

Article

A Multi-Column Deep Framework for Recognizing Artistic Media

Heekyung Yang ¹ and Kyungha Min ^{2,*}

¹ Industry-Academy Collaboration Foundation, Sangmyung University, Seoul 03016, Korea; yanghk@smu.ac.kr

² Department of Computer Science, Sangmyung University, Seoul 0306, Korea

* Correspondence: minkh@smu.ac.kr; Tel.: +82-2-2287-5377

Received: 11 September 2019; Accepted: 30 October 2019; Published: 2 November 2019



Abstract: We present a multi-column structured framework for recognizing artistic media from artwork images. We design the column of our framework using a deep neural network. Our key idea is to recognize the distinctive stroke texture of an artistic medium, which plays a key role in distinguishing artistic media. Since stroke texture is in a local scale, the whole image is not proper for recognizing the texture. Therefore, we devise two ideas for our framework: Sampling patches from an input image and employing a Gram matrix to extract the texture. The patches sampled from an input artwork image are processed in the columns of our framework to make local decisions on the patch, and the local decisions from the patches are merged to make a final decision for the input artwork image. Furthermore, we employ a Gram matrix, which is known to effectively capture texture information, to improve the accuracy of recognition. Our framework is trained and tested using two real artwork image datasets: *WikiSet* of traditional artwork images and *YMSSet* of contemporary artwork images. Finally, we build *SynthSet*, which is a collection of synthesized artwork images from many computer graphics literature, and propose a guideline for evaluating the synthesized artwork images.

Keywords: media recognition; multi-column framework; CNN; deep learning

1. Introduction

Understanding artworks has been a long problem in computer vision, image processing and machine learning field. The recent progress on deep neural network structures presents many interesting schemes that recognize and classify artwork images and photographs according to their styles. We present a deep network-based approach for recognizing artistic media from artwork images and classifying the artwork images according to their creating artistic media. For the background of our approach, we survey *WikiArt*, the famous artwork image database, and list the most frequently used artistic media (see Figure 1). Among them, we select four most highly ranked media, including oil paint, watercolor, pencil and pastel. Since our approach relies on the sample data for training and test, we exclude artistic media whose artworks images are less than 1000. We omit tempera, which is rarely used nowadays. The four artistic media are very frequently used nowadays, and lots of artwork images created by the media are collected from various websites.

The unique stroke patterns of an artistic medium on an artwork image present a key to recognize which medium is employed to create the artwork image. For example, thin and parallel hatching stroke patterns are key to recognize that the artwork is drawn by a pencil and spread marks of colors are a key to a watercolor brush. In Figure 2, we illustrate the examples for pencil and watercolor artwork images. The red boxes on the input artwork images are the patches containing the stroke textures properly. Resizing the input image would harm the stroke textures so that the artistic media that created the

image might not be recognized (see Figure 2c). Therefore, we recognize that an artwork image is created using pencil, if we observe thin and parallel hatching stroke patterns on the image.

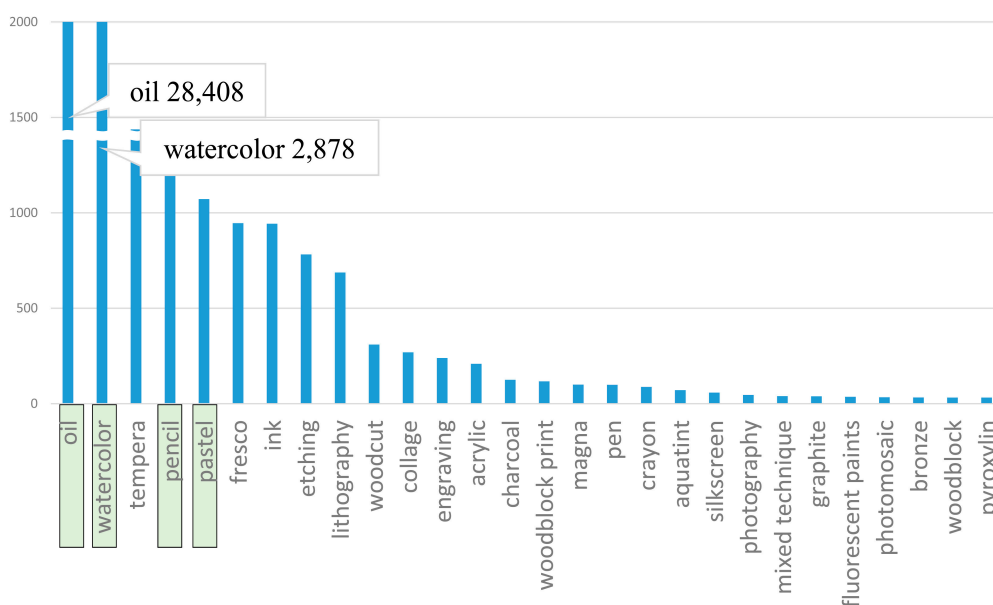


Figure 1. The frequency of artistic media from WikiArt.

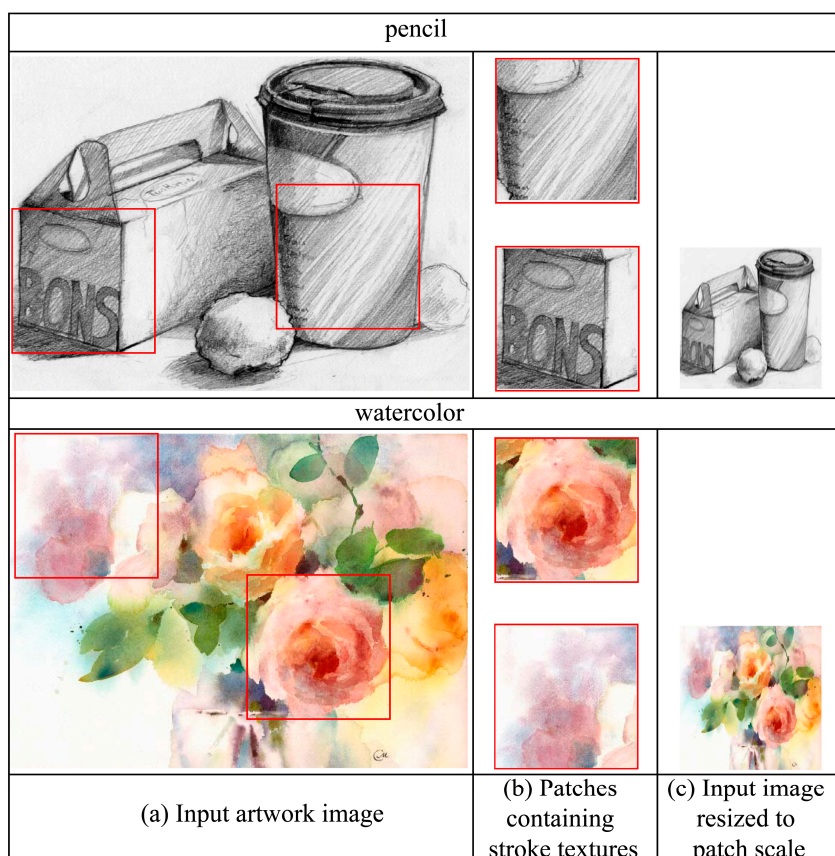


Figure 2. Patches containing stroke textures for pencil and watercolor artwork (a) Input artwork image; (b) patches containing stroke textures; (c) input image resized to patch scale.

In recent machine learning, many researchers have developed schemes for recognizing and classifying styles of photographs and artworks using deep convolutional neural network (CNN)

structures. These structures recognize styles by extracting the features, such as contents, texture, color, tone, composition of the object, and camera, etc. Since the features employed for style recognition are not explicitly defined and the relation between the features and the style is not clearly specified, an implicit approach, such as deep CNN structures is very appropriate for style recognition. Therefore, our approach for media recognition is devised based on a deep CNN structure. We devise the following strategies to enrich our framework that classifies and recognizes artistic media from artwork images.

Our first strategy is to devise a multi-column structured model, which is composed of several recognition modules. Artistic media are recognized through the stroke patterns that convey distinctive characteristics of the artistic media. Our survey on the artwork image database reveals that stroke patterns are on a local scale, not on a global scale. Stroke patterns are observed from the patches of an artwork image, not from the whole artwork image. Therefore, we devise an artistic media recognition strategy in two stages: The stroke patterns in the patches sampled from an artwork image are processed in independent recognition modules, and the decisions from the modules are integrated for a media recognition on an artwork image. In order to properly implement this strategy, we devise a multi-column structured model composed of independent recognition modules.

Our second strategy is to process texture information properly. Many CNN structures process both texture information and the content information simultaneously. Recently, Gatys et al. [1] presented a scheme that separates texture information from a source artwork image and applies it to another image to synthesize the artistic style of the source image to the target image. They proposed a Gram matrix, which is defined as a correlation of the feature vectors that express the texture embedded in the source image. Therefore, the Gram matrix estimated from an artwork image is expected to possess the texture information of the artistic style of the image. We estimate the Gram matrix from an artwork image and process them to recognize the artistic medium used to create the artwork image.

Our approach for recognizing artistic media from artwork images can be employed for evaluating synthesized artwork images. Many researchers in computer graphics have presented diverse techniques that synthesizes artwork images by simulating artistic media, such as pencil, oil paint, watercolor, pastel and ink, etc. Unfortunately, the evaluation for most of these techniques depends on either visual assessment or comparison to the results of previous works and the real artworks created by their aimed media. Our approach can present a quantitative and objective evaluation scheme for the synthesized artwork images, which has not been presented yet. A technique for simulating an artistic medium successfully mimics its target medium, if the stroke patterns of the target media are recognized from the synthesized artistic effects. Therefore, if our recognizer can recognize the target medium from a synthesized artwork image, then the technique that produces the image is evaluated to be successful in mimicking the medium.

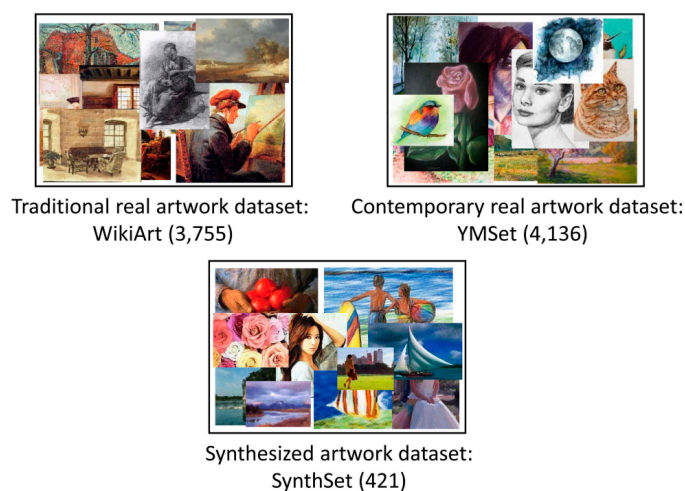


Figure 3. Three datasets for our approach.

We build three artwork image datasets: *Wikiset* of 4 K size, collected from the artwork images on *WikiArt*, the largest historical artwork image database and *YMset* of 4 K size, collected from various websites. Furthermore, we also build *SynthSet* of the synthesized artwork images collected from the literature on computer graphics society. The datasets are illustrated in Figure 3.

This paper is organized as follows. In Section 2, we briefly review our related works. In Section 3, we explain how our recognizer is organized. We suggest the implementation details and the training process of our recognizer in Section 4. Furthermore, we execute some experiments using our recognizer and dataset to show that our approach is valid in Section 5. Finally, we draw conclusions and suggest future work in Section 6. The process of approach is illustrated in Figure 4.

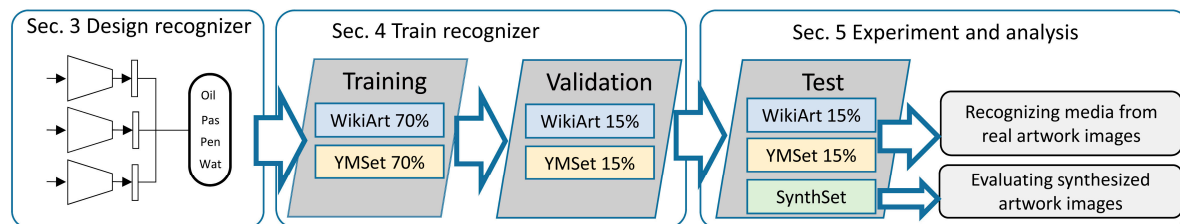


Figure 4. The process of our approach.

2. Related Work

Many machine learning researchers classify artworks and photographs according to their styles. Some of them try to classify artworks according to the material, such as paper, wood, silver, oil, ink and watercolor, etc. In the early days, handcrafted features were widely used. Recently, the convolutional neural network (CNN)-based approaches are spotlighted.

2.1. Schemes Using Handcrafted Features

Keren et al. [2] classified artworks according to their creators by the style of the artwork. Their scheme employed DCT coefficients as their classifying features. As a result, they classify five painters from 30 artworks. Li et al. [3] classified artists of Chinese ink paintings by analyzing wavelets from the brush strokes drawn on the artworks and Lyu et al. [4] applied Li et al.'s scheme to classify 13 drawings. Johnson et al. [5] presented a scheme that classifies van Gogh's paintings. However, these schemes show poor performance. They do not classify sufficient number of artworks and artists.

Shamir et al. [6] presented a style of an artwork as its creator, and the school of the creator. They defined 11 features using histogram analysis and edge statistics and classified three schools and nine creators from 60 artworks. Liu et al. [7] classified DART dataset that has 1.5 K paintings by employing handcrafted features, such as color, composition, line and their combinations. They compared the performances of several decision schemes, including SVM. They classify artworks into six schools of arts, including baroque, cubism, and impressionism. They also classify their creators.

The handcrafted feature-based schemes have difficulties in defining implicitly expressed styles with a set of explicit features. Most of them classify limited artwork classes and test their schemes on only easily distinguishable artworks.

Mensink et al. [8] presented an automatic classification technique on the photographs taken on the collections in Rijksmuseum in the Netherlands. They recognize a creator, a type, material and the creation year from the artworks. They recognize 20 material categories, such as paper, wood, silver, oil, ink and watercolor, etc. Their features are SIFT, which is effective in classifying different patterns. However, SIFT is not effective in classifying similar patterns. Therefore, their scheme shows poor performance for the stroke patterns produced by pencil, pastel and charcoal.

2.2. Schemes Using CNN-Based Features

The recent progress of deep convolutional neural networks motivates some researchers to apply this technique for style classification. These techniques can be classified into three categories: Feature fusion-schemes, CNN-based schemes and Gram matrix-based schemes.

Karayev et al. [9] classify 20 historical styles from 80 K artworks using the feature fusion approach. They employed AlexNet to extract features from artworks. They compared this feature set with other features, such as $L^*a^*b^*$ color histogram, GIST, graph-based visual saliency and meta-class binary features and proved that the features from AlexNet are very effective for artwork classification.

Recently, Tan et al. [10] visualize the feature response from the artwork classification. They extract features using AlexNet and visualize the reasons for prediction. They compare various options for their network: Whether pretrained for 1000 ImageNet dataset or not and decision options (SVM or softmax of neural network). As a result, a combination of pretrained and softmax decision method shows the best performance.

Strezoski et al. [11] introduced a classifying scheme for artistic media from OmniArt dataset, which has 432 K artwork images. They extracted features using recent deep CNNs, such as VGGNet, Inception-v3 and ResNet and compared their performances. They evaluated the performances of the models according to creator-accuracy, type-Mean average precision (MAP), material-MAP and period-mean. They show a limitation by omitting comparison with human baseline performance. Furthermore, they do not concentrate on stroke-based media, such as pencil, pastel, oil paint, and watercolor brush, which are the most widely used artistic media. They also do not discuss how to apply their schemes to classify synthesized artwork images, which are the results of non-photorealistic rendering (NPR) society.

More recently, several researchers employed a Gram matrix to classify styles. Matsuo and Yanai [12] employed VGGNet to extract Gram matrix of the feature maps. They compute PCA from the Gram matrix to classify artwork images from WikiArt that has more than 100 categories. Chu and Wu [13] extracted correlation features by producing the gram matrices from each layer of VGGNet to classify image styles. Sun et al. [14] designed a two-pass CNN structure that extracts object and texture features separately. The texture path employs the Gram matrix of the features to improve the classification performance. Their CNN structures are AlexNet and VGG-19. They do not require explicit definition of features, which is one of the major obstacles. The CNN-based features present better performance than human baseline, such as Mechanical Turk. These schemes also are proved to be effective in classifying more styles from larger dataset than the schemes using handcrafted features.

2.3. Patch-Based Schemes

Recently, patch-based works, which estimate styles from patches that are sampled from an image and collect the styles to classify the style of a target image, were presented. Lu et al. [15] employed AlexNet structure to classify the style of images from AVA dataset. They sample patches from an image randomly and accumulate the feature output in a stochastic way. Anwer et al. [16] introduced double-column CNN [17], which is designed based on VGG-16, to collect features of artworks from local scales, as well as global scales. The features from different scales are fed into different columns of CNN. The proposed technique show better accuracy than the normal networks fed by only global image. They employed AlexNet [15] and VGGNet-16 [16] for the columns of their structures. We have tested various recent CNN structures, including AlexNet, VGGNet, GoogLeNet, ResNet, and DenseNet, for our recognizer and decide DenseNet, since it shows higher performance than other CNN's.

Anwer et al. [16] sampled patches from an image according to the objects embedded in the image. Since the stroke textures of an artistic medium are frequently observed in the vacant area of an image, the object-based sampling would not be effective in our approach.

The existing patch-based schemes do not employ a Gram matrix for extracting features from the texture. By combining Gram matrix-based texture and patch sampling strategy, our scheme shows higher performance than the existing patch-based scheme.

3. Building Our Recognizer for Artistic Media

3.1. A Strategy for Our Recognizer

Our strategy in this study is to recognize artistic media through the stroke patterns observed on artwork images. Since the stroke patterns are in local scale, we sample several patches from an artwork image instead of resizing the image. Many existing studies resize input images into a fixed scale for the input of the deep neural network, but it may smear and distort the stroke patterns on the image. We design our recognizer using a multi-column structure, which is composed of individual recognizing modules that independently process the patches. Each module recognizes media from the stroke patterns located on the patches. A patch with very prominent stroke patterns gives a strong clue to an artistic medium than other patches with indistinct stroke patterns. Therefore, a decision from a module that recognizes the most prominent stroke patterns should dominate the decisions from other modules. We implement this idea in our multi-column structured recognizer.

Lu et al. [15] presented a multi-column structured model to estimate the aesthetic quality of an image. They process patches sampled from an input image in the modules of the multi-column structure and merge the results from the modules through sorting and statistic layers. Our model is different from Lu et al.'s model [15] in the following points. The first difference is the process of generating a final decision. Our model aims to detect stroke patterns that are used for creating artistic media. The most reliable result from a module dominates the results from other modules. Therefore, we choose a voting strategy to generate the final decision instead of sorting and statistics, which are the strategy in Lu et al.'s model. The second difference is the patch sampling strategy. Lu et al.'s model [15] does not consider the overlapping of the patches, since the aesthetic quality may become different with a slight translation of sampling position. However, our scheme avoids overlapping of the patches, since we aim to sample patches that cover the whole image. The third difference is the consideration of texture. Lu et al.'s model [15] does not consider the process of extracting features from texture embedded in the patches. We observe that the stroke patterns are expressed in texture information. Therefore, we employ a Gram matrix [1] that effectively extract features in texture information to extract texture information from stroke patterns in a patch and apply them for the final decision-making voting process.

Anwer et al. [16] presented a style classification scheme using a multi-column structured model whose input is multi-scaled patches. They vary the scales of the patches sampled from an input image and process them through a CNN model to extract features expressed in Fisher vectors. They apply a linear SVM model on the Fisher vectors to produce a final classification. Our model is different from Anwer et al.'s model [16] in the following points. The first difference is the rescaling of sampled patches. Since the stroke patterns observed in the patch is a key in our model, we do not rescale the patches. The second difference is the strategy for the final decision. Anwer et al.'s model [16] considers all the results from the modules for their final decision. Our scheme presents a voting strategy that considers the most reliable result from the modules. Since a decision from a patch with faint stroke patterns may distort the classification of a media, we ignore them through our voting strategy. The third difference is the consideration of texture. Anwer et al.'s model [16] does not consider features extracted texture, either. We employ a Gram matrix [1] to extract feature from textures and employ them in the final voting process.

Our recognizer for artistic media from artwork images is designed based on the existing CNN structure. Many existing style recognition frameworks employ various well-known CNN structures, such as AlexNet [9,10] and VGGNet [12–14]. Strezoski et al. employed several CNN structures, including VGGNet, GoogLeNet, and ResNet, and compared their performances [11]. We follow this strategy for our recognizer. Recently, we have tested five widely-used structures, including AlexNet, VGGNet, GoogLeNet, ResNet, and DenseNet, for recognizing artistic media from real artwork images and concluded that DenseNet, the latest CNN structure, shows best performance among them [18]. Therefore, we employ DenseNet-161 [19] for our recognizer.

Recognizing artistic media from artwork images depends on how the media stroke patterns on the images are properly processed. Therefore, texture-based features play a more important role in recognizing artistic media than the content-based features. To capture texture-based features properly, we employ a Gram matrix [1], which is known to show excellent performance in capturing and processing texture-based features. Sun et al. employed a Gram matrix for the features extracted by VGGNet for recognizing styles from images [14]. We apply Gram matrix for the features extracted by DenseNet-161 for our classifier.

3.2. Structure of Our Recognizer

The structure of our recognizer is illustrated in Figure 5. An artistic medium is predicted based on both Gram matrices extracted from the layers of DenseNet-161 and final result of the DenseNet-161. The input of our recognizer is a series of patches sampled from the input image. We make a different configuration for our classifier according to the stage: A training stage and a test stage. Note that a patch is processed in a single CNN structure for a single recognizing module (See Figure 5a). We train a single recognizing module and employ it for an element of the overall multi-column structure (See Figure 5b). For the test stage, we build k different modules to process the patches separately and combine the predictions for the final prediction of an artistic medium (See Figure 5b).

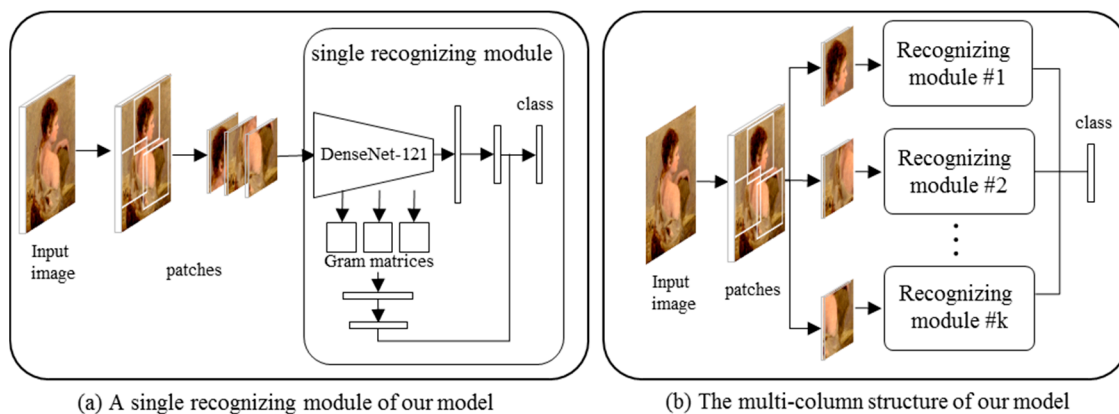


Figure 5. Our structure: (a) A single recognizing module and (b) an overall multi-column structure.

3.3. Estimating Gram Matrix for Our Recognizer

The stroke patterns produced by an artistic medium are observed from the stroke texture embedded in an artwork image. To build a set of feature vectors that properly extract the texture information, we employ the layered structure of a deep convolutional neural network (CNN), which processes the input image in a series of layers whose resolutions are abstracted as the layer goes deeper. The recent study on style transfer based on deep CNN structure [1] presents a Gram matrix that extracts feature maps that convey the texture information by the correlation of the feature vectors, which responds to a linear filter bank. Gram matrix of l -th layer, which is noted as $G_{i,j}^l$ is defined as:

$$G_{i,j}^l = \sum_k F_{i,k}^l F_{j,k}^l \tag{1}$$

where $F_{i,k}^l$ and $F_{j,k}^l$ is feature maps correspond to the i -th and j -th filter at position k , respectively [1]. Since Gram matrix is estimated at each layer of the CNN structure, we compute G^l 's for the layers and concatenate them as $G(x) = \{G^1, G^2, \dots, G^m\}$ for the x -th patch. DenseNet-161 structure has five layer blocks ($m = 5$), and we have five Gram matrices: G^1, \dots, G^5 . We illustrate the Gram matrices and their information in Figure 6. Note that we convert G^l 's into 1-dimensional vector before the concatenation, since G^l is a 2-dimensional vector of various sizes.

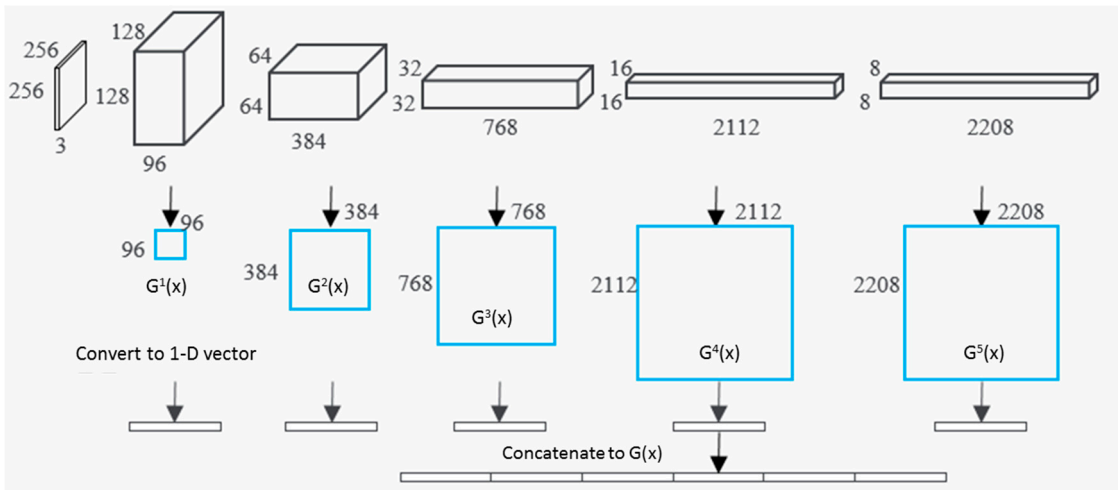


Figure 6. The Gram matrices extracted from DenseNet-161.

In order to effectively detect stroke patterns from the patches, we devise a novel idea for the concatenation. It is widely known that the feature maps extracted in the lower layer of a CNN structure correspond to primitive features and the feature maps extracted in the higher layer of a CNN structure correspond to abstract features. Since stroke patterns we aim to extract are more complex than the primitive features and more simple than the abstract features, we assign lower weights for the G^l 's of low and high l values, and higher weights for medium l values. Therefore, the weight, w_l , for G^l is determined according to Gaussian function:

$$w_l = \text{Gaussian}\left(2\left(\frac{l-1}{m-1}\right) - 1\right). \tag{2}$$

We concatenate them in the following formula:

$$G = w_1G^1 \circ w_2G^2 \circ \dots \circ w_mG^m, \tag{3}$$

where \circ is a concatenation operator. From $G(x)$, we compute $\bar{G}(x)$, which is a four-dimensional one hot vector through a fully connected layer. After averaging $\bar{G}(x)$ with $L(x)$, the final four-dimensional one hot vector from DenseNet-161, we conclude the final recognition for the x -th patch. This process is illustrated in Figure 7.

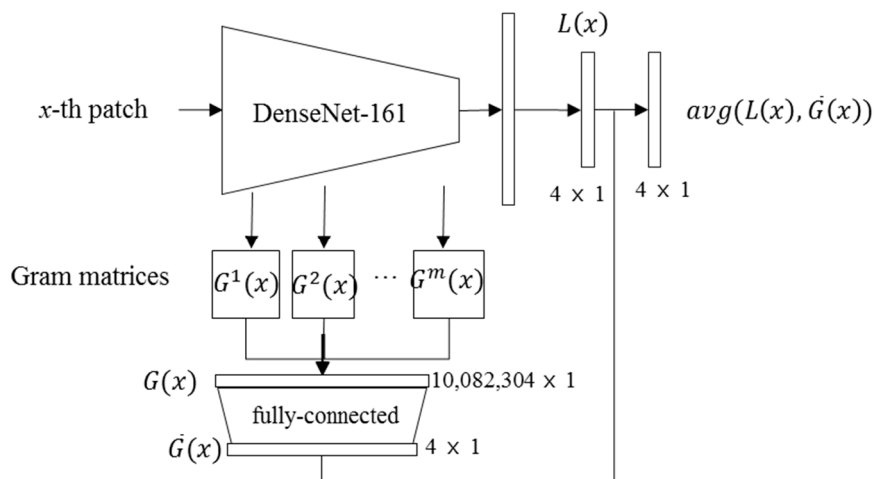


Figure 7. The process of decision making in a recognizing module.

4. Implementation and Result

4.1. Implementation

We have implemented our classifier in a personal computer with Intel®Xeon®CPU E5-2620 with 2.10 GHz, nVidia®Tesla®P40 GPU and 32GByte main memory. Our classifier is developed in Python with PyTorch library implemented on Linux®of Ubuntu version.

The loss function of our classifier, which is designed by employing the concept of cross entropy, is defined as follows:

$$Loss(y, \bar{y}) = - \sum_{d=1}^D \sum_{k=1}^K y \cdot \log \bar{y}, \tag{4}$$

where D is the batch size, K is the number of classes, y is the true label, and \bar{y} is the predicted label. We use D as 24 and K as 4. For the optimization, we employ the Adam strategy [20]. We set mini-batch size as 100, and learning rate as 0.0001. We trained our model for 100 epochs.

4.2. Data Collection

We collect real artwork image dataset in two ways: Historical artwork images and contemporary artwork images. *WikiSet* is a dataset of historical artwork images collected from *WikiArt*, the largest artwork image collection on the internet. *YMSet* is a dataset of the contemporary artwork images collected from various websites. Both datasets contain 4 K images, respectively.

We build *SynthSet* by collecting synthesized artwork images from the following literature: Oil paint brush [1,21–32], pastel [32–34], pencil [35–48] and watercolor brush [23,32,49–51]. We collect 421 images in total—178 oil paint images, 25 pastel images, 183 pencil images and 35 watercolor images. These images are presented in Figures A1–A3 in Appendix A.

4.3. Training and Results

According to the suggestion of an important machine learning textbook [52], we assign 5.2 K (70%) images for training, 0.6 K (15%) images for verification and 0.6 K (15%) for the test. We apply the synthesized artwork images only for test. We execute three tests: (i) *WikiSet* images, (ii) *YMSet* images and (iii) *SynthSet* images. The confusion matrices of these tests are illustrated in Figure 8 and suggest the performances in Table 1, respectively.

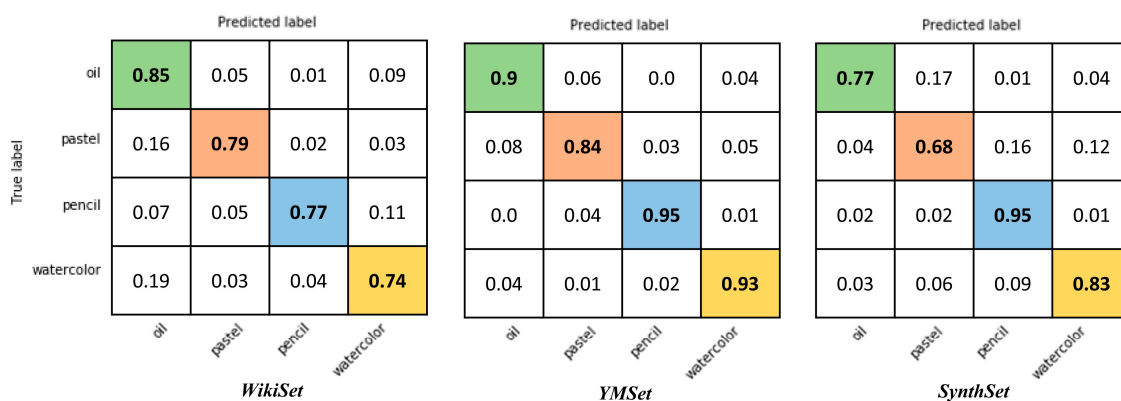


Figure 8. The confusion matrices from *WikiSet*, *YMSet* and *SynthSet*.

Table 1. The performance of our three tests: (i) Test with *WikiSet*, (ii) test with *YMSet*, and (iii) test with *SynthSet*.

Dataset	Media	No.	Acc.	Prec.	Rec.	F1
<i>WikiSet</i>	Oil	1000	0.85	0.68	0.85	0.75
	Pastel	1000	0.82	0.86	0.79	0.82
	Pencil	755	0.84	0.9	0.77	0.83
	Watercolor	1000	0.88	0.78	0.74	0.76
	Total	3755	0.85	0.80	0.79	0.79
<i>YMSet</i>	Oil	914	0.94	0.86	0.86	0.86
	Pastel	1014	0.9	0.83	0.84	0.84
	Pencil	1248	0.93	0.95	0.91	0.93
	Watercolor	960	0.96	0.9	0.94	0.92
	Total	4136	0.93	0.89	0.89	0.89
<i>SynthSet</i>	Oil	178	0.86	0.93	0.7	0.8
	Pastel	25	0.77	0.34	0.42	0.38
	Pencil	183	0.83	0.84	0.91	0.88
	Watercolor	35	0.92	0.51	0.77	0.61
	Total	421	0.85	0.66	0.70	0.67

5. Experiment and Analysis

5.1. Comparison

5.1.1. Comparison with the Existing Models

For comparison, we survey the related works and categorize the existing recognizer models in two aspects: The input and the features they employ. The input of the models is either the whole image or patches sampled from the input image. The feature on the models is from either the result of the last layer of the model or the Gram matrices computed from the overall layers. We select Huang et al.'s work [19] as the work that employs the output of the last layer for the whole image.

Lu et al.'s work [15] as the work that employs the output of the last layer for the sampled patches. Finally, we select Sun et al.'s work [14] as the work the employs a Gram matrix for the whole image. Our work employs a Gram matrix for the sampled patches, which has not been tried by the existing works. Note that Sun et al.'s work [14] and ours employ both the features from the last layer and the feature from the Gram matrix. We compared f1 score of the models for the *WikiSet* and *SynthSet* in Table 2. In comparison, our recognizer outperforms other models for every artistic media and dataset except only for the watercolor images from *SynthSet*.

Table 2. Comparison of four types of classifiers.

Dataset	Features		Input	
	Last Layer	Gram Matrix	Whole Image	Sampled Patches
Huang et al. 2017 [19]	O		O	
Lu et al. 2015 [15]	O			O
Sun et al. 2017 [14]	O	O	O	
Ours	O	O		O

5.1.2. Comparison with the Datasets

Table 1 shows the metrics, including accuracy, precision, recall, and F1, score for three datasets. We also illustrate F1 scores of the three datasets in Figure 9. In comparison, we observe that the F1 scores from oil paint and pencil for three models lie in the range of (0.75–0.86) and (0.83–0.93), respectively.

However, the F1 score of pastel and watercolor from *SynthSet* shows extraordinarily low values. We discuss the reason for this case in Section 5.2.

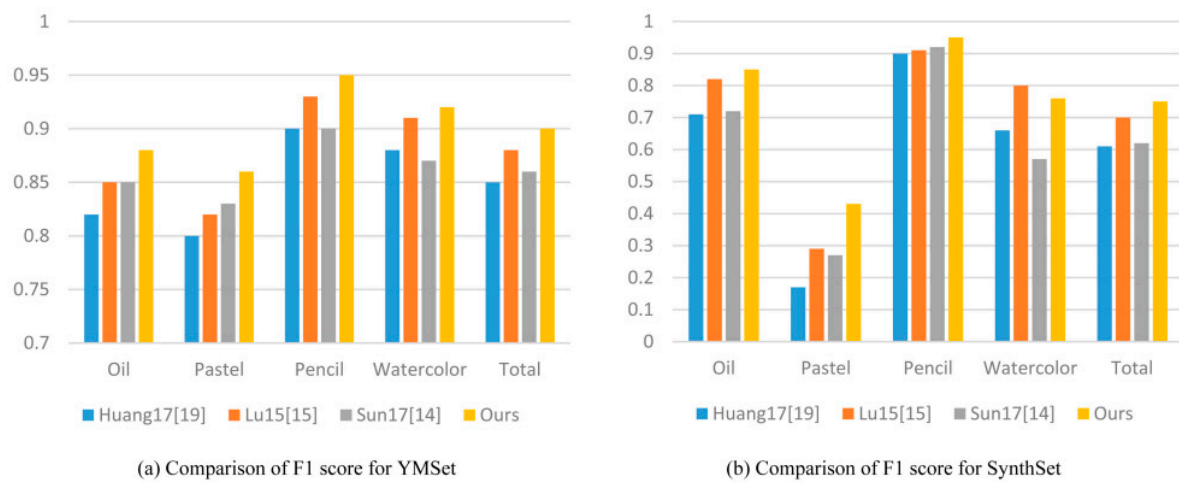


Figure 9. Comparison of F1 scores from the four recognizer models on *YMSets* (a) and *SynthSet* (b).

5.2. Analysis

5.2.1. Why *YMSets* Shows Best Performance?

According to our assumption that stroke texture plays an important role in recognizing the artistic media, the recognition on *YMSets*, the collection of contemporary artwork images, is expected to have higher performance than the recognition on *WikiSet*, the collection of traditional artwork images, since the stroke texture on the artwork images of *YMSets* is less damaged than the texture on the images of *WikiSet*. This assumption is proved to be valid by our experiment. According to Table 1 and Figure 10, *YMSets* shows greater performance than *WikiSet*.

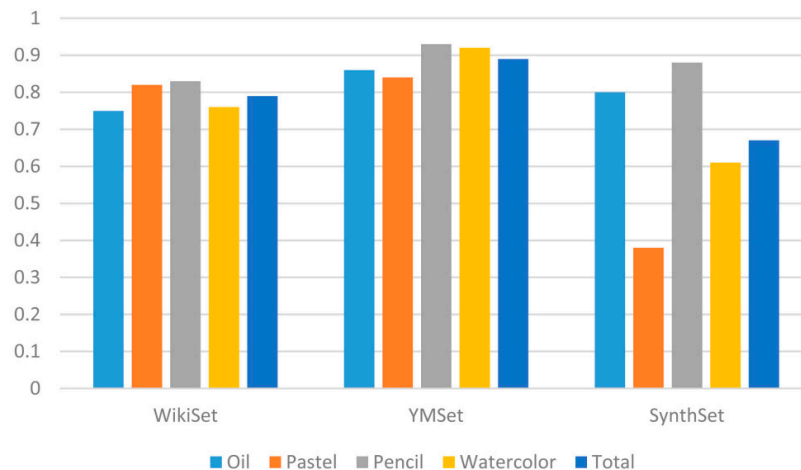


Figure 10. Comparison of F1 scores from three datasets on our recognizer.

5.2.2. The Similarity of the Recognition Pattern for *YMSets* and *SynthSet*

We compare three points for *YMSets* and *SynthSet* in the confusion matrix in Figure 8: (i) The order of recall values, (ii) the most confusing pair, (iii) the least confusing pair.

- i. The recall value of each medium is the diagonal entry of the confusion matrix. The decreasing orders of recall values for *YMSets* and *SynthSet* are illustrated in Figure 11, which notifies that both of the orders match.

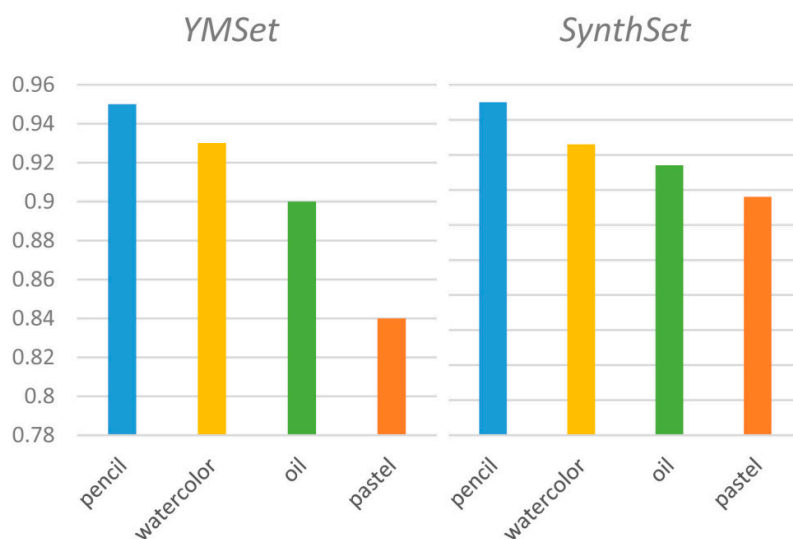


Figure 11. The decreasing order of recall values of the confusion matrices in Figure 8 for *YMSet*s and *SynthSet*.

ii. The most confusing pair of a medium is the medium whose entry is the largest entry except for the diagonal entry in the row of the confusion matrix. For example, the most confusion pair of oil in *YMSet* is pastel, since the entry for pastel, which is 0.06, is the largest in the row for oil except the diagonal entry. The comparison of the most confusing pairs for each medium is illustrated in the left column of Figure 12. In Figure 12, the most confusing pairs for oil and pencil coincides for *YMSet* and *SynthSet*. The reason why most confusing pairs for pastel and watercolor does not coincide is discussed in Section 5.3.

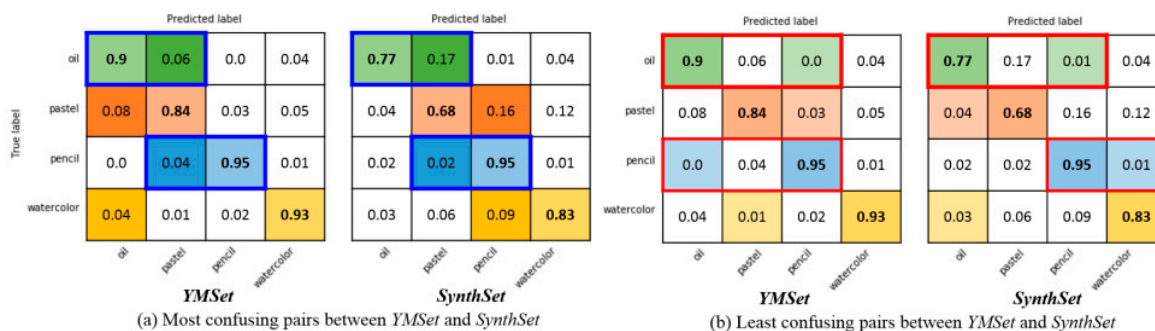


Figure 12. The most confusing pairs and least confusing pairs between *YMSet* (a) and *SynthSet* (b).

iii. The least confusing pair of a medium is the medium whose entry is the smallest entry except for the diagonal entry in the row of the confusion matrix. For example, the least confusion pair of oil in *YMSet* is pencil, since the entry for pencil, which is 0.0, is the smallest in the row for oil except for the diagonal entry. The comparison of the most confusing pairs for each medium is illustrated in the right column of Figure 12. In Figure 12, the least confusing pair for oil coincides, and the least confusion pair for pencil is less than 0.01 for *YMSet* and *SynthSet*. The reason why the least confusing pairs for pastel and watercolor does not coincide is discussed in Section 5.3.

5.2.3. The Evaluation Guideline for Synthesized Artwork Images

In Figure 10, we observe that our recognizer shows very competitive F1 scores for oil and pencil in *SynthSet*, which contains 178 and 183 sample images, respectively. We assume the low F1 scores for pastel and watercolor comes from the lack of sample images. Furthermore, in the Section 5.2.2, we shows that *YMSet*, the contemporary real artwork images, and *SynthSet*, the synthesized artwork

images, have very similar recognition patterns in the order of recall values, the most confusion pairs and the least confusing pairs. From these observations, we can argue that our recognizer can be an evaluation guideline for new techniques that aim to synthesize artwork images by mimicking artistic media. A technique that aims to mimic artistic media is regarded to achieve its purpose, if the resulting image is recognized as if it has been drawn by the aimed media. Until now, the recognition is only evaluated by human assessment. However, our recognizer that shows similar recognition patterns between real artwork images and synthesized artwork images can be an evaluation guideline for the synthesized artwork images. A process of the evaluation guideline is illustrated in Figure 13.

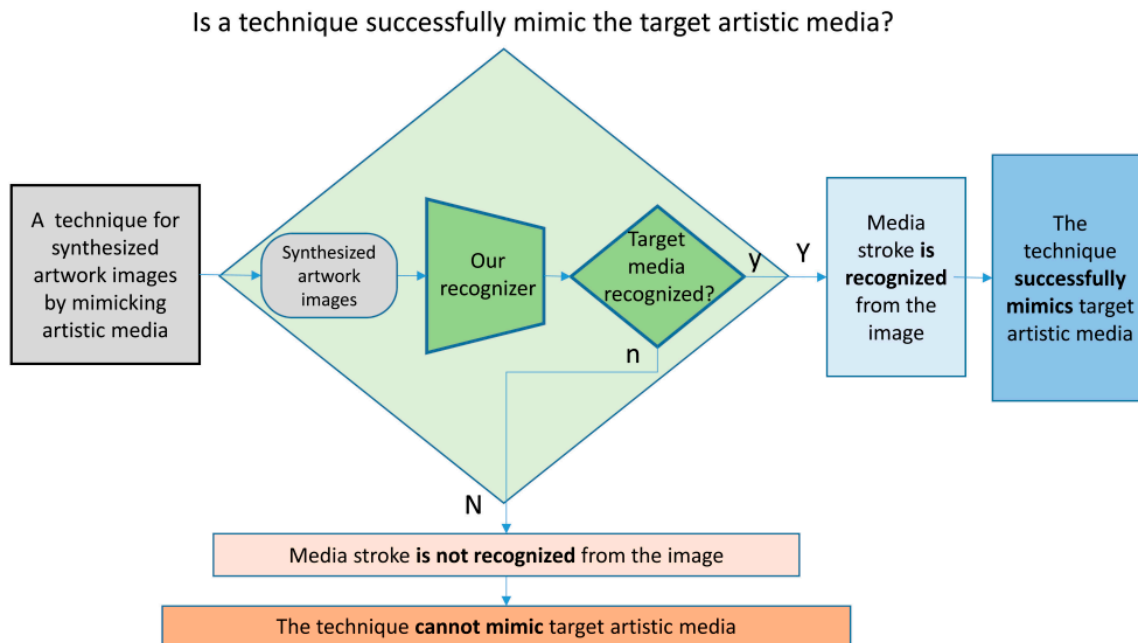


Figure 13. The process of evaluation guideline for synthesized artwork images. A technique that generates synthesized artwork images by mimicking artistic media is evaluated to be successful, if our recognizer can recognize the target media from their result images.

5.3. Limitation

After our experiment, we list the following limitations of our approach.

1. We concentrate on four media.

We have designed a recognizer for four media, which are very frequently used to create both real artwork and synthesized artwork. For traditional artwork, we can list more artistic media, such as tempera, fresco, ink, etching, lithography, etc. Even though they are not as popular nowadays, the study on recognizing traditional artworks may need to recognize artworks created by those media listed above.

2. Lack of samples for synthesized pastel and watercolor.

We have surveyed literature on mimicking artistic media in computer graphics society, and we cannot find a sufficient amount of literature for pastel and watercolor. In pastel, we find two works, one of which aims to mimic crayon, which is very similar to pastel [33]. Due to the lack of sample images for pastel and watercolor in *SynthSet*, the results on both media is less confident than those from oil paint and pencil.

3. Strategy for sampling patches.

We sample patches from an artwork image in random to capture stroke texture lying in an artwork image. Since most of the stroke textures lie in local scale, sampling several patches may not catch proper stroke texture. To avoid this problem, we need an intelligent strategy for sampling patches that would not miss stroke texture. Such a strategy may require low-level information from image processing techniques, such as gradient, curvature, and saliency.

6. Conclusions and Future Work

We have presented a multi-column structured framework for recognizing artistic media from artwork images. We sample several patches from an input artwork image and process them in each column of the framework to recognize stroke textures of an artistic medium that is used to create the artwork image. The local decisions on the patches are merged to make a final decision for the artwork image. We employ a Gram matrix, which is known to be capture texture information from an image very effectively. We trained and tested our framework using real artwork datasets and compared the performance with the existing CNN-based recognizers to show that our recognizer shows the best performance. Furthermore, we also build synthesized artwork images and test them using our recognizer. Our recognizer shows the possibility of presenting a guideline for evaluating synthesized artwork images.

We are going to improve the accuracy of our recognizer by sampling patches from an input artwork image more intelligently. We also extend our datasets to cover other media or artistic style, such as abstraction and pop art.

Author Contributions: Conceptualization, H.Y. and K.M.; methodology, H.Y. and K.M.; software, H.Y.; validation, H.Y. and K.M.; formal analysis, H.Y. and K.M.; investigation, H.Y. and K.M.; resources, H.Y.; data curation, K.M.; writing—original draft preparation, H.Y. and K.M.; writing—review and editing, K.M.; visualization, H.Y. and K.M.; supervision, K.M.; project administration, H.Y.; funding acquisition, K.M.

Funding: This paper is supported by National Research Foundation of Korea (NRF) through NRF-2018R1D1A1A02050292.

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

Appendix A

We present the images in *SynthSet* in Figures A1–A3.

Literature	Synthesized oilpaint artwork images
Litwinowicz 1997 [21]	
Hertzmann 1998 [22]	
Hays & Essa 2004 [23]	
Kagaya et al. 2011 [24]	
Zeng et al. 2009 [25]	
Lin et al. 2010 [26]	
Zhao & Zhu 2010 [27]	
Zhao & Zhu 2011 [28]	
O'Donovan & Hertzmann 2012 [29]	
Wu et al. 2013 [30]	
Gatys et al. 2016 [1]	
Selim et al. 2016 [31]	
Fišer et al. 2017 [32]	

Figure A1. Oil paint images in *SynthSet*.



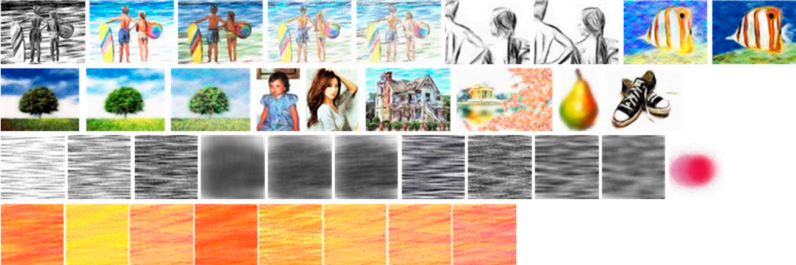

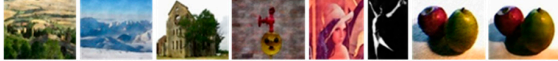



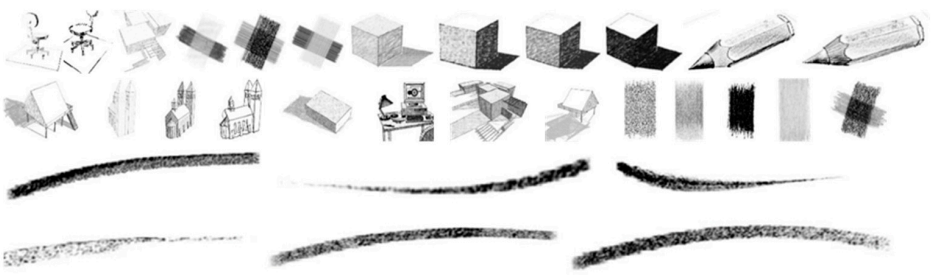

Literature	Synthesized pastel artwork images
Rudolf 2005 [33]	
Fišer et al. 2017 [32]	
Yang and Min 2017 [34]	
Literature	Synthesized watercolor artwork images
Hays and Essa 2004 [23]	
Bousseau et al. 2006 [49]	
Bousseau et al. 2007 [50]	
Wang et al. 2014 [51]	
Fišer et al. 2017 [32]	
Literature	Synthesized pencil artwork images (1)
Sousa and Buchanan 1999A [35]	
Sousa and Buchanan 1999B [36]	

Figure A2. Pastel, watercolor and pencil images in SynthSet.

Literature	Synthesized pencil artwork images (2)
Takagi et al. 1999[37]	
Lake et al. 2000 [38]	
Mao et al. 2002 [39]	
Yamamoto et al. 2004A [40]	
Yamamoto et al. 2004B [41]	
Matsui et al. 2005 [42]	
Lee et al. 2006 [43]	
Xie et al. 2007 [44]	
Xie et al. 2010 [45]	
Hata et al. 2012 [46]	
Lu et al. 2012 [47]	
Yang et al. 2012 [48]	
Fišer et al. 2017 [32]	

Figure A3. Pencil images in SynthSet.

References

1. Gatys, L.; Ecker, A.; Bethge, M. Image style transfer using convolutional neural networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2414–2423.
2. Keren, D. Painter identification using local features and naive bayes. In Proceedings of the International Conference on Pattern Recognition 2002, Quebec City, QC, Canada, 11–15 August 2002; pp. 474–477.
3. Li, J.; Wang, J. Studying digital imagery of ancient paintings by mixtures of stochastic models. *IEEE Trans. Image Process.* **2004**, *13*, 340–353. [[CrossRef](#)] [[PubMed](#)]

4. Lyu, S.; Rockmore, D.; Farid, H. A digital technique for art authentication. *Proc. Natl. Acad. Sci. USA* **2004**, *101*, 17006–17010. [[CrossRef](#)] [[PubMed](#)]
5. Johnson, J.; Alahi, A.; FeiFei, L. Perceptual losses for real-time style transfer and super-resolution. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 694–711.
6. Shamir, L.; Macura, T.; Orlov, N.; Eckley, D.; Goldberg, I. Impressionism, expressionism, surrealism: Automated recognition of painters and schools of art. *ACM Trans. Appl. Percept.* **2010**, *7*, 8. [[CrossRef](#)]
7. Liu, G.; Yan, Y.; Ricci, E.; Yang, Y.; Han, Y.; Winkler, S.; Sebe, N. Inferring painting style with multi-task dictionary learning. In Proceedings of the International Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015; pp. 2162–2168.
8. Mensink, T.; van Gemert, J. The rijksmuseum challenge: Museum-centered visual recognition. In Proceedings of the ACM International Conference on Multimedia Retrieval, Glasgow, UK, 1–4 April 2014; p. 451.
9. Karayev, S.; Trentacoste, M.; Han, H.; Agarwala, A.; Darrell, T.; Hertzmann, A.; Hertzmann, A.; Winnemoeller, H. Recognizing image style. In Proceedings of the British Machine Vision Conference, Nottingham, UK, 1–5 September 2014; pp. 1–20.
10. Tan, W.; Chan, C.; Aguirre, H.; Tanaka, K. Ceci n'est pas une pipe: A deep convolutional network for fine-art paintings classification. In Proceedings of the IEEE International Conference on Image Processing, Phoenix, AZ, USA, 25–28 September 2016; pp. 3703–3707.
11. Strezoski, G.; Worring, M. Omniart: Multi-task deep learning for artistic data analysis. *arXiv* **2017**, arXiv:1708.00684.
12. Matsuo, S.; Yanai, K. Cnn-based style vector for style image retrieval. In Proceedings of the ACM International Conference on Multimedia Retrieval, New York, NY, USA, 6–9 June 2016; pp. 309–312.
13. Chu, W.T.; Wu, Y.L. Deep correlation features for image style classification. In Proceedings of the ACM International Conference on Multimedia, Amsterdam, The Netherlands, 15–19 October 2016; pp. 402–406.
14. Sun, T.; Wang, Y.; Yang, J.; Hu, X. Convolution neural networks with two pathways for image style recognition. *IEEE Trans. Image Process.* **2017**, *26*, 4102–4113. [[CrossRef](#)]
15. Lu, X.; Lin, Z.; Shen, X.; Mech, R.; Wang, J.Z. Deep multipatch aggregation network for image style, aesthetics, and quality estimation. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 990–998.
16. Anwer, R.; Khan, F.; van de Weijer, J.; Laaksonen, J. Combining holistic and part-based deep representations for computational painting categorization. In Proceedings of the ACM International Conference on Multimedia Retrieval, New York, NY, USA, 6–9 June 2016; pp. 339–342.
17. Peng, K.C.; Chen, T. A framework of extracting multi-scale features using multiple convolutional neural networks. In Proceedings of the International Conference on Multimedia and EXPO, Turin, Italy, 29 June–3 July 2015; pp. 1–6.
18. Yang, H.; Min, K. Classification of basic artistic media based on a deep convolutional approach. *Vis. Comput.* **2019**, 1–20. [[CrossRef](#)]
19. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
20. Wilson, A.C.; Roelofs, R.; Stern, M.; Srebro, N.; Recht, B. The marginal value of adaptive gradient methods in machine learning. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 4148–4158.
21. Litwinowicz, P. Processing images and video for an impressionist effect. In Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 3–8 August 1997; pp. 407–414.
22. Hertzmann, A. Painterly rendering with curved brush strokes of multiple sizes. In Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, Orlando, FL, USA, 19–24 July 1998; pp. 453–460.
23. Hays, J.; Essa, I. Image and video based painterly animation. In Proceedings of the Non-Photorealistic Rendering and Animation, Annecy, France, 7–9 June 2004; pp. 113–120.
24. Kagaya, M.; Brendel, W.; Deng, Q.; Kesterson, T.; Todorovic, S.; Neill, P.; Neill, P.J.; Zhang, E. Video painting with space-time varying style parameters. *IEEE Trans. Vis. Comput. Graph.* **2011**, *17*, 74–87. [[CrossRef](#)]

25. Zeng, K.; Zhao, M.; Xiong, C.; Zhu, S. From image parsing to painterly rendering. *ACM Trans. Graph.* **2009**, *21*, 2. [[CrossRef](#)]
26. Lin, L.; Zeng, K.; Lv, H.; Wang, Y.; Xu, Y.; Zhu, S. Painterly animation using video semantics and feature correspondence. In Proceedings of the Non-Photorealistic Rendering and Animation, Annecy, France, 7–10 June 2010; pp. 73–80.
27. Zhao, M.; Zhu, S. Sisley the abstract painter. In Proceedings of the Non-Photorealistic Rendering and Animation, Annecy, France, 7–10 June 2010; pp. 99–107.
28. Zhao, M.; Zhu, S. Portrait painting using active templates. In Proceedings of the Non-Photorealistic Rendering and Animation, Vancouver, BC, Canada, 5–7 August 2011; pp. 117–124.
29. O'Donovan, P.; Hertzmann, A. Anipaint: Interactive painterly animation from video. *IEEE Trans. Vis. Comput. Graph.* **2012**, *18*, 475–487. [[CrossRef](#)]
30. Wu, Y.; Tsai, Y.; Lin, W.; Li, W. Generating pointillism paintings based on seurat's color composition. *Comput. Graph. Forum* **2013**, *32*, 153–162. [[CrossRef](#)]
31. Selim, A.; Mohamed, E.; Linda, D. Painting style transfer for head portraits using convolutional neural networks. *ACM Trans. Graph.* **2016**, *35*, 129. [[CrossRef](#)]
32. Fiser, J.; Jamriska, O.; Simons, D.; Shechtman, E.; Lu, J.; Asente, P.; Lukáč, M.; Sýkora, D. Example-based synthesis of stylized facial animations. *ACM Trans. Graph.* **2017**, *36*, 155. [[CrossRef](#)]
33. Rudolf, D.; Mould, D.; Neufeld, E. A bidirectional deposition model of wax crayons. *Comput. Graph. Forum* **2005**, *24*, 27–39. [[CrossRef](#)]
34. Yang, H.; Min, K. A multi-layered framework for color pastel painting. *KSII Trans. Internet Inf. Syst.* **2017**, *11*, 3143–3165.
35. Sousa, M.C.; Buchanan, J. Computer-generated graphite pencil rendering of 3d polygonal models. *Comput. Graph. Forum* **1999**, *18*, 195–208. [[CrossRef](#)]
36. Sousa, M.C.; Buchanan, J. Observational model of blenders and erasers in computer-generated pencil rendering. *Graph. Interface* **1999**, *99*, 157–166.
37. Takagi, S.; Nakajima, M.; Fujishiro, I. Volumetric modeling of colored pencil drawing. In Proceedings of the Seventh Pacific Conference on Computer Graphics and Applications (Cat. No. PR00293), Seoul, Korea, 7 October 1999; pp. 250–258.
38. Lake, A.; Marshall, C.; Harris, M.; Blackstein, M. Stylized rendering techniques for scalable real-time 3d animation. In Proceedings of the 1st International Symposium on Non-Photorealistic Animation and Rendering, Annecy, France, 5–7 June 2000; pp. 13–20.
39. Mao, X.; Nagasaka, Y.; Imamiya, A. Automatic generation of pencil drawing using lic. In Proceedings of the ACM Siggraph 2002 Conference Abstracts and Applications, San Antonio, TX, USA, 21–26 July 2002; p. 149.
40. Yamamoto, S.; Mao, X.; Imamiya, A. Enhanced lic pencil filter. In Proceedings of the International Conference on Computer Graphics, Imaging and Visualization, Penang, Malaysia, 2 July 2004; pp. 251–256.
41. Yamamoto, S.; Mao, X.; Imamiya, A. Colored pencil filter with custom colors. In Proceedings of the 12th Pacific Conference on Computer Graphics and Applications, Seoul, Korea, 6–8 October 2004; pp. 329–338.
42. Matsui, H.; Johan, H.; Nishita, T. Creating colored pencil images by drawing strokes based on boundaries of regions. In Proceedings of the International 2005 Computer Graphics, Stony Brook, NY, USA, 22–24 June 2005; pp. 148–155.
43. Lee, H.; Kwon, S.; Lee, S. Real-time pencil rendering. In Proceedings of the 4th International Symposium on Non-Photorealistic Animation and Rendering, Annecy, France, 5–7 June 2006; pp. 37–45.
44. Xie, D.; Zhao, Y.; Xu, D.; Yang, X. Convolution filter based pencil drawing and its implementation on gpu. *Lect. Notes Comput. Sci.* **2007**, *4847*, 723–732.
45. Xie, D.; Xuan, Y.; Zhang, Z. A colored pencil-drawing generating method based on interactive colorization. In Proceedings of the 2010 International Conference on Computing, Control and Industrial Engineering, Wuhan, China, 5–6 June 2010; pp. 166–169.
46. Hata, M.; Toyoura, M.; Mao, X. Automatic generation of accentuated pencil drawing with saliency map and lic. *Vis. Comput.* **2012**, *28*, 657–668. [[CrossRef](#)]
47. Lu, C.; Xu, L.; Jia, J. Combining sketch and tone for pencil drawing production. In Proceedings of the Symposium on Non-Photorealistic Animation and Rendering, Annecy, France, 4–6 June 2012; pp. 65–73.

48. Yang, H.; Kwon, Y.; Min, K. A stylized approach for pencil drawing from photographs. *Comput. Graph. Forum* **2012**, *31*, 1471–1480. [[CrossRef](#)]
49. Bousseau, A.; Kaplan, M.; Thollot, J.; Sillion, F. Interactive watercolor rendering with temporal coherence and abstraction. In Proceedings of the Non-Photorealistic Rendering and Animation, Annecy, France, 5–7 June 2006; pp. 141–149.
50. Bousseau, A.; Neyret, F.; Thollot, J.; Salesin, D. Video watercolorization using bidirectional texture advection. *ACM Trans. Graph.* **2007**, *26*, 104. [[CrossRef](#)]
51. Wang, M.; Wang, B.; Fei, Y.; Qian, K.; Chen, W.W.J.; Yong, J. Towards photo watercolorization with artistic verisimilitude. *IEEE Trans. Vis. Comput. Graph.* **2014**, *20*, 1451–1460. [[CrossRef](#)] [[PubMed](#)]
52. Ng, A. Machine Learning Yearning. 2017. Available online: <http://mlyearning.org> (accessed on 1 October 2018).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).