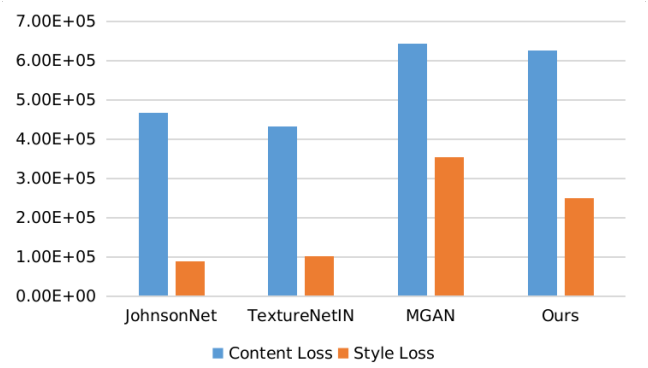


Fig. 15. Quantitatively evaluation of the style transfer qualities on 4 different methods using the averaged content loss and style loss as in (7) and (9).

F. Computational Performance

To evaluate the online computational performance of the proposed method, 1000 content images with 4 different resolutions are tested for the style transfer task using the pre-trained networks from 4 different algorithms. Our computational platform is an Intel Core i7 4790 CPU, 16G memory, with a NVIDIA GTX 1080 GPU. For each network, their averaged time consumptions per inference of image are summarized in Table.III. Specifically, to further evaluate their performances on pure CPU platform, we have also banned the availability of GPU and repeated the experiments above. The averaged

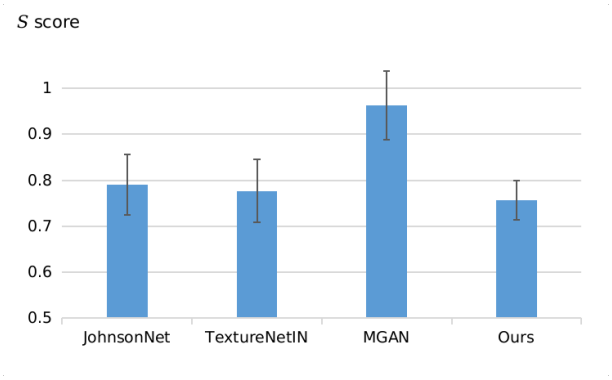


Fig. 16. Quantitatively evaluation of the transfer qualities on 4 different methods using the proposed LBP descriptor based criterion as in (8).

online inference speeds as well as the averaged memory consumptions are summarized in Table.IV. and V. respectively.

It can be found that for both GPU and pure CPU environments, the time consumptions for all methods are proportional to the image size. For all methods, the inference speed on GPU is 50 ˜150 times faster than on pure CPU. On both platforms, our network overwhelms all 3 others with the inference speed of 2 ˜3 times faster and the memory consumption of 2 ˜30 times saving. These improvements have verified the effectiveness and efficiency of our discriminator/generator network architecture design. Specifically, on processing the HD images such as 1024×1024, our network only costs about 1s CPU time without GPU support and less than 500M memory. This result shows the potential of real-time style transfer application using our method on the light-weight embedding platform.

TABLE III
COMPARISON OF AVERAGED COMPUTATIONAL TIME (IN MS) PER STYLE TRANSFERRING ON GPU PLATFORM USING 4 DIFFERENT METHODS. THE SHORTEST TIME ON EACH IMAGE RESOLUTION IS MARKED IN **BOLD**.

Method \ Image size	128	256	512	1024
JohnsonNet[12]	5.1	9.2	29.2	128
TextureNetIN[15]	7.9	12.7	35.1	137
MGAN[8]	8.2	13.2	40.7	-
Ours	**3.6**	**5.6**	**13.2**	**46.2**

TABLE IV
COMPARISON OF AVERAGED COMPUTATIONAL TIME (IN MS) PER STYLE TRANSFERRING WITHOUT GPU SUPPORT USING 4 DIFFERENT METHODS. THE SHORTEST TIME ON EACH IMAGE RESOLUTION IS MARKED IN **BOLD**.

Method \ Image size	128	256	512	1024
JohnsonNet[12]	29.73	109.7	454.8	1812
TextureNetIN[15]	127.3	484.9	1974	8213
MGAN[8]	133.2	552.7	2312	9268
Ours	**18.44**	**71.26**	**275.3**	**1072**

G. Video Style Transfer

Finally, the proposed method has been tested on the video style transfer task. 5 HD videos with resolution of 1280-by-720 are used here. For simplicity, each frame of these videos is computed as an independent content image using the offline trained generator network, and the techniques that take into consideration of inter-frame relationship such as temporal regularization have not been utilized. Some results of single frame randomly selected from the transferred videos are shown in Fig.17.

It can be found that our method can produce equally good results for consecutive frames in videos without any change as in the image style transfer experiments. Specifically, our method achieved an amazing online speed of 28Hz for re-rendering these HD videos using the platform above with GPU support, which implies the potential of our method on faster video style transfer applications.

TABLE V
COMPARISON OF AVERAGED MEMORY CONSUMPTION (IN MB) PER STYLE TRANSFERRING WITHOUT GPU SUPPORT USING 4 DIFFERENT METHODS. THE LOWEST MEMORY COST ON EACH IMAGE RESOLUTION IS MARKED IN **BOLD**.

Method \ Image size	128	256	512	1024
JohnsonNet[12]	190	330	360	550
TextureNetIN[15]	200	570	2200	9000
MGAN[8]	200	600	2200	9100
Ours	**180**	**300**	**340**	**390**

V. CONCLUSION

In this paper, a novel Patch Permutation GAN (P^2-GAN) network that can be efficiently trained by a single style image for the style transfer task is proposed. We use patch permutation to generate multiple training samples, and use a novel patch discriminator to process the patch-wise images and the natural images simultaneously. Our method provides the possibility to precisely simulate the expected stroke style from the customer designated single image, and avoids the difficulty of collecting multiple training images with the same style. Both subjective and quantitative experimental results verified that our method overwhelmed most state-of-the-arts single image style transfer methods in terms of the rendering quality and the computational efficiency.

VI. ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China (No.2018AAA0102504), and the Provincial Key Laboratory Program of Shaanxi (No.2013SZS12-K01).

REFERENCES

[1] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," *International Conference on Computer Graphics and Interactive Techniques*, pp. 341–346, 2001.
[2] N. Ashikhmin, "Fast texture transfer," *IEEE Computer Graphics and Applications*, vol. 23, no. 4, pp. 38–43, 2003.
[3] B.-K. Kim, G. Kim, and S.-Y. Lee, "Style-controlled synthesis of clothing segments for fashion image manipulation," *IEEE Transactions on Multimedia*, vol. 22, no. 2, pp. 298–310, 2019.
[4] Y. Chen, Y. Lai, and Y. Liu, "CartoonGAN: Generative adversarial networks for photo cartoonization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9465–9474.

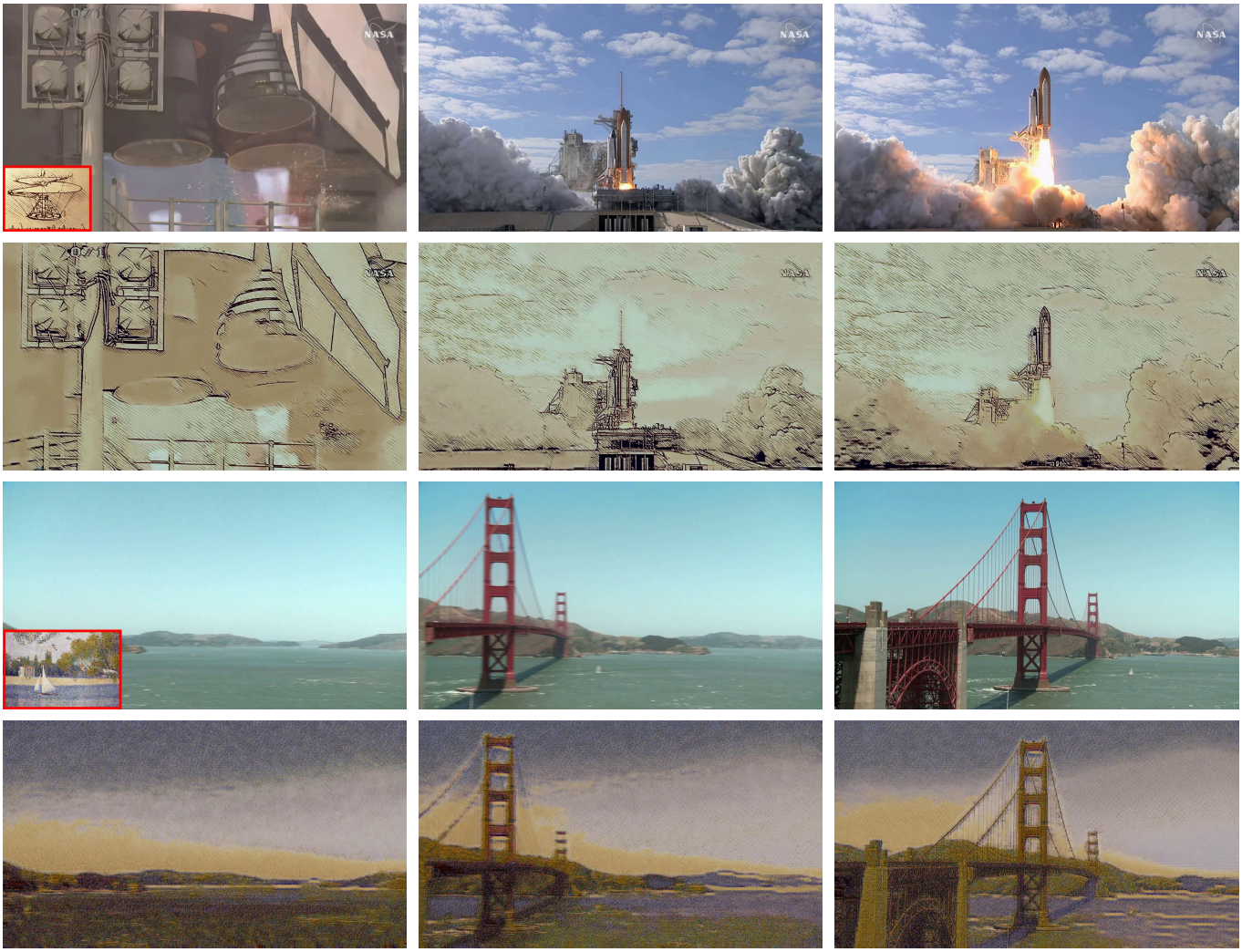


Fig. 17. Some frames of video stylization using the proposed method. The 1^{st} and 3^{rd} rows are the content frames extracted from two raw videos, and the corresponding style images are attached in the red boxes. The 2^{nd} and 4^{th} rows are the stylized results respectively. With the impressive details of style strokes, our method can stylize 1280-by-720 HD videos at 28Hz.

[5] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, "Neural style transfer: A review." *IEEE Transactions on Visualization And Computer Graphics*, 2019.

[6] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv: 1508.06576*, 2015.

[7] ——, "Image style transfer using convolutional neural networks." in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2414 – 2423.

[8] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *European Conference on Computer Vision*, 2016, pp. 702–716.

[9] J. Liu, W. Yang, X. Sun, and W. Zeng, "Photo stylistic brush: Robust style transfer via superpixel-based bipartite graph," *IEEE Transactions on Multimedia*, vol. 20, no. 7, pp. 1724–1737, 2018.

[10] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang, "Visual attribute transfer through deep image analogy." *ACM Transactions on Graphics*, vol. 36, no. 4, 2017.

[11] S. Gu, C. Chen, J. Liao, and L. Yuan, "Arbitrary style transfer with deep feature reshuffle." in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8222–8231.

[12] J. Johnson, A. Alahi, and F. F. Li, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*, 2016, pp. 694–711.

[13] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, "StyleBank: An explicit representation for neural image style transfer," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2770–2779.

[14] A. Sanakoyeu, D. Kotovenko, S. Lang, and B. Ommer, "A style-aware content loss for real-time hd style transfer," in *European Conference on Computer Vision*, 2018, pp. 715–731.

[15] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis." in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4105–4113.

[16] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. S. Lempitsky, "Texture networks: Feed-forward synthesis of textures and stylized images," in *Proceedings of the 33nd International Conference on Machine Learning*, M. Balcan and K. Q. Weinberger, Eds., vol. 48, 2016, pp. 1349–1357.

[17] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks." in *IEEE International Conference on Computer Vision*, 2017, pp. 2242–2251.

[18] X. Chen, C. Xu, X. Yang, L. Song, and D. Tao, "Gated-GAN: Adversarial gated networks for multi-collection style transfer," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 546–560, 2019.

[19] P. Wilmot, E. Risser, and C. Barnes, "Stable and controllable neural texture synthesis and style transfer using histogram losses," *CoRR*, vol. abs/1701.08893, 2017. [Online]. Available: http://arxiv.org/abs/1701.08893

[20] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: a randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, 2009.

[21] C. Li and M. Wand, "Combining markov random fields and convolutional neural networks for image synthesis." in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2479 – 2486.

[22] Z. Yi, H. R. Zhang, P. Tan, and M. Gong, "DualGAN: Unsupervised dual learning for image-to-image translation," in *IEEE International Conference on Computer Vision, ICCV*, 2017, pp. 2868–2876.

[23] Z. Li, C. Deng, E. Yang, and D. Tao, "Staged sketch-to-image synthesis via semi-supervised generative adversarial networks," *IEEE Trans. Multim.*, vol. 23, pp. 2694–2705, 2021.

[24] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2018, pp. 8789–8797.

[25] Y. Guo, Q. Chen, J. Chen, Q. Wu, Q. Shi, and M. Tan, "Auto-embedding generative adversarial networks for high resolution image synthesis," *IEEE Trans. Multim.*, vol. 21, no. 11, pp. 2726–2737, 2019.

[26] X. Guo, R. Nie, J. Cao, D. Zhou, L. Mei, and K. He, "FuseGAN: Learning to fuse multi-focus image via conditional generative adversarial network," *IEEE Trans. Multim.*, vol. 21, no. 8, pp. 1982–1996, 2019.

[27] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *CoRR*, vol. abs/1411.1784, 2014.

[28] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Pixel-wise crowd understanding via synthetic data," *Int. J. Comput. Vis.*, vol. 129, no. 1, pp. 225–245, 2021.

[29] Q. Wang, J. Gao, and X. Li, "Weakly supervised adversarial domain adaptation for semantic segmentation in urban scenes," *IEEE Trans. Image Process.*, vol. 28, no. 9, pp. 4376–4386, 2019.

[30] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks." in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5967–5976.

[31] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-assisted Intervention*, 2015, pp. 234–241.

[32] Y. Liu, W. Chen, L. Liu, and M. S. Lew, "SwapGAN: A multistage generative approach for person-to-person fashion style transfer," *IEEE Transactions on Multimedia*, vol. 21, no. 9, pp. 2209–2222, 2019.

[33] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

[34] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2017, pp. 1800–1807.

[35] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, 2016. [Online]. Available: http://distill.pub/2016/deconv-checkerboard

[36] L. Galteri, L. Seidenari, M. Bertini, and A. D. Bimbo, "Deep universal generative adversarial compression artifact removal," *IEEE Transactions on Multimedia*, vol. 21, no. 8, pp. 2131–2145, 2019.

[37] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge." *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211 – 252, 2015.

[38] X. Li, S. Liu, J. Kautz, and M.-H. Yang, "Learning linear transformations for fast image and video style transfer," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3809–3817.

[39] Y. Yao, J. Ren, X. Xie, W. Liu, Y.-J. Liu, and J. Wang, "Attention-aware multi-stroke style transfer," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1467–1475.

[40] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[41] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge." *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[42] T. Tieleman and G. Hinton., "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude." *COURSERA: Neural Networks for Machine Learning, 4(2)*, 2012.

Zhentan Zheng received the B.E. degree from the Henan Agricultural University, China, in 2017. He is currently pursuing MA.Eng. degree Xi'an Jiaotong University, China. His research interests include computer vision and deep learning.

Jianyi Liu received the B.Eng. and the M.Eng. degrees in information and communication engineering from Xi'an Jiaotong University, China in 1998 and 2001 respectively, and the Ph.D. degree in control science and engineering from Xi'an Jiaotong, China in 2009. He is currently an associate professor in college of artificial intelligence, Xi'an Jiaotong University, China. His research interests include image processing, pattern recognition and machine learning.

Nanning Zheng graduated from the Department of Electrical Engineering, and received the ME degree from Xi'an Jiaotong University, China in 1975 and 1981 respectively. Received his Ph. D. degree in electrical engineering from Keio University, Japan, in 1985. He is currently a professor and the director of the Institute of Artificial Intelligence and Robotics at Xi'an Jiaotong University. His research interests include computer vision, pattern recognition, and hardware implementation of intelligent systems. He is the Chinese representative on the Governing Board of the International Association for Pattern Recognition. He is a member of the Chinese Academy Engineering since 1999. He is a fellow of the IEEE.