

---

# Identification and Synthesizing of Raphael’s Paintings from the Forgeries

---

Jialun Shen

School of Data Science  
Fudan University  
Shanghai, China  
16307110030@fudan.edu.cn

Ruifan Deng

School of Data Science  
Fudan University  
Shanghai, China  
18300180053@fudan.edu.cn

## Abstract

Artistic authentication and image style transfer for painting have important applications in art. These two tasks are challenging in computer vision due to the complexity of artworks. In this project, we focus on identification and synthesizing of Raphael’s paintings from the forgeries. For artistic authentication task, we test combinations of different feature extraction methods and classification algorithms, and predict labels for disputed paintings whose author may be Raphael or not. For image style transfer task, we utilize a classic model-optimisation-based neural style transfer method to transfer arbitrary pictures into ones with the style of Raphael paintings.

## 1 Introduction

*Artistic Authentication* is the identification of genuine paintings by famous artists and detection of forgery paintings by imitators. Traditional approaches usually rely on experts’ discerning eyes and experience. In the past decade, advances have been made to apply statistical and quantitative techniques for authorship and style evolution in literary arts, by using high-resolution digital images of artists’ collections. For art authentication, the key point is to find the appropriate features which give a good separation between the artist’s paintings and those by his imitators. Various feature extraction methods have been proposed for artistic authentication. Geometric tight frame (Liu et al., 2016) and Gabor wavelet transformation (Johnson et al., 2008) are two representative feature extraction methods, which will be compared through experiments in Section 4.2.

*Image Style Transfer* is the task of rendering an image with artistic features guided by a style reference while maintaining the original content. In the past, re-drawing an image in a particular style requires a well-trained artist and lots of time. Since the mid-1990s, the art theories behind the appealing artworks have been attracting the attention of not only the artists but many computer science researchers. There are plenty of studies and techniques exploring how to automatically turn images into synthetic artworks. Inspired by the power of *Convolutional Neural Networks* (CNNs), Gatys et al. (2015) first study how to use a CNN to reproduce famous painting styles on natural images. This process of using CNNs to render a content image in different styles is referred to as *Neural Style Transfer* (NST). Since then, NST has become a trending topic both in academic literature and industrial applications. Common uses for NST are the creation of artificial artwork from photographs, for example, by transferring the appearance of famous paintings to user-supplied photographs. NST algorithms enable artists and designers around the globe to develop new artwork based on existent style(s).

## 2 Related work

### 2.1 Artistic authentication

Johnson et al. (2008) report their analysis of van Gogh's brushstrokes using several types of wavelet transformation. Li et al. (2011) further make an effort to extract visually salient brushstrokes of van Gogh based on an integrative technique of both edge detection and clustering based segmentation. With the extracted brushstrokes, some definitions of brushstroke features for art authentication were given in distinguishing van Gogh paintings from forgeries.

Qi et al. (2013) use background selection and wavelet-HMT-based Fisher information distance for authorship and dating of impressionist and post-impressionist paintings. They find that background information is much more reliable than the details of an intricate object which cannot represent the artist's natural style because of multiple edits and corrections. They also find that an artist's style should be interpreted as a probability distribution over a set of possible textures, and not just simply from the textures themselves.

Liu et al. (2016) propose a geometric tight frame based visual stylometry method to discriminate paintings by van Gogh from those by imitators. The methodology consists of some simple statistics of the geometric tight frame coefficients, as well as a boosting procedure for feature selection.

### 2.2 Image style transfer

Artistic stylisation is a long-standing research topic. Due to its wide variety of applications, it has been an important research area for more than two decades. Before the appearance of NST, the related researches have expanded into an area called *non-photorealistic rendering* (NPR). Some *image-based artistic rendering* (IB-AR) methods without CNNs have been proposed, such as stroke-based rendering (Hertzmann, 1998), region-based rendering (Kolliopoulos, 2005; Gooch et al., 2002; Song et al., 2008), example-based rendering (Hertzmann et al., 2001; Zhao and Zhu, 2011), image processing and filtering (Winnemöller et al., 2006). These algorithms without CNNs are capable of faithfully depicting certain prescribed styles. However, they typically have the limitations in flexibility, style diversity, and effective image structure extractions. Therefore, there is a demand for novel algorithms to address these limitations, which gives birth to the field of NST.

NST derived from two well-studied tasks: *Visual Texture Modelling*, which models and extracts artistic style from an image, and *Image Reconstruction*, which reconstruct the whole input image from the extracted image representation. Following the taxonomy of NST techniques proposed by Jing et al. (2019), current NST methods fit into one of the two categories: *Image-Optimisation-Based Online Neural Methods* and *Model-Optimisation-Based Offline Neural Methods*.

Due to the locality and spatial invariance in CNNs, it is difficult to extract and maintain the global information of input images. Therefore, traditional NST methods are usually biased and content leak can be observed by running several times of the style transfer process with the same reference style image. Some novel methods that do not rely on CNN have been proposed to tackle this problem. Deng et al. (2021) propose the first transformer-based style transfer method, taking long-range dependencies of input images into account for unbiased style transfer.

In this project, we focus on traditional NST methods.

#### 2.2.1 Image-optimisation-based online neural methods

Image-optimisation-based NST transfers the style by iteratively optimising an image (Gatys et al., 2015; Li et al., 2017a; Risser et al., 2017; Li et al., 2017b; Li and Wand, 2016a). The common limitation of image-optimisation-based NST algorithms is that they are computationally expensive, due to the iterative image optimisation procedure.

Gatys et al. (2015, 2016) propose the first NST algorithm. They find that a deep convolutional neural network is capable of extracting image content from an arbitrary photograph and some appearance information from the well-known artwork. They build the content component of the newly stylised image by penalising the difference of high-level representations derived from content and stylised images, and further build the style component by matching Gram-based summary statistics of style and stylised images.

### 2.2.2 Model-optimisation-based offline neural methods

Model-optimisation-based NST optimises a generative model offline and produces the stylised image with a single forward pass. Depending on the number of artistic styles a single generative model can produce, model-optimisation-based NST algorithms are further divided into *Per-Style-Per-Model* (PSPM) methods Johnson et al. (2016); Ulyanov et al. (2016, 2017); Li and Wand (2016b), *Multiple-Style-Per-Model* (MSPM) methods (Dumoulin et al., 2016; Chen et al., 2017; Li et al., 2017c; Zhang and Dana, 2018), and *Arbitrary-Style-Per-Model* (ASPM) methods (Huang and Belongie, 2017; Chen and Schmidt, 2016; Gu et al., 2018; Ghiasi et al., 2017; Li et al., 2017d).

PSPM approaches can produce stylised images two orders of magnitude faster than previous image-optimisation-based NST methods, achieving real-time style transfer. Separate generative networks have to be trained for each particular style image.

Johnson et al. (2016) and Ulyanov et al. (2016) share a similar idea of pretraining a feed-forward style-specific network and produce a stylised result with a single forward pass at testing stage.

Ulyanov et al. (2017) further find that simply applying normalisation to every single image rather than a batch of images leads to a significant improvement in stylisation quality. This single image normalisation is called *instance normalization* (IN). The style transfer network with IN is shown to converge faster than BN and also achieves visually better results.

MSPM improves the flexibility of PSPM by further incorporating multiple styles into one single model. MSPM is achieved either by tying only a small number of parameters in a network to each style (Dumoulin et al., 2016; Chen et al., 2017) or still exploiting only a single network like PSPM but combining both style and content as inputs (Li et al., 2017c; Zhang and Dana, 2018).

Chen et al. (2017) use separate network components to learn the corresponding content and style information. Mid-level convolutional filters (called “StyleBank” layer) are used to individually learn different styles. The rest components in the network are used to learn content information, which is shared by different styles. Their algorithm also supports flexible incremental training, which is to fix the content components in the network and train a “StyleBank” layer for a new style.

Given  $N$  target styles, Li et al. (2017c) design a selection unit for style selection, which is a  $N$ -dimensional one-hot vector. Each bit in the selection unit represents a specific style in the set of target styles. For each bit in the selection unit, they first sample a corresponding noise map from a uniform distribution and then feed it into the style sub-network to obtain the corresponding style encoded features. By feeding the concatenation of the style encoded features and the content encoded features into the decoder part of the style transfer network, the desired stylised result can be produced.

ASPM aims at one-model-for-all, i.e., one single trainable model to transfer arbitrary artistic styles. There are two types of ASPM, one built upon Non-parametric Texture Modelling with MRFs (Chen et al., 2017; Gu et al., 2018) and the other one built upon Parametric Texture Modelling with Summary Statistics (Huang and Belongie, 2017; Ghiasi et al., 2017; Li et al., 2017d).

## 3 Approach

### 3.1 Artistic authentication

Artistic authentication requires delicate analysis of specific style and habits. Although perceptual meaningfulness is a subjective concept, it can be measured with a computer by using techniques derived from biological and psychological models of the human visual system. The key point is to find the appropriate features which give a good separation between the artists’ paintings and those by his imitators. We use geometric tight frame (Liu et al., 2016) and Gabor wavelet transformation (Johnson et al., 2008) for feature engineering.

**Geometric tight frame filters & moment statistics** The Geometric tight frame we use has 18 filters  $\tau_0, \tau_1, \dots, \tau_{17}$ . They can capture the first and second-order differences in the horizontal,

vertical and diagonal directions in every small neighborhood of the paintings. The first three filters are shown below. Please refer to Liu et al. (2016) to check all the filters.

$$\tau_0 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \tau_1 = \frac{1}{16} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \tau_2 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}. \quad (1)$$

Given the  $i$ -th color painting, with  $m_i$ -by- $n_i$  pixels, we represent its grey-scale intensity by an  $m_i$ -by- $n_i$  matrix  $P_i$ . Convolve  $P_i$  with each  $\tau_j$ ,  $0 < j \leq 17$ , to get the corresponding  $m_i$ -by- $n_i$  tight frame coefficient matrices  $A^{(i,j)}$ :

$$A^{(i,j)} = P_i * \tau_j = \begin{pmatrix} a_{1,1}^{(i,j)} & \dots & a_{1,n_i}^{(i,j)} \\ \vdots & & \vdots \\ a_{m_i,1}^{(i,j)} & \dots & a_{m_i,n_i}^{(i,j)} \end{pmatrix}, \quad 0 \leq j \leq 17. \quad (2)$$

After the decomposition, there are 18 corresponding matrices for each painting. To capture the characteristics of the decomposition, we use three statistics as proposed by Liu et al. (2016). They are, specifically,

(i) the mean of the entries in the coefficient matrix  $A^{(i,j)}$ :

$$\mu^{(i,j)} = \frac{1}{m_i n_i} \sum_{l=1}^{m_i} \sum_{k=1}^{n_i} a_{l,k}^{(i,j)}, \quad (3)$$

(ii) the standard deviation of the entries in  $A^{(i,j)}$ :

$$\sigma^{(i,j)} = \left( \frac{1}{m_i n_i - 1} \sum_{l=1}^{m_i} \sum_{k=1}^{n_i} (a_{l,k}^{(i,j)} - \mu^{(i,j)})^2 \right)^{\frac{1}{2}}, \quad (4)$$

(iii) the percentage of the “tail entries”, which are entries that lies outside one standard deviation from the mean:

$$p^{(i,j)} = \#(\hat{A}^{(i,j)}) / (m_i n_i), \quad (5)$$

where  $\#(\hat{A}^{(i,j)})$  is the number of nonzero entries in the tail matrix  $\hat{A}^{(i,j)}$  which is defined by:

$$\hat{a}_{l,k}^{(i,j)} = \begin{cases} a_{l,k}^{(i,j)}, & \text{if } |a_{l,k}^{(i,j)} - \mu^{(i,j)}| \geq \sigma^{(i,j)} \\ 0, & \text{otherwise} \end{cases}. \quad (6)$$

Therefore, the feature vector of the  $i$ -th painting is represented by

$$(\mu^{(i,0)}, \dots, \mu^{(i,17)}, \sigma^{(i,0)}, \dots, \sigma^{(i,17)}, p^{(i,0)}, \dots, p^{(i,17)}). \quad (7)$$

**GaborWavelet filters & energy values** Next, we apply Gabor wavelet transformation (Johnson et al., 2008) in feature extraction. Gabor wavelet is similar to human visual system in terms of response mechanism, and can capture local features in multiple orientations and scales. Multiscale-oriented Gabor wavelet filters are sensitive to particular wave-like patterns. The Gabor wavelet filters come in pairs:  $G_{even}(x, y, \sigma, \alpha, \omega)$  and  $G_{odd}(x, y, \sigma, \alpha, \omega)$  are, respectively, the real and imaginary parts of the function:

$$e^{2\pi i \omega(x \sin \alpha + y \cos \alpha)} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (8)$$

where  $x$  and  $y$  are the spatial coordinates assuming the filters are centered at the origin,  $\alpha$  sets the spatial orientation and  $\omega$  the spatial frequency of the filter pair.

In theory, there are infinite number of Gabor filters. In an attempt to balance efficiency and convenience, we apply a set of filters that covers a range of orientations and scales: six orientations ( $\alpha = (k/6)\pi$ , for  $k \in \{0, 1, 2, 3, 4, 5\}$ ) and four scales (manually set  $\omega$  and  $\sigma$ ). Figure 1 visualizes 24 filters.

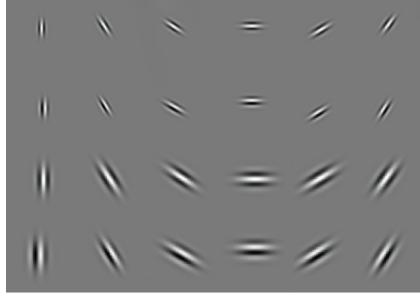


Figure 1: Visualization of 24 Gabor filters (Johnson et al., 2008)

Convolving these filter with an image yields a decomposition of the image structure into “energy values”, one energy value for each pixel, orientation, and scale. The Gabor wavelet energy value of an image patch is defined as the sum of the squared values obtained by convolving both components with an image patch. The whole process is shown in Figure 2.

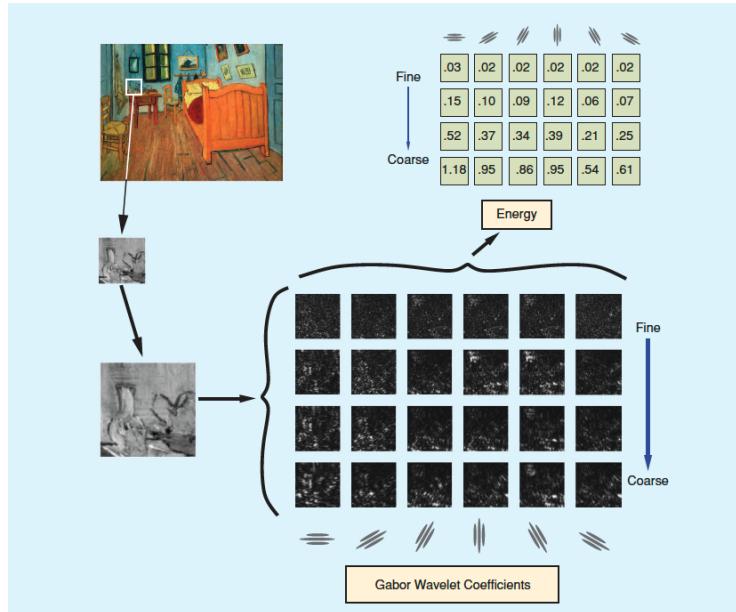


Figure 2: Example of an image patch (left) and the responses of the filters (right; dark to bright indicate low to high energy) (Johnson et al., 2008)

After feature extraction, we use several classification algorithms to perform the artistic authentication task. We use: (i) *Support Vector Machine* (SVM), without kernel and with penalty coefficient 1; (ii) *Decision Tree* (DT); (iii) *K-Nearest Neighbor* (KNN); (iv) *Neural Network* (NN), with one hidden layer containing 7 hidden units and a dropout layer with dropout probability 0.5. In addition, we apply forward stage-wise feature selection according to AUC.

### 3.2 Image style transfer

The NST algorithm we use is a PSPM method based on the work of Gatys et al. (2015); Johnson et al. (2016); Ulyanov et al. (2017). The main idea for style transfer task is to extract the low level features like the texture and color from the *style image*  $I_s$ , and apply it to some more semantic and high level features on a *content image*  $I_c$  to derive the *style-transferred image*  $I$ .

Roughly speaking, the output image  $I$  should be the one preserving the objects and concepts in the content image  $I_c$  whilst deviating little from the style image  $I_s$  in terms of the style. We can introduce a loss function to measure the difference between  $I$  and  $\{I_s, I_c\}$ , and what we need to do is

to produce the optimal  $I^*$  that minimizes the loss function. In this way, image style transfer problem can be reduced to an optimization problem:

$$I^* = \arg \min_I (\alpha \mathcal{L}_{content}(I_c, I) + \beta \mathcal{L}_{style}(I_s, I)), \quad (9)$$

where  $\alpha$  and  $\beta$  are two hyper-parameters that allow us to control how much emphasis we want to put on the content relative to the style,  $\mathcal{L}_{content}(I_c, I)$  and  $\mathcal{L}_{style}(I_s, I)$  are the content and style loss function, respectively.

The key point Gatys et al. (2015) proposed is that content and style losses are not based on pixel loss (the Euclidean distance between the output  $I$  and the target  $I_c$  or  $I_s$ ), but instead in terms of some more perceptual differences between the images. Pre-trained CNNs on a large image data set are able to encode perceptual and semantic information that we need to grasp the semantic differences. In our experiments (Section 4.3), VGG-19 network (Simonyan and Zisserman, 2014) trained on the ImageNet data set (Russakovsky et al., 2015) is used as the loss network, denoted as  $\phi$ .

Let  $\phi^j(I)$  be the activations of the  $j$ -th layer of the loss network  $\phi$ . Following the definition of Gatys et al. (2015); Johnson et al. (2016), we define the content and style loss functions, as well as a total variational regularization term.

**Content loss** The content loss is the squared Euclidean distance between feature representations of the content and combination images:

$$\mathcal{L}_{content}(I_c, I) = \|\phi^j(I_c) - \phi^j(I)\|_2^2. \quad (10)$$

**Style loss** Style loss is based on a *Gram matrix*  $G^l(I)$ , which is proportional to the covariances of corresponding sets of features in the  $l$ -th layer. The elements of  $G^l(I)$  is defined as:

$$G_{ij}^l(I) = \frac{1}{M^l} \sum_{k=1}^{M^l} \phi_{ik}^l(I) \phi_{jk}^l(I), \quad (11)$$

where  $\phi_{ik}^l$  denotes the  $k$ -th position of the  $i$ -th vectorized feature map in the  $l$ -th layer,  $M^l$  is the number of elements in each feature map. The loss function of style representation in the  $l$ -th layer is then defined as the squared Frobenius norm of the difference between the Gram matrices of the style and combination images:

$$\mathcal{L}_{style}^l(I_s, I) = \|G^l(I_s) - G^l(I)\|_F^2, \quad (12)$$

and the total style loss is a weighted sum of the contribution from each layer:

$$\mathcal{L}_{style}(I_s, I) = \sum_{l=1}^L w_l \mathcal{L}_{style}^l(I_s, I). \quad (13)$$

**Total variational regularization** A regularization term is added to the loss function, so as to encourage spatial smoothness. The total variation encourages images to consist piece-wise constant patches. For a continuous function  $f : \mathbb{R}^2 \supset \Omega \rightarrow \mathbb{R}$ , the total variation is given by:

$$\mathcal{R}(f, \delta) = \int_{\Omega} \left[ \left( \frac{\partial f}{\partial u}(u, v) \right)^2 + \left( \frac{\partial f}{\partial v}(u, v) \right)^2 \right]^{\delta/2} du dv. \quad (14)$$

For a discrete input image  $x \in \mathbb{R}^{H \times W}$ , the total variation is replaced by the finite discrete approximation:

$$\mathcal{R}(x, \delta) = \sum_{i,j} \left( (x_{i,j+1} - x_{i,j})^2 + (x_{i+1,j} - x_{i,j})^2 \right)^{\delta/2}, \quad (15)$$

where  $\delta$  is modified to balance the sharpness of the image against the removal of the artifacts called “spikes”. According to Krishnan and Fergus (2009), regularizer with  $\delta < 1$  is often used as a better match of the gradient statistics of natural images, while by choosing  $\delta > 1$  changes in images are distributed across regions rather than concentrated at a point or curve. When the target of the reconstruction is a colour image, the regulariser is summed for each colour channel.

**Total loss function** Based on the loss functions and regularization term defined above, the total loss function becomes:

$$\mathcal{L}(I_c, I_s, I) = \alpha \mathcal{L}_{content}(I_c, I) + \beta \mathcal{L}_{style}(I_s, I) + \gamma \mathcal{R}(I, \delta), \quad (16)$$

where  $\alpha, \beta, \gamma, \delta$  are hyperparameters. The final style-transferred image  $I^*$  satisfies:

$$I^* = \arg \min_I \mathcal{L}(I_c, I_s, I). \quad (17)$$

Besides, we also use the instance normalization technique proposed by Ulyanov et al. (2017).

## 4 Experiments

### 4.1 Datasets and settings

The data set provided by Prof. Yang Wang from HKUST<sup>1</sup> consists of high resolution scans of 28 paintings, scaled to a uniform density of 200 dots per painted inch. The picture size vary from 1192\*748 to 6326\*4457 pixels. Of the 28 paintings, 12 have been attributed to Raphael, 9 have been known to be non-Raphael, and others are currently questioned by experts.

For artistic authentication task, we change them into grey-scale scans for easier studying and feature capturing. Usually, turning a colored picture into grey-scale might lose information. In this specific problem, however, the pictures given are all single-colored sketches (like the two examples shown in Figure 3). Thus, grey-scale does not make much difference.



Figure 3: Example pictures from the data set

There are numerous ways for grey-scale transformation. A simple and widely used formula is from BT.601, a standard issued in 1982:

$$\text{Grey} = \text{Red} * 0.299 + \text{Green} * 0.587 + \text{Blue} * 0.114. \quad (18)$$

Note that the data set is quite small, with only 21 labeled pictures, which is enough for synthesization but not for classification. Also, the size of each picture is quite large. In an attempt to enlarge this data set, we cut each labeled image into 4 parts. Thus we get 84 labeled pictures in total.

For image style transfer task, we choose two paintings attributed to Raphael (as shown in Figure 4) as the style images. We use the COCO2014 training images<sup>2</sup> data set as content images during training. The data set has 82,783 images.

---

<sup>1</sup><https://drive.google.com/folderview?id=0B-yDtwSjhaSCZ2FqN3AxQ3NJNTA&usp=sharing>

<sup>2</sup><http://images.cocodataset.org/zips/train2014.zip>



(a) No.21 Raphael



(b) No.22 Raphael

Figure 4: Chosen style images

#### 4.2 Artistic authentication

We randomly split 84 labeled pictures into training set and test set. We adopt leave-one-out cross-validation on training set to select features. Experiment results with *true positive rate* (TPR), *true negative rate* (TNR) and accuracy of all models are shown in Table 1.

Table 1: Experiment result for artistic authentication

| Feature extraction method | Model | TPR          | TNR          | Accuracy     |
|---------------------------|-------|--------------|--------------|--------------|
| Tight frame               | SVM   | 0.923        | 0.875        | 0.905        |
|                           | DT    | 0.923        | <b>1.000</b> | <b>0.952</b> |
|                           | KNN   | 0.615        | 0.625        | 0.619        |
|                           | NN    | 0.846        | 0.500        | 0.714        |
| Gabor wavelet             | SVM   | <b>1.000</b> | 0.000        | 0.619        |
|                           | DT    | 0.846        | 0.625        | 0.762        |
|                           | KNN   | <b>1.000</b> | 0.500        | 0.81         |
|                           | NN    | 0.846        | 0.375        | 0.667        |

The overall performance based on Tight Frame filters seems to be better than that of Gabor Wavelet. Models with Gabor wavelet have low TNRs, which means fake paintings cannot be identified correctly. It also seems that different combination of feature selection and classification methods vary a lot in performance. Decision tree with tight frame performs best among all the 8 models, with the highest accuracy of 0.952. Table 2 shows its prediction of disputed paintings, where  $p$  is the probability of being Raphael's work. Note that we also divide the test paintings into 4 parts, for each part predicted as being Raphael's work, we add its  $p$  with 0.25.

Table 2: Prediction for disputed paintings

| Painting No. | $p$  | Label              |
|--------------|------|--------------------|
| 1            | 0.5  | (remains disputed) |
| 7            | 1    | Raphael            |
| 10           | 1    | Raphael            |
| 20           | 0.25 | non-Raphael        |
| 23           | 0.75 | Raphael            |
| 25           | 0.25 | non-Raphael        |
| 26           | 0    | non-Raphael        |

#### 4.3 Image style transfer

We train our model on a Nvidia GTX 2080Ti GPU. As the method we use is model-based, it can perform fast (in less than a second on GPUs) style transfer after training. With low computational cost, it can also work on CPUs to perform offline image transfer. The default setting we use for Raphael style transfer is shown in Table 3. With the default settings, training time is about 1.5 hours.

Table 3: Default setting for Raphael style transfer

| Hyperparameter                     | Value  |
|------------------------------------|--------|
| style image                        | No. 22 |
| number of epochs                   | 2      |
| batch size                         | 4      |
| content weight $\alpha$            | 7.5    |
| style weight $\beta$               | 100    |
| total variation weight $\gamma$    | 200    |
| total variation parameter $\delta$ | 1      |
| learning rate                      | 0.001  |

Figure 6 shows how the model gradually learns to transfer image styles. Actually, there seem to be little difference among style-transferred images after the first epoch. We tried different style weights  $\alpha \in \{1, 10, 100\}$  to test the effect of the ratio  $\alpha/\beta$ . A comparision of style transfer effect after 2 epoches is shown in Figure 7. These pictures look very similar expect for some details of dark area. Furthermore, we also try different works of Raphael as the style image. Figure 5 shows the effect of different style images.

To see all the effects of synthesized testing images, please refer to Appendix A.

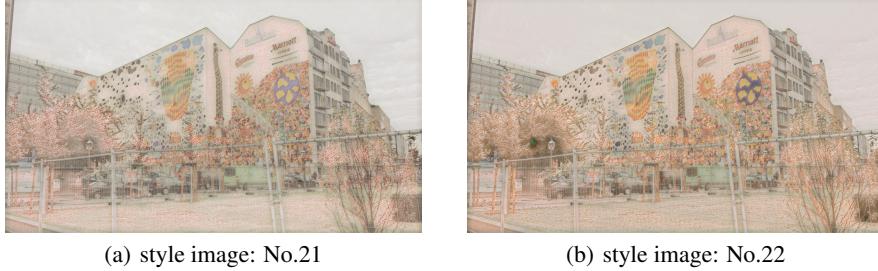


Figure 5: Synthesized images with different style images

## 5 Conclusion

In this project, we use machine learning and computer vision techniques to complete two challenging problems: artistic authentication and image style transfer. Specifically, we study methods for identification and synthesizing of Raphael's paintings from the forgeries. We test combinations of two feature extraction methods (geometric tight frame and Gabor wavelet transformation) and four classification algorithms (SVM, decision tree, KNN, neural network). Through experiments, we find that “decision tree + geometric tight frame” method has the best performance, and use it to predict labels for disputed Raphael paintings. For image style transfer task, we utilize a CNN-based, model-optimisation-based neural style transfer method, with well-designed content loss, style loss and total variational regularization, to transfer arbitrary pictures into ones with the style of Raphael paintings. For future work, we can try combing styles from multiple Raphael's works with similar original style, to acquire a more general style-transferer.

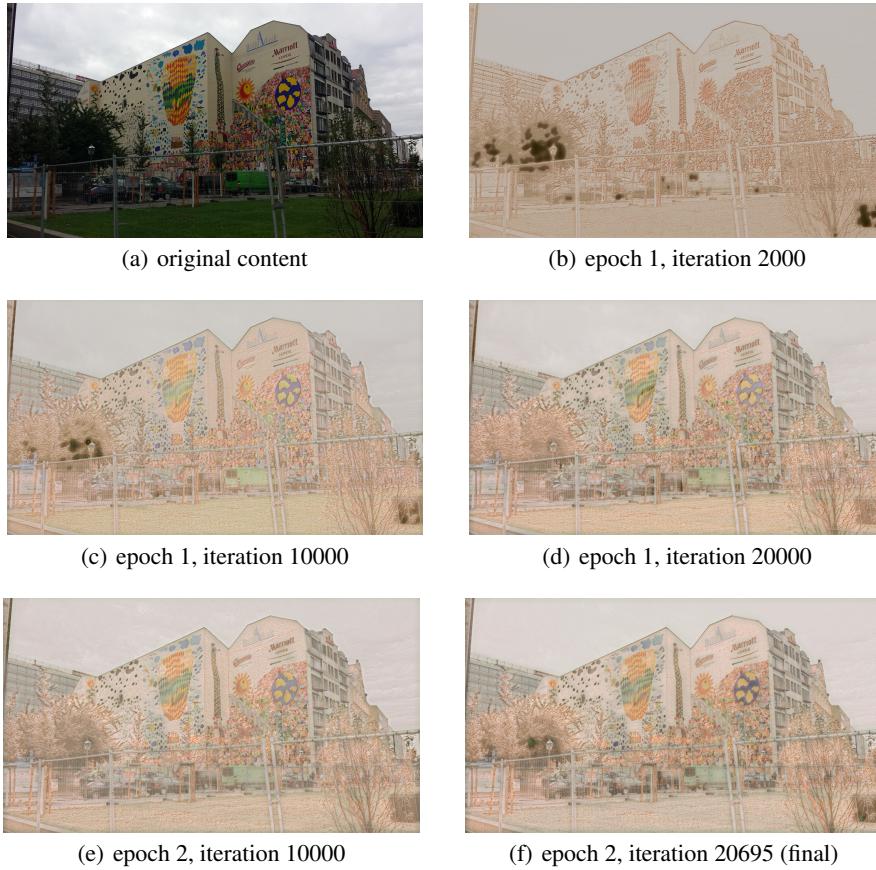


Figure 6: Synthesized images in different iterations during training

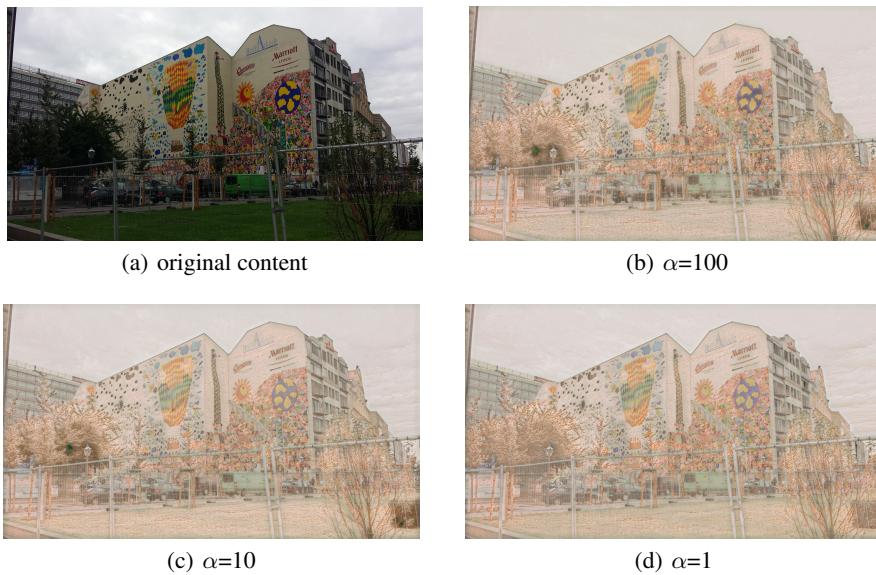


Figure 7: Synthesized images with different style weights

## References

- H. Liu, R. H. Chan, Y. Yao, Geometric tight frame based stylometry for art authentication of van Gogh paintings, *Applied And Computational Harmonic Analysis* 41 (2) (2016) 590–602.
- C. R. Johnson, E. Hendriks, I. J. Berezhnoy, E. Brevdo, S. M. Hughes, I. Daubechies, J. Li, E. Postma, J. Z. Wang, Image processing for artist identification, *IEEE Signal Processing Magazine* 25 (4) (2008) 37–48.
- L. A. Gatys, A. S. Ecker, M. Bethge, A neural algorithm of artistic style, arXiv preprint arXiv:1508.06576 .
- J. Li, L. Yao, E. Hendriks, J. Z. Wang, Rhythmic brushstrokes distinguish van Gogh from his contemporaries: findings via automated brushstroke extraction, *IEEE transactions on pattern analysis and machine intelligence* 34 (6) (2011) 1159–1176.
- H. Qi, A. Taeb, S. M. Hughes, Visual stylometry using background selection and wavelet-HMT-based Fisher information distances for attribution and dating of impressionist paintings, *Signal Processing* 93 (3) (2013) 541–553.
- A. Hertzmann, Painterly rendering with curved brush strokes of multiple sizes, in: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, 453–460, 1998.
- A. Koliopoulos, Image segmentation for stylized non-photorealistic rendering and animation, Citeseer, 2005.
- B. Gooch, G. Coombe, P. Shirley, Artistic vision: painterly rendering using computer vision techniques, in: Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering, 83–ff, 2002.
- Y.-Z. Song, P. L. Rosin, P. M. Hall, J. P. Collomosse, Arty Shapes., in: CAe, Citeseer, 65–72, 2008.
- A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, D. H. Salesin, Image analogies, in: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, 327–340, 2001.
- M. Zhao, S.-C. Zhu, Portrait painting using active templates, in: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering, 117–124, 2011.
- H. Winnemöller, S. C. Olsen, B. Gooch, Real-time video abstraction, *ACM Transactions On Graphics (TOG)* 25 (3) (2006) 1221–1226.
- Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, M. Song, Neural style transfer: A review, *IEEE transactions on visualization and computer graphics* 26 (11) (2019) 3365–3385.
- Y. Deng, F. Tang, X. Pan, W. Dong, C. Xu, et al., StyTr<sup>2</sup>: Unbiased Image Style Transfer with Transformers, arXiv preprint arXiv:2105.14576 .
- Y. Li, N. Wang, J. Liu, X. Hou, Demystifying neural style transfer, arXiv preprint arXiv:1701.01036 .
- E. Risser, P. Wilmot, C. Barnes, Stable and controllable neural texture synthesis and style transfer using histogram losses, arXiv preprint arXiv:1701.08893 .
- S. Li, X. Xu, L. Nie, T.-S. Chua, Laplacian-steered neural style transfer, in: Proceedings of the 25th ACM international conference on Multimedia, 1716–1724, 2017b.
- C. Li, M. Wand, Combining markov random fields and convolutional neural networks for image synthesis, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2479–2486, 2016a.
- L. A. Gatys, A. S. Ecker, M. Bethge, Image Style Transfer Using Convolutional Neural Networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
- J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in: European conference on computer vision, Springer, 694–711, 2016.

- D. Ulyanov, V. Lebedev, A. Vedaldi, V. S. Lempitsky, Texture networks: Feed-forward synthesis of textures and stylized images., in: ICML, vol. 1, 4, 2016.
- D. Ulyanov, A. Vedaldi, V. Lempitsky, Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 6924–6932, 2017.
- C. Li, M. Wand, Precomputed real-time texture synthesis with markovian generative adversarial networks, in: European conference on computer vision, Springer, 702–716, 2016b.
- V. Dumoulin, J. Shlens, M. Kudlur, A learned representation for artistic style, arXiv preprint arXiv:1610.07629 .
- D. Chen, L. Yuan, J. Liao, N. Yu, G. Hua, StyleBank: An Explicit Representation for Neural Image Style Transfer, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
- Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, M.-H. Yang, Diversified texture synthesis with feed-forward networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3920–3928, 2017c.
- H. Zhang, K. Dana, Multi-style Generative Network for Real-time Transfer, in: Proceedings of the European Conference on Computer Vision (ECCV) Workshops, 2018.
- X. Huang, S. Belongie, Arbitrary style transfer in real-time with adaptive instance normalization, in: Proceedings of the IEEE International Conference on Computer Vision, 1501–1510, 2017.
- T. Q. Chen, M. Schmidt, Fast patch-based style transfer of arbitrary style, arXiv preprint arXiv:1612.04337 .
- S. Gu, C. Chen, J. Liao, L. Yuan, Arbitrary style transfer with deep feature reshuffle, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 8222–8231, 2018.
- G. Ghiasi, H. Lee, M. Kudlur, V. Dumoulin, J. Shlens, Exploring the structure of a real-time, arbitrary neural artistic stylization network, arXiv preprint arXiv:1705.06830 .
- Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, M.-H. Yang, Universal style transfer via feature transforms, arXiv preprint arXiv:1705.08086 .
- K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 .
- O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, International journal of computer vision 115 (3) (2015) 211–252.
- D. Krishnan, R. Fergus, Fast image deconvolution using hyper-Laplacian priors, Advances in neural information processing systems 22 (2009) 1033–1041.

## A Style transfer results for all testing images

Figures below show the original testing images and style transferred ones.

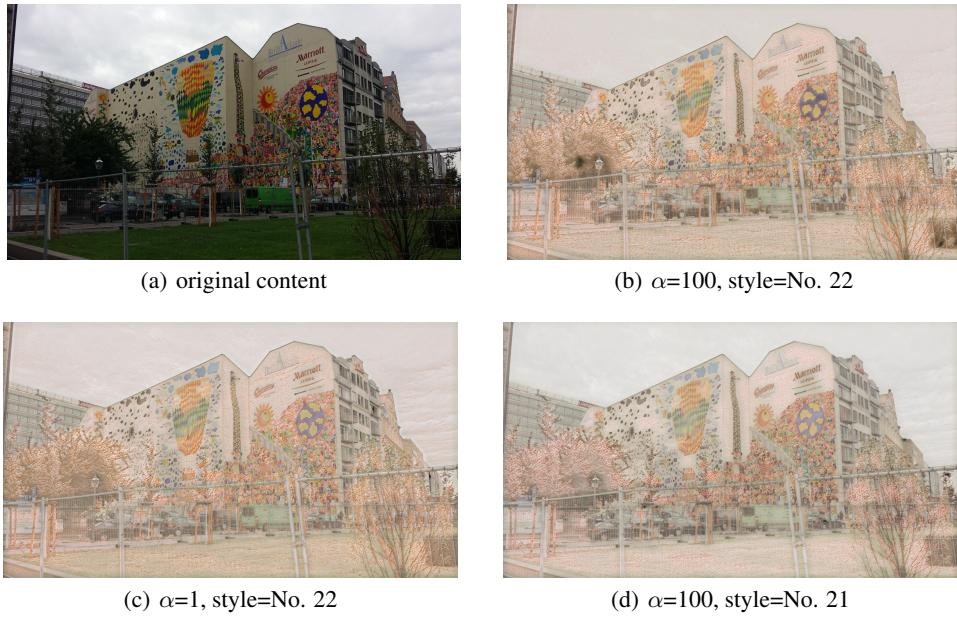


Figure 8: Synthesized images for test image No. 1

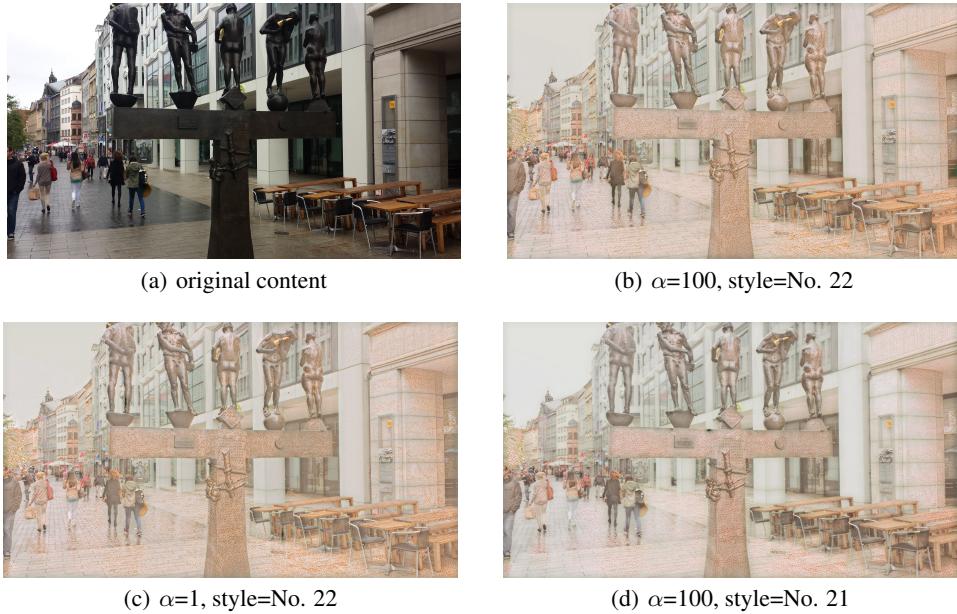


Figure 9: Synthesized images for test image No. 2

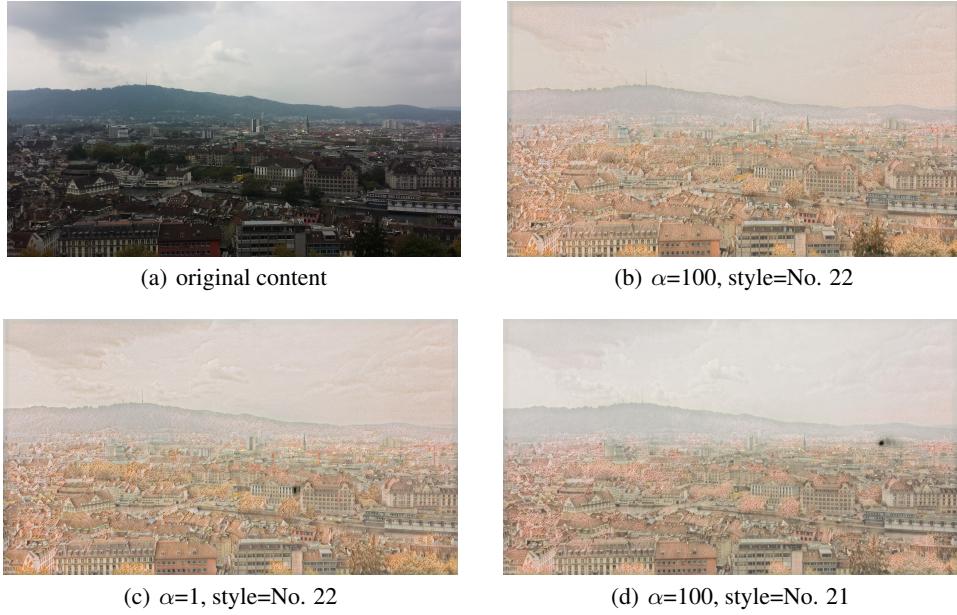


Figure 10: Synthesized images for test image No. 3

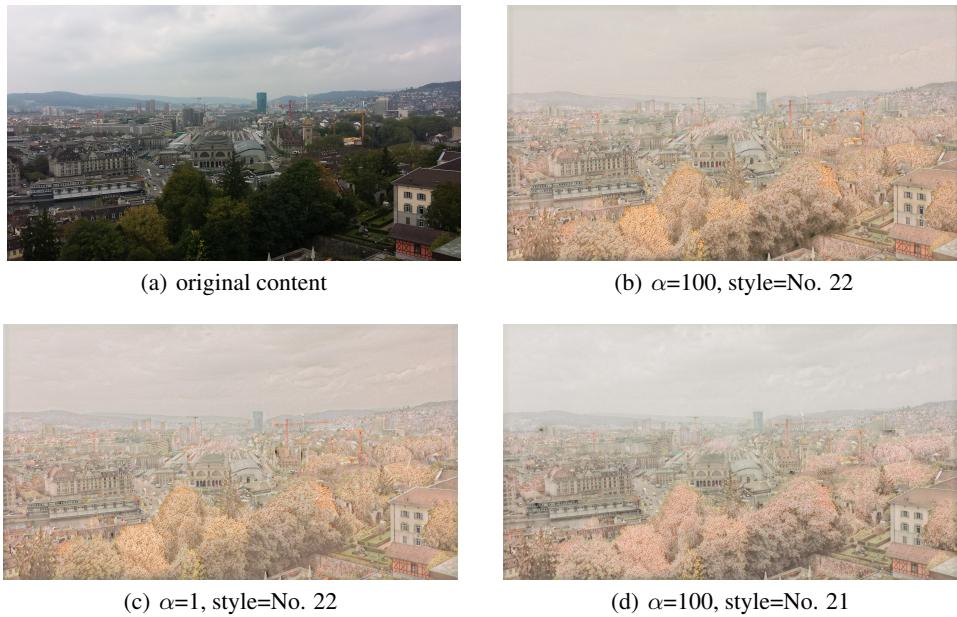


Figure 11: Synthesized images for test image No. 4

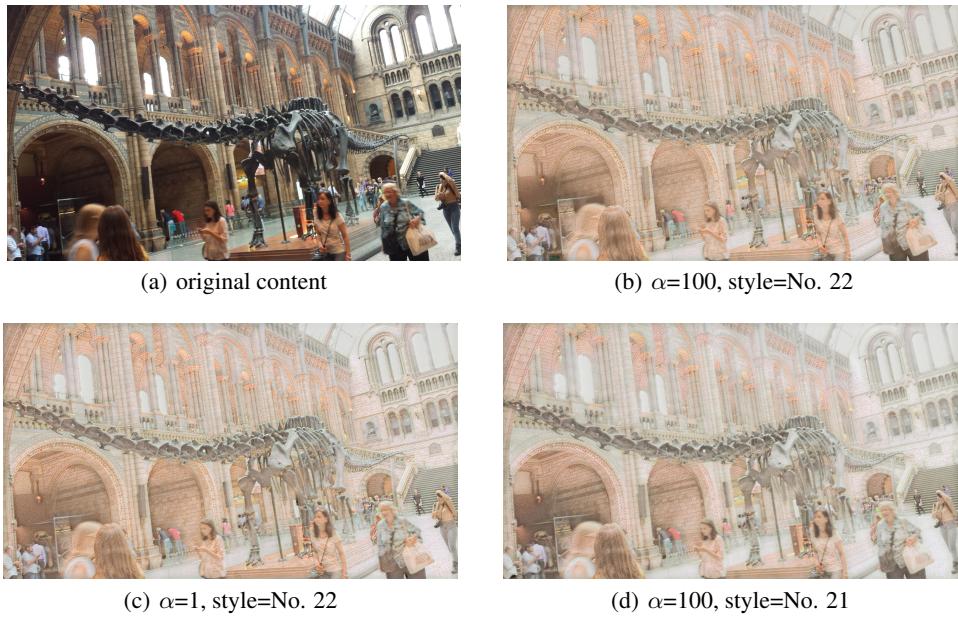


Figure 12: Synthesized images for test image No. 5

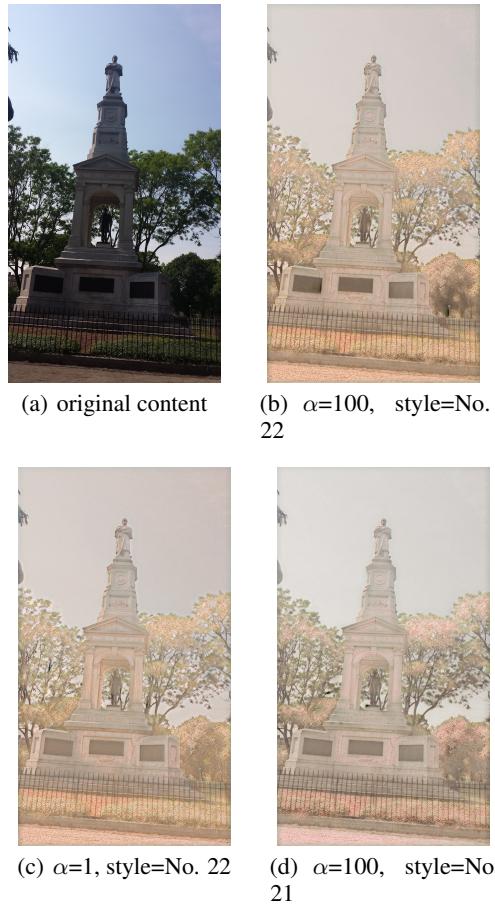


Figure 13: Synthesized images for test image No. 6

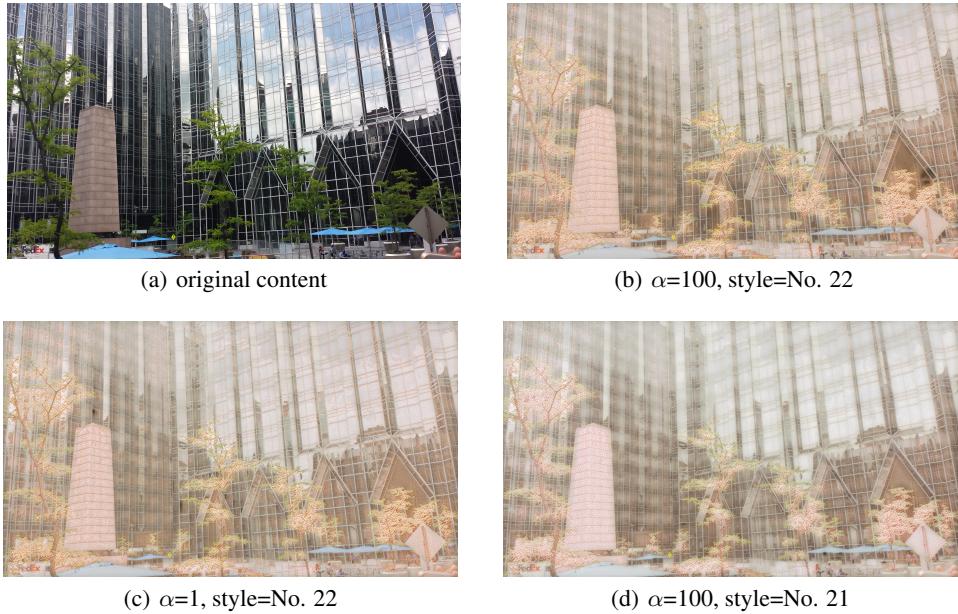


Figure 14: Synthesized images for test image No. 7



Figure 15: Synthesized images for test image No. 8