Raphael Painting Authentication and Style Transfer

Anonymous Author(s)
Affiliation
Address
email

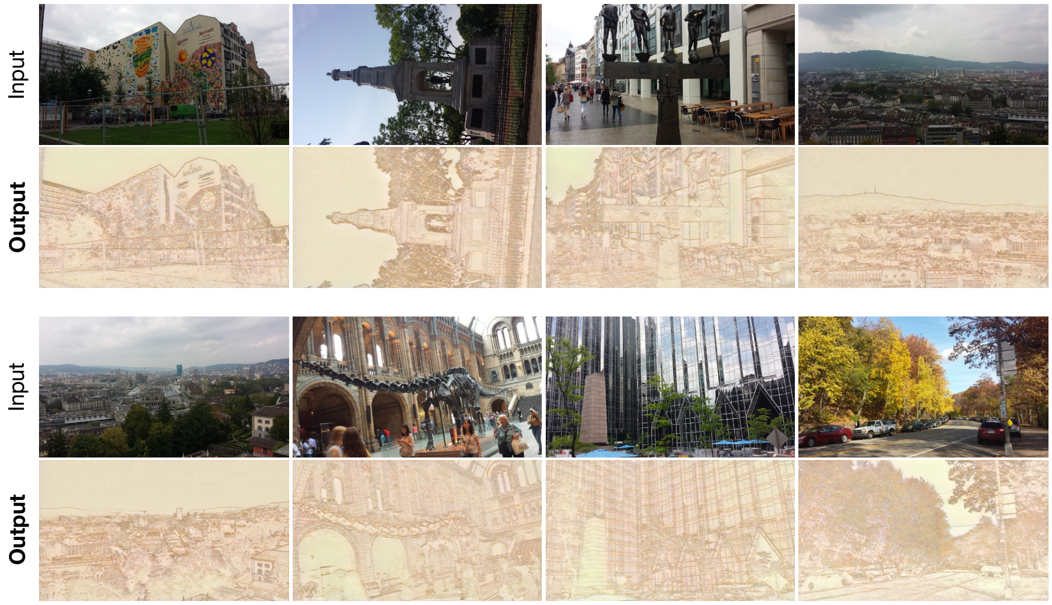


Figure 1: A quick view of our results in style transfer (you can zoom to see larger images)

Abstract

Painting authentication and image style transfer are two essential but difficult image processing tasks. The major difficulty is to extract efficient features that represent a painting's style. Here, we introduce two feature extracting methods to the first task: 1) a geometric tight frame with three statistics; 2) a style representation derived from a pre-trained CNN. Furthermore, we apply a forward feature selection algorithm and get satisfying results in authentication of Raphael's paintings.

In the style transfer task, we implement Gatys' Nerual Algorithm of Artistic Style and improve it by preprocessing content image, such as contour extraction, edge enhancement and extracting the painter's style from 12 genuine pictures. We also apply the Encoder-Decoder techniques like Discovery GAN. Our preprocessing techniques greatly improve the quality of output images, as they match more to Raphael's sketch style and our attempt of applying Discovery GAN in this task is also successful.

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

1 Raphael Painting Authentication

1.1 Introduction

Art authentication is always a hard problem even for those experts of certain artist and worse yet, it sometimes costs a great amount of money, which may surpass the value of the painting itself, to apply high-techs, such as Isotope tracing and other chemical analysis.

Fortunately, with the rapid development of AI, especially in machine learning and deep learning, it is possible to do such authentication through relating algorithms without human interaction.

Related work Generally, there are two main methodologies in this field. One is stroke based [Elgammal et al., 2017], the other focuses on more general features [Liu et al., 2016, Li et al., 2017]. Considering the particular painting style of Raphael within our data, we will mainly follow the latter one in this paper.

1.2 Data

The data set is provided by Prof.Yang Wang, HKUST, which consists of high resolution scans of 28 paintings. The picture sizes are different from each other, ranging from 1192*748 to 6326*4457 pixels. Among the 28 paintings, 12 have been classified as genuine, 9 have been known to be forgeries, and remaining 6 are currently questioned by experts.

1.2.1 Data preprocessing

Note that some images of the raw data are stored as tiff files (images 2/3/4/5/6/8/9/24/27/28) while others are jpg files (images 11/12/13/14/15/16/17/18/19/21/22), the problem is that tiff files contains four channels which are RGBA (i.e. Red/Green/Blue/Alpha) and jpg files only contains the first three channels. However, after normalization, we also noticed that, in each image, every entry of the alpha channel equals to one. Hence, we assume it is safe to draw a conclusion that alpha channel doesn't affect much in this task.

After a quick skim through the dataset, we make a note about the boundary of the paintings here. Since almost every painting is centered, intuitively we would agree that the edges of the canvas in the paintings may not be useful information for art authentication, and hence we have excluded these edges in our numerical experiments. More precisely, for each painting in the dataset, we crop off 100 pixels from its four sides, and use only the interior of the image in our numerical tests.

Talking about doing data augmentation, in order to make up for the small size of dataset, we use two methods to cut images, one is to cut raw image into small patches by specified pixel size (e.g. 227 × 227), another is to cut it by specified number of patches (e.g. 16 patches per image). Also, for the purpose of avoiding situations like we crop a component object into two parts, we allowed 20% overlapping area when cropping. The procedure mentioned above is shown in Figure 2.

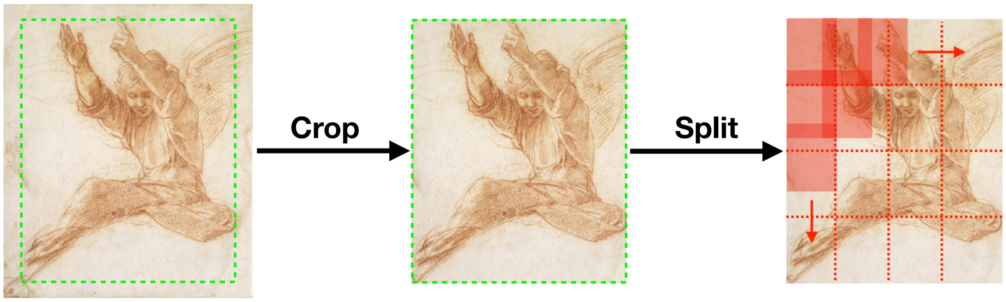


Figure 2: Data preprocessing example

To apply the geometric tight frame of Li et al. [2010] and Li et al. [2011], we first turn the pictures into grey-scale images with one channel.

Normally, the tight frame is applied on grey-scale images with format ‘uint8’, of which any pixel ranges from 0 to 255. Here, we are curious about those constant coefficients of the tight frame and thus, we try on another format of the images, ‘float’, of which any pixel ranges from 0-1. The latter one performs better after forward stage-wise feature selection.

1.3 Basic Method

1.3.1 Feature extraction

The feature extraction procedure here is the same as that of Liu et al. [2016].

What’s more, we find that because of the specificity of our training data, the image matrix is ‘sparse’ to some extent. And this problem is even worse if we apply the geometric tight frame. That is to say a certain amount of features are close to zero, which is a nightmare to some classifiers. So, we apply standardization to the features (rescale to mean 0 and standard error 1).

In summary, together with the data preprocessing procedures, we will have mainly two groups of features: patch or no-patch, each of which has four type of features: ‘uint8’ or ‘float’ and standardization or not. Note that re-encode an image from uint8 to float is the same as normalization. So the name of these different features are just like what’s shown in Figure 3 & 4.

1.3.2 Training procedure

The main idea of this classification task is outlier detection [Liu et al., 2016]. With the intuition that the genuine ones will ‘gather together’ while the fake ones would be more ‘far away’. So the classifier is built mainly on the Euclidian distance, which is also called ‘2-norm’.

The main procedure here could be summarized as three steps: (more details please refer to Liu et al. [2016])

- get the ‘genuine center’ of training data
- get the threshold of the distance from one sample to the center
- label the validation data with the center and the threshold

Note that our training data is limited (only 21 pictures without augmentation). So here we simply apply the leave-one-out cross validation(LOOCV) procedure to avoid overfitting problem.

1.3.3 Feature selection

In previous part, we induce totally 54 features for each grey-scale image. Intuitively, there would be some noise within so many features. That is to say, fewer features can perform better for this classification task.

Speaking of feature selection algorithms, there are also two main methodologies, forward selection and backward selection. For computational efficiency, we choose the forward one. While for the rank boosting algorithm proposed by Liu et al. [2016], we think it may not suit the problem well enough. A good reason is that *artistic authentication* is not a recognition problem, of which the dominant features can do most of the job. But for authentication, sometimes a group of sub-dominant features do help.

So, we propose a more direct forward selection algorithm based on the LOOCV. (see Table 1)

The results of feature selection on different type of features can be seen in Figure 3, in which we can clearly conclude that there are much noise in all 54 features because as the number of features grows, the performance generally goes down.

1.4 Other Potential Methods

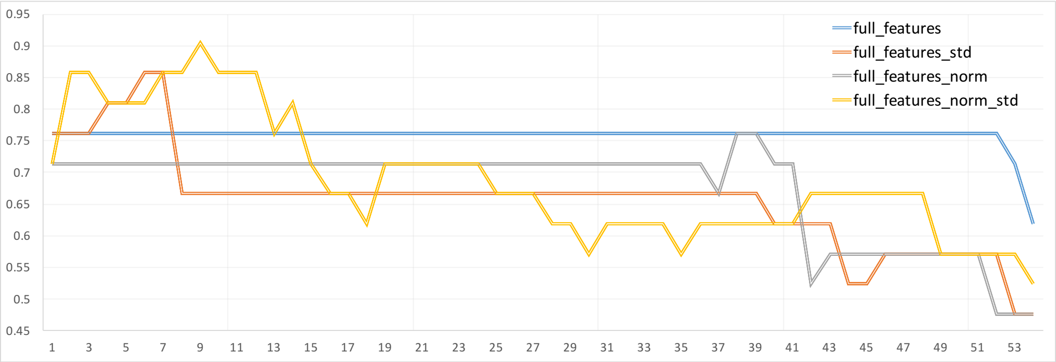
1.4.1 Classic classifiers

Here we try other classic classifiers, such as KNN, SVM and decision tree, during which we introduce data augmentation by cutting the original images into 16 patches. (see section 1.2.1) We have a good

Table 1: forward selection algorithm

```
Input   current_set ← $\emptyset$
        $X$ ← (21,54) data matrix
loop    $i$ from 1 to 54:
        remain_set ← $\{0, 1, 2, \cdots, 53\} - current\_set$
        loop j in remain_set:
            $F_j \leftarrow current\_set \cup \{j\}$
            $p_j \leftarrow$ the performance by LOOCV using $F_j$
        end
        $j^* \leftarrow argmax_j\{p_j\}$
        $P_i \leftarrow p_{j^*}$
        current_set ← $current\_set \cup \{j^*\}$
end
        best number of features ← $argmax_i\{P_i\}$
        and it's easy to get the corresponding features.
```

Figure 3: Performance of different number of features



reason that these classifiers' performance will grow with the increase in the volume of data. The results of this part are in Figure 4.

1.4.2 Style Feature Method

Introduction It is widely acknowledged that Convolutional Neural Networks(CNN) can capture local information of a picture like brushwork, textures which we generally call style. Gatys et al. [2016] introduced a neural algorithm of artistic style that can render a the semantic content of an image into a different style with CNN. The main idea is that a pre-trained CNN (e.g VGG-19) can extract high-dimension features in a given picture. The model introduced by Gatys et al. [2016] uses features generated by CNN filters as content representation, the Gram matrixes as style representation and also uses Gradient Descent to learn a picture with small content loss and style loss between output and the target content and style.

Model In the task of authentication of paintings, it is natural to implement this idea in the process of feature extraction because we want to determine based on information of the painter's style rather than the painting's content. We believe that Gram matrixes, which consist of the correlation between the different filter responses, can remove the content information, because every point in a Gram matrix represent not a local but a global feature of the picture. Thus, they can dig deep into the picture's information. More Details will be shown in the latter part of style transfer.

Like Gatys et al. [2016], we use VGG-19 as our pre-trained network, and use Gram matrixes 'conv1', 'conv2', 'conv3', 'conv4' and 'conv5' as style features. We reshape the Gram matrixes into a vector and concatenate them into a long vector, which we use as style features. The network architecture is shown in Figure 5.

Figure 4: Best performance of different classic model

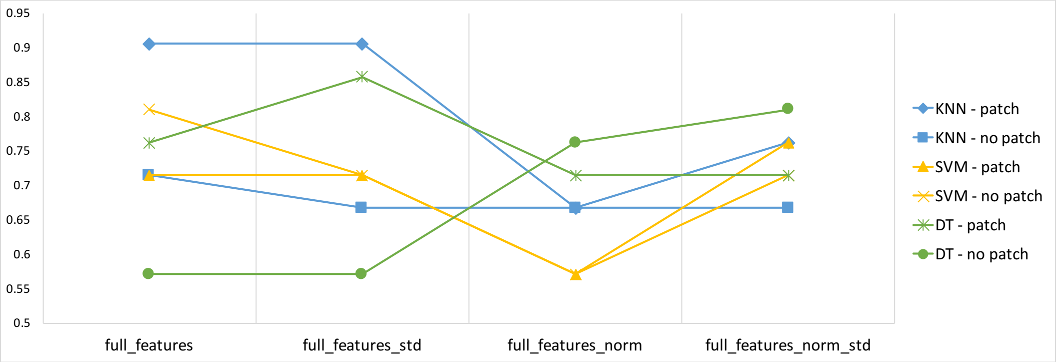
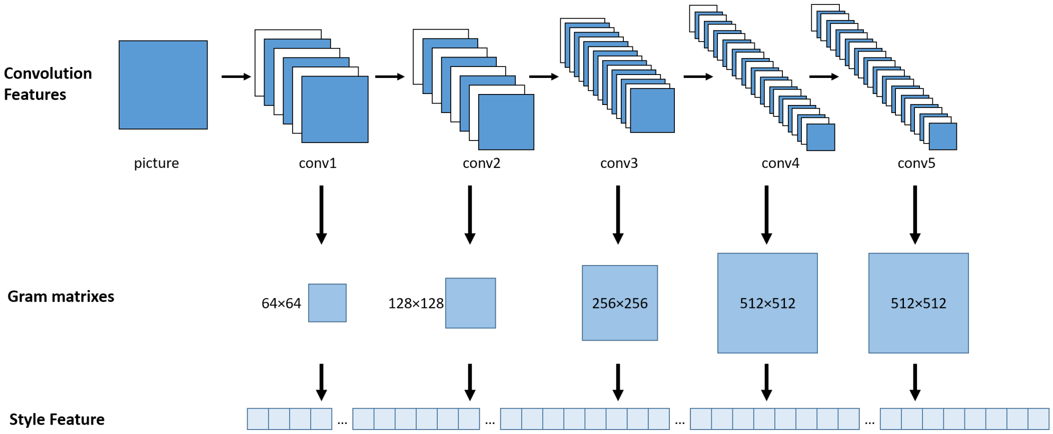


Figure 5: Process to extract style features



Experiments Then we apply K-Nearest Neighours (KNN), Support Vector Machine (SVM) and Decision Tree classifiers with leave-one-out validation. Due to memory constraint, we resize the picture to 256/512/1024 pixels. We divide the picture into 16 patches in KNN classifier to augment training data. The result is shown in Table 2. We find that since the dimension is too high (above 100,000), SVM is not applicable. KNN performs better on features got from low-definition pictures with 16 seperated patches and Decision Tree performs better on features got from low-definition pictures.

Table 2: Leave-one-out result with style features

Feature Extraction	Model	TPR	TNR	Classification Accuracy
Style Features-256	KNN	0.833	0.889	**0.857**
	SVM	1.000	0.000	0.571
	Decision Tree	0.667	0.556	0.619
Style Features-512	KNN	1	0.333	0.714
	SVM	1.000	0.000	0.571
	Decision Tree	0.833	0.889	**0.857**
Style Features-1024	KNN	0.667	0.444	0.571
	SVM	1.000	0.000	0.571
	Decision Tree	0.833	0.889	**0.857**

Predictions Upon our style-features models, we give our prediction to the 7 pictures remain disputed (Pic1/7/10/20/23/25/26). We pick three Models performed best in validation. We predict that Picture 10/25 are genuine, and Picture 1/7/20 are counterfeit. For picture 23/26, our results varies, so we have reservations about it. The result is shown in Table 3 (0 means counterfeit and 1 means genuine).

Table 3: Predictions upon style-features models

	Pic1	Pic7	Pic10	Pic20	Pic23	Pic25	Pic26
Style Features-256 with KNN	0	0	1	0	0	1	0
Style Features-512 with Decision Tree	0	0	1	0	1	1	1
Style Features-1024 with Decision Tree	0	0	1	0	1	1	0

1.5 Baseline and Main results

Baseline Our baseline mainly comes from the work of Li et al. [2017], for their dataset is the same.

Table 4: Our final results of different models with different feature extractions

Feature Extraction	Model	TP	TN	Classification Accuracy
Tight Frame	Forward Stage-wise	83.3%	100%	**90.5%**
	SVM	83.3%	77.8%	81.0%
	Decision Tree	83.3%	88.9%	85.7%
	KNN	91.7%	88.9%	**90.5%**
Style Features	Decision Tree	83.3%	88.9%	85.7%
	KNN	83.3%	88.9%	85.7%

Our final results (see Table 4)

Table 5: Predictions of the two best models in Table 4

	Pic1	Pic7	Pic10	Pic20	Pic23	Pic25	Pic26
Forward Stage-wise	1	0	1	1	1	1	1
KNN	1	0	1	0	1	0	1

Our conclusion on the 7 disputed paintings Our final prediction on the 7 disputed paintings are based on a voting model, which is a combined model of the two most successful model in Table 4 and the style feature model in section 1.4.2. The rule is that only if the two model both give positive prediction then the test image will be predicted as genuine painting. Otherwise, we will refer to the predictions of style feature model in Table 3.

Based on this, we conclude that picture 1/10/23/25/26 are probably genuine and picture 7 is probably a forgery.

1.6 Remaining problems and Future work

Remaining problems

- The reason why the features extracted by the tight frame and those three statistics work still remains unknown.
- Also, the predictions given by different models vary from each other, of which the predicting criterion varies from each other.

Future work

- Apart from the tight frame, other frames should be explored and tested to help find out the reason for the ability of authentication.

- The style features extracted in section 1.4.2 also do a good job. However, the number of total features is over 100,000. So an efficient feature selection algorithm should be designed to reduce those noise features.

2 Raphael Artistic Style Transfer

2.1 Introduction

Before the advent of Neural Network, to transfer a style is to establish a mathematical model to extract the style information and then apply this model to a content image. Despite that such work can do a style transfer task, it is restrained by the certain style and certain mathematical model. If the style image changes, people need to go over the whole work again.

This phenomenon changes greatly after the work of Gatys et al. [2016], which will be the main reference to our work in this paper.

Related work In recent years, Deep Convolutional Neural Networks (CNN) [Krizhevsky et al., 2012] has attracted many attention because of its deep features generated ability. Zeiler and Fergus [2013] has been shown in high-level image recognition tasks that such deep features are better representations for images. This inspired work on neutral style transfer [Gatys et al., 2016], which successfully applied CNN (pre-trained VGG-16 networks [Zeiler and Fergus, 2013] to the problem of style transfer, or texture transfer [Gatys et al., 2015].

Style transfer is used as a means to migrate an artistic style from an example image to a source image. The decomposition of content and style in artistic images is bound to the coupling between the source content and the example style. Using CNN architecture is able to produce more impressive stylization results than traditional texture transfer, since a CNN is effective in decomposing content and style from images. Selim et al. [2016] further extended this idea to portrait painting style transfer by adding face constraints. The most related work to ours is patch-based style transfer by combining a Markov Random Field (MRF) and a CNN [Li and Wand, 2016].

Generative adversarial networks (GAN) [Goodfellow et al., 2014] are a powerful class of generative models that cast generative modeling as a game between two networks: a generator network produces synthetic data given some noise source and a discriminator network discriminates between the generator?s output and true data. GANs can produce very visually appealing samples, and it has many applications: estimating a high-resolution (HR) image from its low-resolution (LR) counterpart which can be referred to as super-resolution (SR) [Ledig et al., 2016], scene understanding including scene object retrieval [Gulrajani et al., 2017] and image-to-image translation [Isola et al., 2017].

2.2 Basic Method

Our basic algorithm for style transfer comes from the main idea of Gatys et al. [2016], in which a pre-trained deep convolutional neural network (in our paper is VGG19) is introduced to extract the information of style from the style image as well as the content from content image.

- **content loss**
  Let X be our input data matrix, and then F_{XL} is denoted as the features at layer L. So input data X's content information at layer L is determined by F_{XL}. And if we have our target content image X and the image Y that we want to reconstruct, the content loss of Y at layer L w.r.t X is defined as follows.

$$\mathcal{L}^{L}_{content}(Y,X) := \left|\left|F_{YL} - F_{XL}\right|\right|^2_2 \tag{1}$$

- **style loss**
  Again, for image X, Y at layer L, we have the features F_{XL}, F_{YL} respectively. First, the style information of X at layer L is defined as follows (Y is similar).

$$\begin{aligned}\mathcal{I}^{L}_{style}(X) &:= \Big(G_{XL}(i,j)\Big)_{K_L \times K_L} \\ G_{XL}(i,j) &:= \quad < F^{i}_{XL}, F^{j}_{XL} > \end{aligned} \tag{2}$$

where F^i_{XL} is denoted as the vectorized i^{th} feature of the X-features at layer L, K_L is denoted the number of vectorized features at layer L, which means G_{XL} is a $K_L \times K_L$ matrix, and $<, >$ is denoted as the inner product of two vectors.

Then the style loss of image Y w.r.t image X at layer L is defined as follows.

$$\mathcal{L}^L_{style}(Y, X) := \left|\left|\mathcal{I}^L_{style}(Y) - \mathcal{I}^L_{style}(X)\right|\right|^2_2 = \left|\left|G_{YL} - G_{XL}\right|\right|^2_2 \tag{3}$$

The total loss is defined as the linear combination of the loss of content and style [Gatys et al., 2016].

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{content} + \beta\mathcal{L}_{style}$$

So during the back-propagation, the gradient of Y at layer L is the linear combination of the content and style.

$$\nabla_L(Y, X_C, X_S) = \alpha_L\nabla_L(Y, X_C) + \beta_L\nabla_L(Y, X_S) \tag{4}$$

$$\nabla_{total}(Y, X_C, X_S) = \sum_{L_C}\alpha_{L_C}\nabla_{L_C}(Y, X_C) + \sum_{L_S}\beta_{L_S}\nabla_{L_S}(Y, X_S) \tag{5}$$

where L_C, L_S are the layers that determines content and style, respectively, and $\alpha_{L_C}, \beta_{L_S}$ are corresponding weights.

Here, we simply take uniform weights.[Gatys et al., 2016] That is to say,

$$\forall L_C,\ \ \alpha_{L_C} = \alpha\ ;\ \ \forall L_S,\ \ \beta_{L_S} = \beta$$

More details about the layers that determines content and style will be covered in the following section, as well as the choice of weights, $\alpha\&\beta$.

Unlike the traditional training procedure of Neural Network, the back-propagation process here aims to *reconstruct* the output image other than the parameters within the network layers.

Note that those successful results in Gatys et al. [2016], Liao et al. [2017], Chen and Koltun [2017] and many others have been a testament to the capability of CNN network to extract the information of style and content from a certain image. Still, the reason of CNN's such ability remains unknown, for the information of style is something subjective. The ultimate tool to test whether an image's style is the same as the other one mostly depends on people's eyes and visual perception.

2.2.1 Basic version (Multi-color)

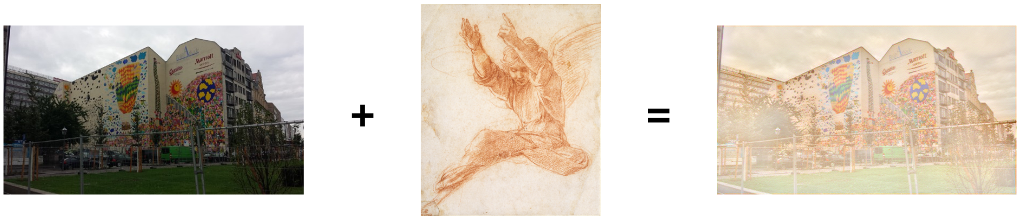


Figure 6: the basic version of style transfer

From Figure 6 we could see that the reconstructed image still looks like a photograph instead of a real art work of Raphael. To find the reason, we may take a closer look at the deep convolutional neural network.

So from Figure 7 we could see that the reconstructed content image remains most of information of the input content image, including the objects and the colors. Meanwhile, the reconstructed style image also resembles the input style image a lot, which can be another testament to the study of Gatys et al. [2016].

However, one problem is obvious: the color. Note that the artistic style of Raphael within our dataset is the same as the second picture in Figure 6. (Other styles of Raphael please refer to the section of further discussion) And so, the image that we synthesized should be single-color, like the style image, instead of the multi-color version.

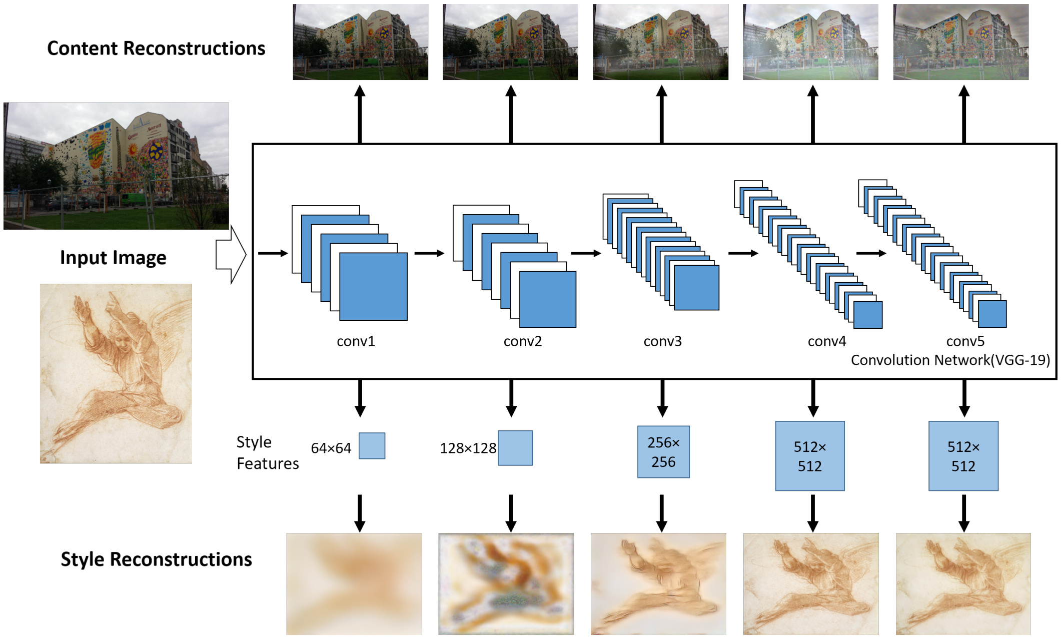


Figure 7: Detailed deep convolutional neural network

Remarks:

- Because the artistic style of Raphael within our dataset is not so strong as that of *Starry Night by Vincent van Gogh*, our weights are chosen to focus more on content: $\alpha = 4$ & $\beta = 1000$, compared to $\alpha = 1$ & $\beta = 1000$ in Gatys et al. [2016].
- Choice of different style image affects little on the synthesized image. It only changes the color as you can see in Figure 8. So we default the style image to be the same in our paper (and also we default the content image to be the same). If you want to see our results using other style images and other content images, please refer to section 2.5.2.

2.2.2 Advanced version (Single-color)

To deal with the problem of multi-color, we introduce some image preprocessing techniques. (see Figure 9)

From Figure 9 we can say that we've almost done the style transfer work. While if you are careful enough, you may find that there are still some little defects, such as the 'missing cloud' and the 'hollow tree'. (see Figure 10)

2.3 Another Potential Method: Discovery GAN

DiscoGAN [Kim et al., 2017] is a method based on GAN that learns to discover relations between different domains. Using the discovered relations, it is capable of successfully transferring style from one domain to another while preserving key attributes such as orientation. Hugely inspired by this work, we tried this architecture on our Raphael artistic style transfer task, wondering if we can get some unexpected achievements.

Results and Analysis As we crop images into small pieces, the smaller these pieces are, the big dataset we will get. Hence, details of the four dataset are: 5854 images and 227×227 for each image, 673 images and 600×600 for each image, 210 images and 1000×1000 for each image, 10 raw images and each image's size varies.

Results can be concluded in Figure 11 (Note that in order to present the results, we resize them to 600×600). We can clearly see that performances of DiscoGAN are not as good as the performances of Gatys et al. [2016] since all these pictures are blur and insufficient of details. However, among these

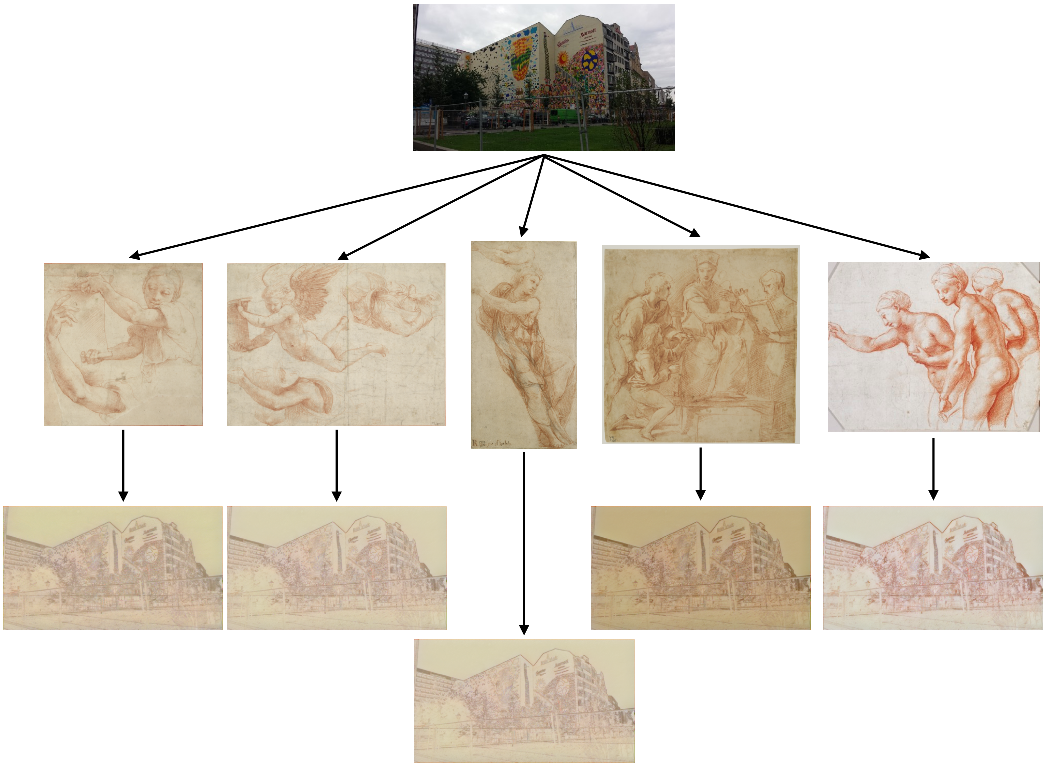


Figure 8: The effects of different input style image

four pictures, we are surprised to find out that both image size and dataset size play an important role in this task. For example, 600×600 dataset (with middle image size and middle dataset size) achieves the best performance while 227×227 dataset (with the smallest image size but the largest dataset) and raw dataset (with the largest image size but the smallest dataset) both have poor performances.

Reasons of such performances may be explained as follows. It's easy to collect two styles of pictures, while it's hard to capture two styles of pictures that represent exactly the same content. In our unpaired pictures, a large amount of data is needed to fully learn features of training data. Otherwise the model may not be good enough and may learn some noise data instead.

2.4 Our Results

A quick view of our final results can be seen in Figure 1. The full results images can be downloaded from the URL[1]. (Note that there are more than 1000 images in all.)

2.5 Further Discussion

2.6 Average Painter's style

In the experiment above we always pick one content picture and one style picture and make a one-one combination. However, in this case, we always transfer the content picture to a certain painting's style but not the painter, Raphael's style. From this point of view, we want to find a way to accomplish the task of style transfer using all 12 genuine Raphael's paintings.

From the model above, we represent a picture's style by the Gram matrixes of the feature space in different convolutional layers. So we can take the average of style features of 12 genuine paitings, use it as the painter's style and take it as the target style features to compute the style loss (MSE). Since the target style is fixed in the training process, this approach is feasible.

[1] `https://pan.baidu.com/s/1i6KBpNf` and the password is *jkwg*

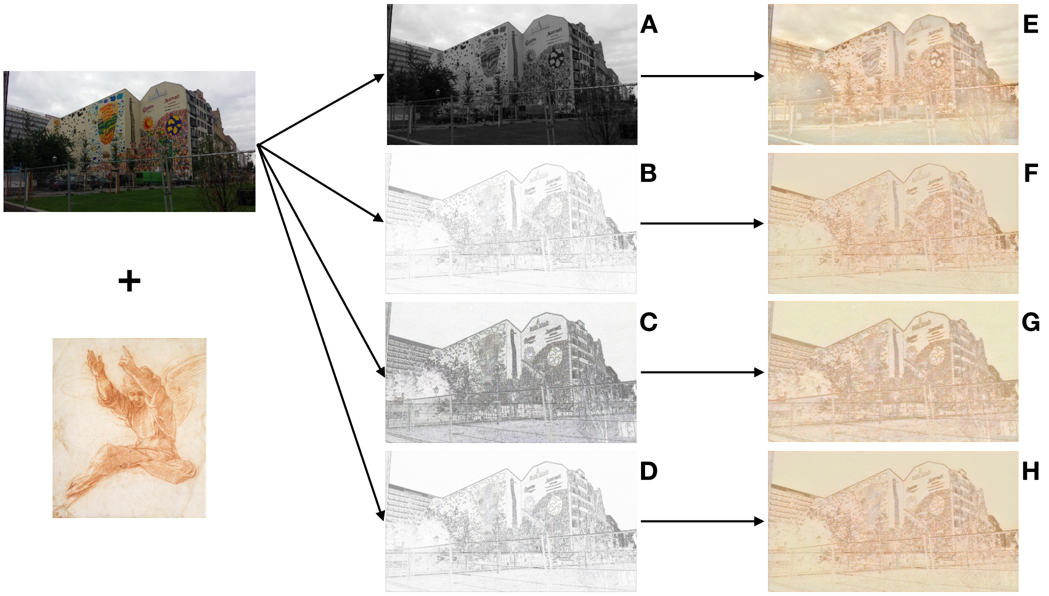


Figure 9: Here are four image preprocessing techniques: grey-scale, contour extraction, edge enhancement and sharpening. Image A, B, C, D are the content images after applying the four techniques, respectively, and image E, F, G, H are the corresponding synthesized images.

We implement this method on the test images and the result is shown in Figure 12. We compare the output trained from average features and one-picture features and find that average features are a more stable way to transfer to the painter's style and are available to get a picture more like a painting rather than still a photo.

2.6.1 Other artistic style transfer demo

We have applied our style transfer model using other famous artist's paintings. (see Figure 13) It turns out that our model performs well.

2.6.2 The problem of resolution

As mentioned in Gatys et al. [2016], a serious problem of style transfer is about image resolution. That is to say, the synthesized image is always small in size. (In our experiment, the size of synthesized image is 512×910)

Related work can be summarized as the algorithms, such as SCRNN[Hsu, 2009], ESPCN and DRCN[Chen and Koltun, 2017]. And there are some resolution work on photographs using GAN.[Ledig et al., 2016]

However, we find that the resolution algorithms mentioned in these articles are all about to deal with very small pictures and the size of output image are no greater than 512×512. So compared to our output image size 512×910, we think they won't be helpful to our results.

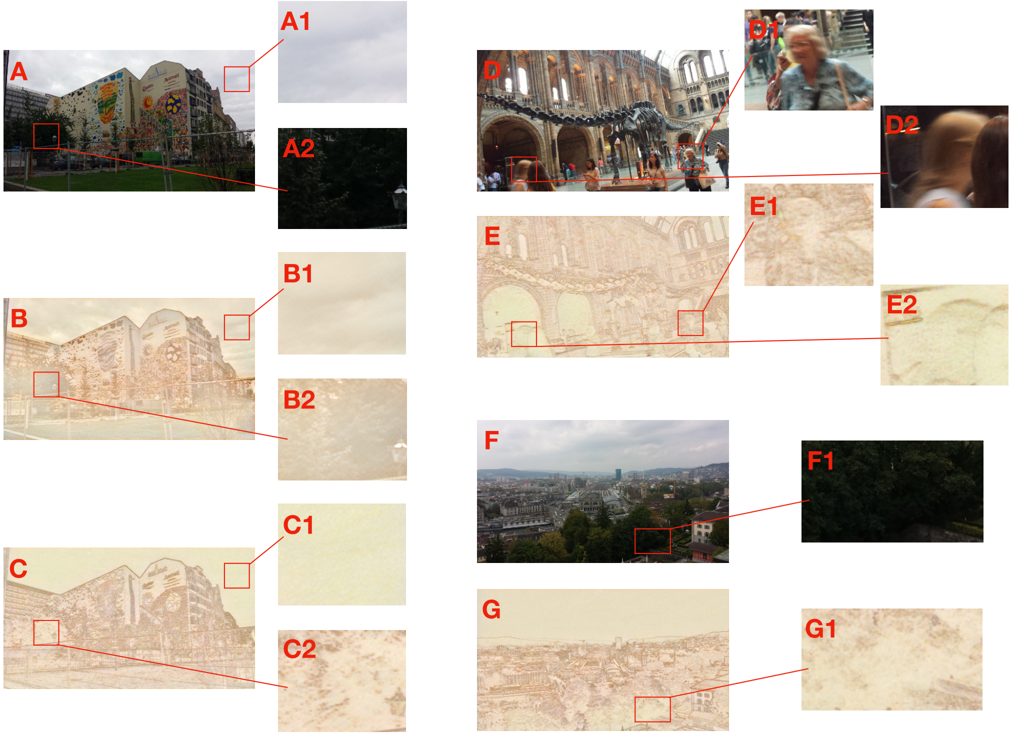


Figure 10: Some little defects. Here we mainly have two defects. One is shown as image C2, E1, E2 and G1, which can be concluded as detail loss. The other is presented in image C1, which can be concluded as object loss. The previous one can be attributed to the mistiness of the input image (see A2, D1, D2 and F1). While the latter one is a shortage of the image preprocessing techniques. We can see that the result with grey-scale input B performs better with both the clouds(B1) and the tree(B2). However, the problem is again the color. Till ddl, we are still not sure why a grey-scale input content can lead to a color like green(B2). Nevertheless, we still choose images like C to be our final results. (see Figure 1)

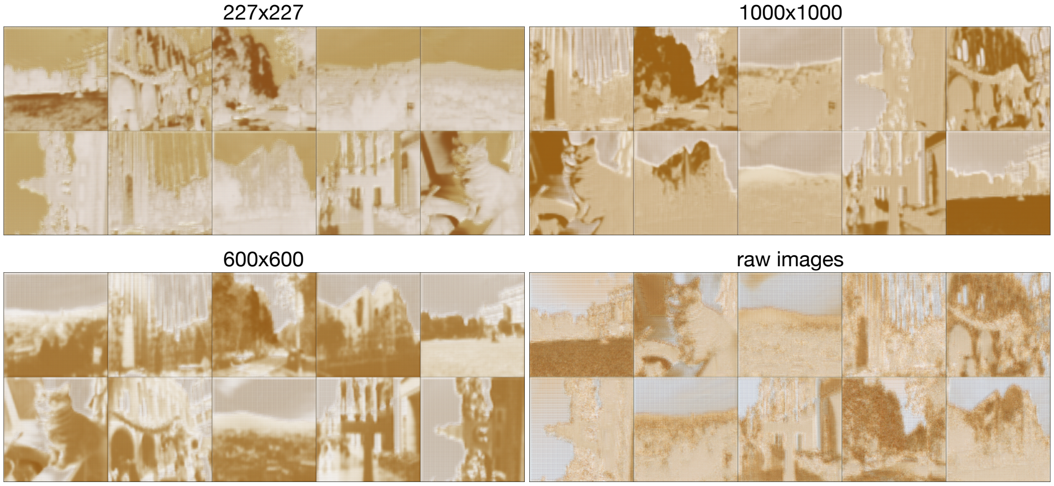


Figure 11: different performances under different sizes of training image (DiscoGAN)

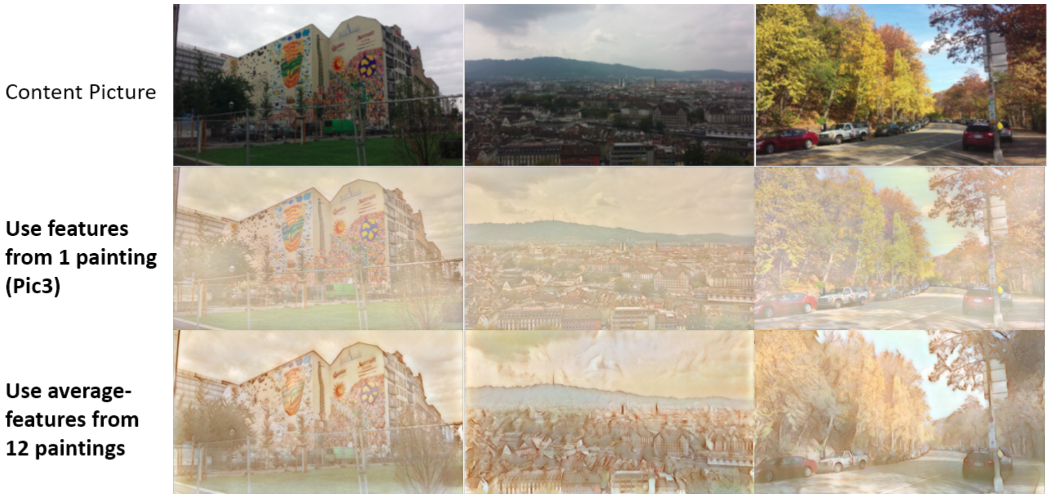

Content Picture
Use features from 1 painting (Pic3)
Use average-features from 12 paintings

Figure 12: Comparison of using one-picture features and average-features

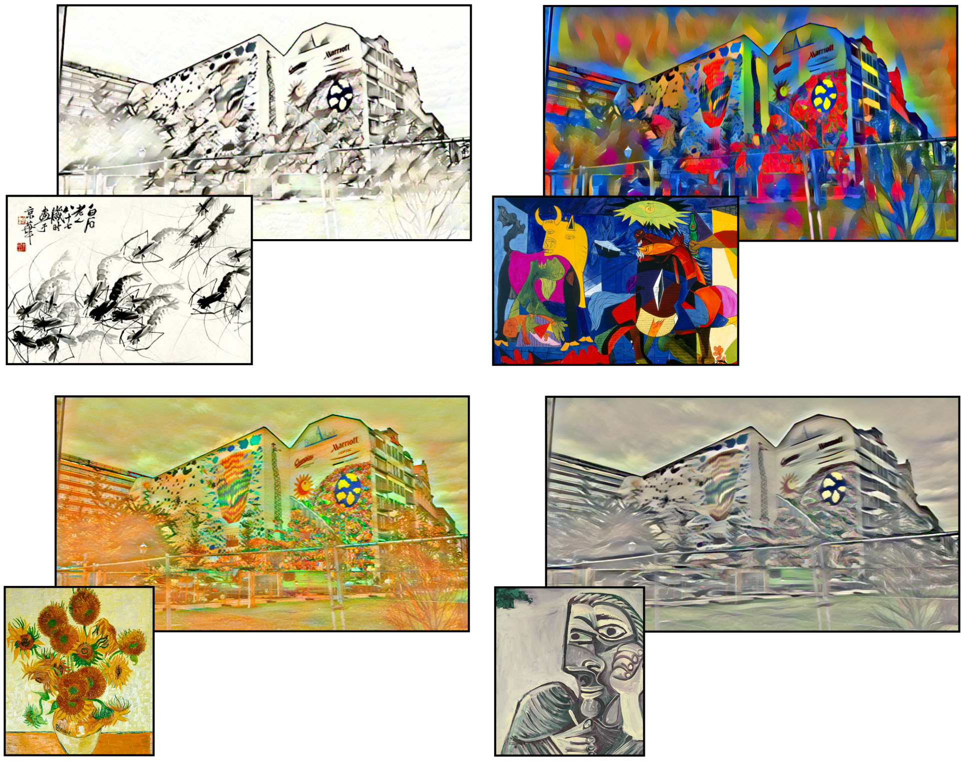


Figure 13: Other artistic style transfer demo. Top left is from Baishi Qi's *Shrimp*; top right is from Picasso's *Guernica*; down left is from Van Gogh's *Sunflowers*; down right is from Picasso's *Self portrait*.

References

Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 1, 2017.

Ahmed Elgammal, Yan Kang, and Milko Den Leeuw. Picasso, matisse, or a fake? automated analysis of drawings at the stroke level for attribution and authentication. *arXiv preprint arXiv:1711.03536*, 2017.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. *Febs Letters*, 70(1):51–55, 2015.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2414–2423. IEEE, 2016.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *International Conference on Neural Information Processing Systems*, pages 2672–2680, 2014.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5769–5779, 2017.

Chun Fei Hsu. Intelligent position tracking control for lcm drive using stable online self-constructing recurrent neural network controller with bound architecture. *Control Engineering Practice*, 17(6): 714–722, 2009.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jungkwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*, 2017.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *International Conference on Neural Information Processing Systems*, pages 1097–1105, 2012.

Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint*, 2016.

Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2479–2486, 2016.

Yan-Ran Li, Dao-Qing Dai, and Lixin Shen. Multiframe super-resolution reconstruction using sparse directional regularization. *IEEE Transactions on Circuits and Systems for Video Technology*, 20 (7):945–956, 2010.

Yan-Ran Li, Lixin Shen, Dao-Qing Dai, and Bruce W Suter. Framelet algorithms for de-blurring images corrupted by impulse plus gaussian noise. *IEEE Transactions on Image Processing*, 20(7): 1822–1837, 2011.

Yue Li, Jinglin Chen, Shijie Cui, and Yuxuan Zhou. Artistic authentication of raphael paintings. 2017.

Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. Visual attribute transfer through deep image analogy. *arXiv preprint arXiv:1705.01088*, 2017.

Haixia Liu, Raymond H Chan, and Yuan Yao. Geometric tight frame based stylometry for art authentication of van gogh paintings. *Applied And Computational Harmonic Analysis*, 41(2): 590–602, 2016.

Ahmed Selim, Mohamed Elgharib, and Linda Doyle. Painting style transfer for head portraits using convolutional neural networks. *Acm Transactions on Graphics*, 35(4):1–18, 2016.

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. 8689: 818–833, 2013.