Authenticating Chinese Painting (Draft)

Mason Lin and **Leying Hu**
{ml4046,lh2871}@columbia.edu
Columbia University
New York, NY 10025
U.S.A.

Abstract

Art forging has been a historical issue. It is always hard for people to distinguish the authentic paintings from the fake ones. Since producing fake arts makes great profits, art appraiser themselves will participate in the forging process. Forgers went deep into the painters' styles, stamps, inscriptions, paper, and ink. Their fake paintings as long as the fake collection certificates increase the difficulty of art authentications.

With the development of the technology, fake arts are easier to produce. From this perspective, the team decides to analyze the brush strokes. The strokes, as the subconsciousness of the artists, are unlikely to change even if there are dramatic changes in styles. The team proposes that Kernel Image Processing (KIP), Wavelet Transformation (WT), and Convolutional Neural Networks (CNNs) are powerful tools for the art authentication.

Section 5 discusses our current challenges.

The Github repository can be found at `https://github.com/Kirstihly/A8_ml4046_lh2871`.

1 Introduction

With the advance of Kaggle, Sklearn, TensorFlow, the team is able to analyze the massive amount of image datasets. These sources provide numerous digital image processing and machine learning algorithms for convenient applications. However, the authentication faces many difficulties. For example, artists may change their styles in their lifetimes. Sometimes, famous artists have assistants to help complete their drawings. In addition, contemporary artists begin to focus on multimedia, abstraction, or consciousness. They overlook the skills training so that it is hard to perceive the characteristics from the drawings. State-of-Art techniques include classifying different art styles from a single art piece (Lyu, Rockmore, and Farid 2004), demonstrating using the visual characteristics (Noord, Hendriks, and Postman 2015), and maximizing classification accuracy (Viswanathan).

2 Background

The goal of this project is to identify Chinese painting forgeries. Currently, experiments are conducted on a dataset containing a mix of Western art of all genres, allowing us to begin the data processing and feature extraction stages before we receive our data on Chinese paintings. This section discusses our dataset, data processing, feature extraction, and model selection.

Dataset and Data Processing

Currently we are experimenting with a subset of data on various artworks from Kaggle, compiled from Wikiart (Kaggle). The subset we are working with contains 5000 artworks consist of paintings of all genres from 1178 artists. Furthermore, the paintings also have varying sizes.

Each image is imported as a matrix in its spatial domain, converted to gray scale.Features are then extracted in the frequency domain and other transformation.

In addition, paintings from the Artchive are used for training Bag of Visual Words. The paintings provided by Artchive are scanned with the high definition. They are cropped to train for the object detection.

Feature Extraction

Kernel Image Processing (KIP) A Kernel matrix in image processing is a filter to convolve with an image to perform the edge detection. Changing the elements of the matrix leads to sharpening, enhancing, blurring, and embossing. The reason is that the Kernel operators are used as high-pass or low-pass filters.

The team is applying Sobel Operator to create an image emphasizing edges. It is an isotropic 3x3 image gradient matrix. The elements of the matrix are gradients of the original image intensity. The directional edge detector has high-pass in one direction and low-pass in the orthogonal direction. It is particularly useful for high-frequency variations in the image.

Define A as the source image, G_x as the horizontal derivative, and G_y as the vertical derivative. (Gonzalez and Woods 2018)

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * A \text{ and } G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

The gradient magnitude can be obtained using,

$$G_m = \sqrt{G_x^2 + G_y^2}$$

Setting a threshold to be the edge pixel value to ensure a good result. Saving the edge map values larger than the threshold to be 255 and those smaller than the threshold to be 0. The threshold can be determined by a certain percentage of the pixels.

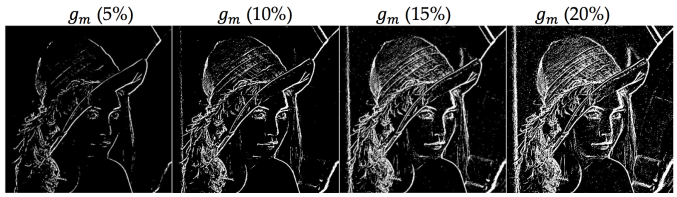

Figure 1: Gradient magnitudes with different thresholds

Haar Wavelet Transform The wavelet transform allows us to decompose a 2-dimensional pixel array into multiple subband domain, represented by basis functions (wavelets). Specifically, we applied the Haar wavelet for our decomposition. The subbands, Vertical, Horizontal, and Vertical, are the result of applying a low-pass and high-pass filter across our image in the spatial domain. The decomposition is applied recursively for each scale of $i \in 1, 2, ..., n$. Fig. 2 provides an idea of how a domain is decomposed.

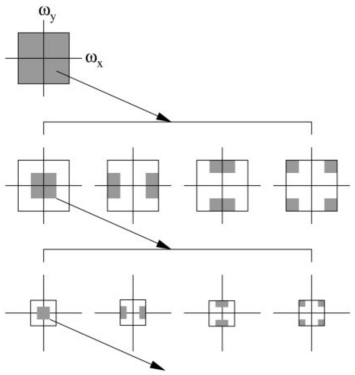

Figure 2: Ideal decomposition of a frequency domain at multiple scale. Decomposition is performed recursively for the low-pass subband. Each decomposition results in low-pass, vertical, horizontal, and diagonal domain (from left to right).

As suggested in (Lyu, Rockmore, and Farid 2004), the multiscale decomposition captures the textual details of the paintings by extracting subband coefficients in the frequency domain. Below is an example of a decomposition of a Chinese painting in Fig. 3 by a well-known artist. Fig. 4 is a level 1 decomposition using the Harr wavelet. Fig. 5 shows the differences in level $i = 1, 2, 3, 4$ decomposition of the diagonal subband.

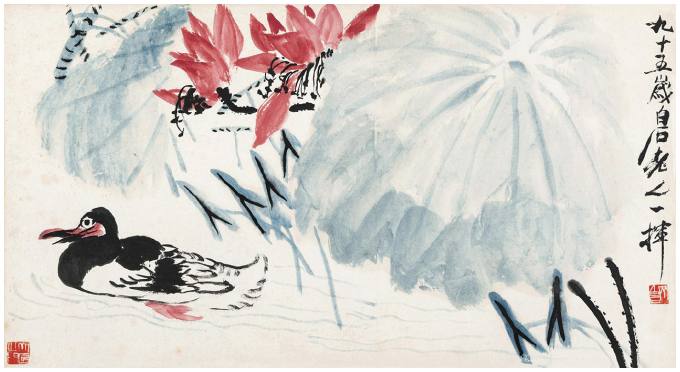

Figure 3: Chinese painting by Qi Baishi

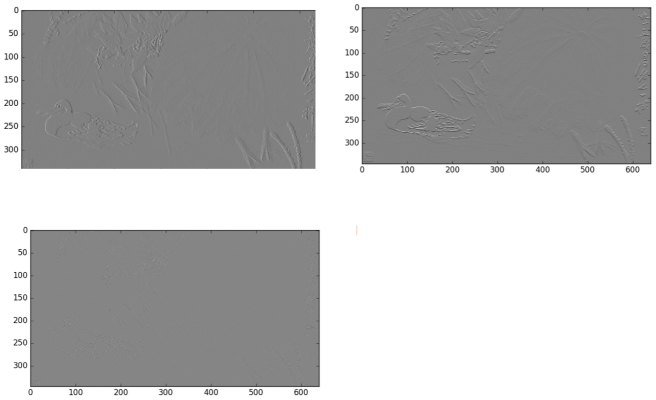

Figure 4: Clockwise: Vertical, Horizontal, Diagonal

From the diagonal decomposition, we can see that the details are more apparent for higher level decomposition. After decomposing the image into $i = 1, 2, 3, 4, 5$, we collected two sets of statistics (mean, variance, skew, kurtosis) based on the subbands coefficients and its neighbors. Let V_i, H_i, D_i denote the vertical, horizontal and diagonal subband respectively.

In the first set of statistics, we collected the statistics of V_i, H_i, D_i for each subband, giving us $12(n-2)$ statistics where $n = 5$ is the level of decomposition.

The second set of statistics is consisted of the prediction error of a subband coefficient based on its neighbors. As mentioned in (Lyu, Rockmore, and Farid 2004), subbands coefficients are related to its neighbors in both the current and higher level of subbands. Hence, we constructed a least-square fit based on a $3x3$ neighborhood for the magnitude of the subband coefficient. Consider the vertical subband V_i, its neighbors are given as:

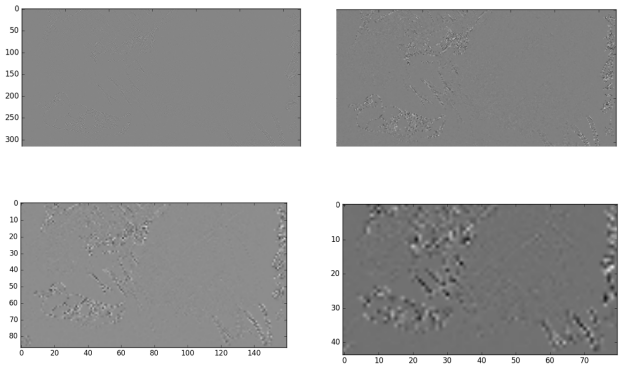

Figure 5: Clockwise: Diagonal subband for level $i = 1, 2, 3, 4$

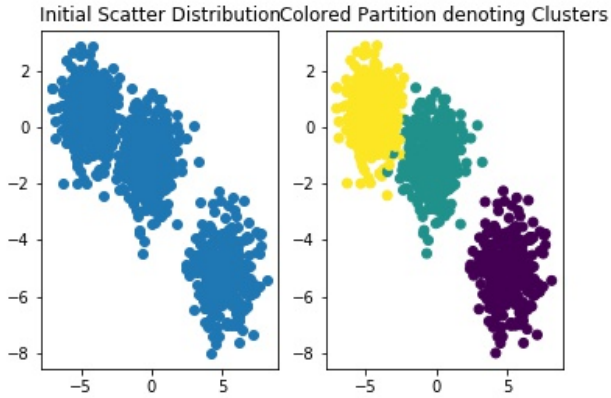

Figure 6: 3 Clusters on 1000 Data Points

$$V_i(x-c_x, y-c_y), H_i(x-c_x, y-c_y), D_i(x-c_x, y-c_y)$$
$$V_{i+1}(x/2-c_x, y/2-c_y), H_{i+1}(x/2-c_x, y-c_y), D_{i+1}(x/2-c_x, y/2-c_y)$$
$$V_{i+2}(x/4-c_x, y/4-c_y), H_{i+2}(x/4-c_x, y/4-c_y), D_{i+2}(x/4-c_x, y/4-c_y)$$

where $c_x, c_y \in -1, 0, 1$ and we only consider the magnitude of the coefficients. Integer division is considered for the location of neighbors in higher level.

Clustering Cluster analysis is an unsupervised learning method for grouping a set of objects which are similar. The objections in an image are differentiated and separated on the basis of color only. K means clustering is applied in the Bag of Visual Words model. Define $X = x_i, x_2, ..., x_n$ as a set of objects to be separated by k clusters. μ_i is the mean of each set of points S in each cluster.

$$\arg\min_s \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2$$

With the K means iteration, the ideal cluster centroids are determined when the distance of every data point inside its cluster is minimized.

Scale-invariant Feature Transform (SIFT) Applying clustering to Bag of Visual Words, the images are differentiated by their features. The SIFT algorithm uses the difference of Gaussian blurring to find the local maxima across the scale and the space. The local extrema are potential key points. The gradient and direction around the key points are calculated to form the orientation histogram. The higher peaks in the histogram are taken to ensure the scale-invariant.

To create key point descriptors, A 16x16 neighborhood around the key point is divided into 16 sub-blocks of 4x4. There is an 8 bin orientation histogram for each sub-block so a key point descriptor vector has 128 elements. Fig. 7 shows the key points and the gradient magnitude, orientation at each location.

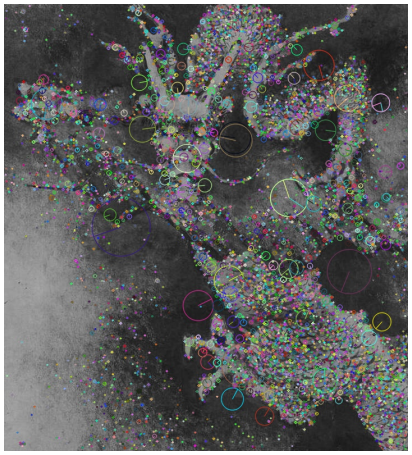

Figure 7: SIFT Transform: The size of the circle and the line depicts the gradient magnitude and orientation respectively

Tamura Features [include in final draft] **Support Vector Machine (SVM)** SVMs are supervised learning models that are very popular in the computer vision. The training sets are classified into two categories. The goal is to determine what a new data point belongs to.

Given the training data set $(\overrightarrow{x_1}, y_1), ..., (\overrightarrow{x_1}, y_n)$, where $\overrightarrow{x_i}$ is the point vector and y_i is the point value 1 or -1. The SVM is used for finding the hyperplane that divides the 1 and -1 groups. The distance between the hyperplane and the nearest $\overrightarrow{x_i}$ from either group is maximized.

Define $\overrightarrow{w}$ is the normal vector to the hyperplane, and b is the intercept. The hyperplane should satisfy $\overrightarrow{w} \times \overrightarrow{x} - b = 0$. The offset between the hyperplane and origin along $\overrightarrow{w}$ is $\frac{b}{|\overrightarrow{w}|}$.

Kernel Descriptors One challenge with histogram based

descriptors, such as SIFT, is that it is capturing low-level features for detecting objects in a painting. For a more robust descriptor, we turn to kernel descriptors, capturing the gradient, shape, and color of a painting (Bo, Ren, and Fox 2010). Note that one benefit with kernel descriptors is that they can be easily trained with SVM for classification.

Let K_{grad} denote the gradient kernel defined by:

$$K_{grad}(P, P') = \sum_{z \in P} \sum_{z' \in P'} \tilde{m}(z)\tilde{m}(z')k_o(\tilde{\theta}(z), \tilde{\theta}(z'))k_p(z, z')$$

where $z \in P$ is a pixel in image patch P and $\tilde{m}(z)$ and $\tilde{\theta}(z)$ are normalized gradient and orientation at pixel z. Furthermore, $k_o(\tilde{\theta}(z), \tilde{\theta}(z')) = exp(-\gamma_o||\tilde{\theta}(z) - \tilde{\theta}(z')||^2)$ is a Gaussian kernel over the gradient orientation. Similarly, $k_p(z, z')$ is also a Gaussian kernel over the spatial position of pixels, measuring how similar z and z' spatially.

Observe that K_{grad} is composed of three kernel functions and can be re-wrtten as the following inner products: $k_{\tilde{m}}(z, z') =< \phi_{\tilde{m}}(z), \phi_{\tilde{m}}(z') >$, $k_o(z, z') =< \phi_o(\tilde{\theta}(z), \phi_o(\tilde{\theta}(z') >$, $k_p(z, z') =< \phi_p(z), \phi_p(z') >$ where $\phi_{\tilde{m}}(.)$, $\phi_o(.)$, $\phi_p(.)$ are the feature maps for the corresponding kernels. Note that $k_{\tilde{m}}(z, z')$ is a linear kernel and $\phi_{\tilde{m}}(z) = \tilde{m}(z)$ is simply the normalized gradient magnitude. The descriptor can be extracted as:

$$F_{grad}(P) = \sum_{z \in P} \phi_{\tilde{m}}(z)\phi_o(\tilde{\theta}(z) \otimes \phi_p(z)$$

Adapting from the $K_{grad}(P, P')$, $K_{shape}(P, P')$, $K_{color}(P, P')$, the authors (Karmakar et al. 2017) constructed Tamura texture kernel descriptors using similar derivation [include all descriptors in final draft].

Since both $k_o(z, z')$ and $k_p(z, z')$ are non-linear (Gaussian) kernel with infinite dimension, the corresponding feature maps has to be approximated. The author in (Bo, Ren, and Fox 2010) proposed approximating $\phi_o(.)$ and $\phi_p(.)$ by projecting them on to a set of basis vectors X where $x_i \in X$ is a sampled normalized gradient vectors. The intuition is that pixel attributes are low-dimension and good approximation can be achieved if we sample enough basis vectors. However, a specified support region must be given in order to sample the set of basis vectors. We turn to approximating the Gaussian kernel with random feature mapping.

Orthogonal Random Feature Map Kernel approximation is an essential tool to make kernel methods scalable. The Gaussian kernel can be approximated with Random Fourier features where the feature map can be extracted as $\phi(\boldsymbol{z}) = \sqrt{\frac{1}{D}}[sin(\boldsymbol{w_1} \cdot \boldsymbol{x}), ..., sin(\boldsymbol{w_D} \cdot \boldsymbol{x}), cos(\boldsymbol{w_1} \cdot \boldsymbol{x}), ..., cos(\boldsymbol{w_D} \cdot \boldsymbol{x})]^T$ with each $\boldsymbol{w_i}$ is sampled i.i.d. from the Gaussian distribution (**?**). The linear transformation $\boldsymbol{Wx}$ where $\boldsymbol{W} = \{\boldsymbol{w_i}\}_{i=1}^{d}$ is essential for the kernel approximation as it determines whether the estimation converges to the desired kernel and how concentrated is the resulted estimation (variance of estimation). The author (Yu et al. 2016) proposed that imposing orthogonality on $\boldsymbol{W}$ achieves not only an unbiased estimator of the kernel but also better concentration result. The orthogonal matrix $\boldsymbol{W}$ is given as:

$$\boldsymbol{W_{orf}} = \frac{1}{\sigma}\boldsymbol{SQ}$$

where $\boldsymbol{Q}$ is a uniformly distributed orthogonal matrix and $\boldsymbol{S}$ is a diagonal matrix with entries sampled from χ^2 with D-degree of freedom. Note that $\boldsymbol{Q}$ can be constructed with QR-decomposition after sampling a matrix from $\mathcal{N}(0, 1)$ independently. The resulting matrix $\boldsymbol{Q}$ is distributed uniformly in the span of all orthogonal matrices based on the Bartlett decomposition theorem.

3 Experiment

Refer to Sec. 5 for current approach.

Bag of Visual Words(BOV)

BOV is applied to image classification and covers clustering, SIFT and SVM (Vyas). The team trained on features of birds, dragons, fish, etc. from famous Chinese paintings. The algorithm generated the vocabularies and the histogram is shown in Fig. 8.

Fig. 9 are good testing results for single feature images. Fig. 12 is the wrong classification. It is probably because the fish scales are very similar to dragon scales. There is another possibility that there are too many fish in one figure. The left of Fig. 9 is extracted from Fig. 12 and the prediction is right. Nevertheless, Fig. 10 and Fig. 11 are good testing results for multiple features images.

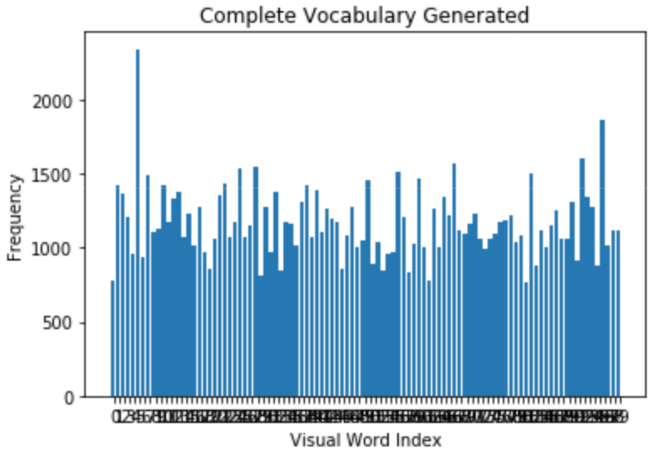

Figure 8: Statistics of the Vocabulary

4 Results

To be updated when the project is done at the end of April.

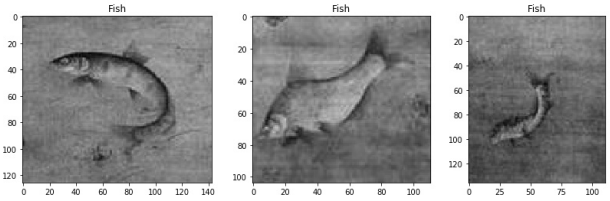

Figure 9: BOV Prediction of Fish from Fish

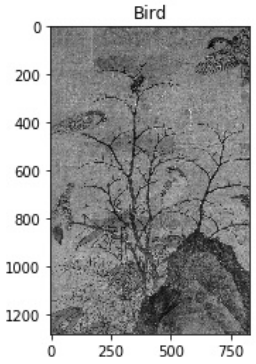

Figure 10: BOV Prediction of Bird from Bird

5 Conclusions

Challenges

Here we discuss a few challenges we have encountered. First, the least-square fit of the magnitudes of coefficients relative to a neighborhood described in Sec. 2 is computationally expensive even in Numpy. Therefore, we have omitted this in feature extraction.

Currently, our goal is classifying the painting's author. We used a SVM with a RBF-kernel for classification. As we are still in the training phase, train/test split was not performed on our training set. Instead, we performed a $k = 20$-fold cross validation on our data set. Unfortunately, the model did not performed poorly. This could be due to the fact that paintings were not normalized on a uniform scale before feature extraction, as suggested in (Stanford).

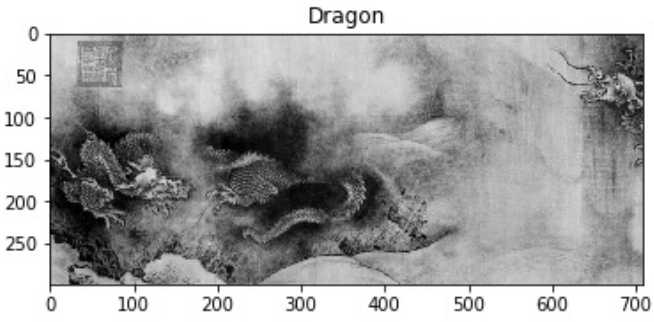

Figure 11: BOV Prediction of Dragon from Dragon

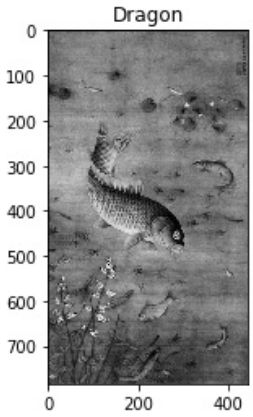

Figure 12: BOV Prediction of Dragon from Fish

To address this problem, we will look into segmenting the each painting into a fixed-size, non-overlapping region before wavelet decomposition. This allows us to explore multiple regions of the painting. As suggested before, normalizing the dimension could also benefit our prediction. We will also explore incorporating a CNN with wavelet decomposition layers as suggested in (Fujieda, Takayama, and Hachisuka 2017). The result shows that it has benefits over conventional CNN in textual analysis.

Lastly, we will need to consider whether to take a quantitative or qualitative approach in our analysis. Since the number of authentic and forged Chinese painting is limited, ML and DL approaches may not be appropriate as due to a small set of data. Hence, qualitative approaches would be based more on frequency analysis and other signal processing methods.

6 Contributions

To be updated when the project is done at the end of April.

7 Acknowledgements

This research is a Advanced Big Data Analytics provided by Professor Ching-Yung Lin, Columbia University.

References

Bo, L.; Ren, X.; and Fox, D. 2010. Kernel descriptors for visual recognition. In Lafferty, J. D.; Williams, C. K. I.; Shawe-Taylor, J.; Zemel, R. S.; and Culotta, A., eds., *Advances in Neural Information Processing Systems 23*. Curran Associates, Inc. 244–252.

Fujieda, S.; Takayama, K.; and Hachisuka, T. 2017. Wavelet Convolutional Neural Networks for Texture Classification. *ArXiv e-prints*.

Gonzalez, R. C., and Woods, R. E. 2018. *Digital Image Processing*. Pearson. chapter 3.

Kaggle. Painter's by numbers. https://www.kaggle.com/c/painter-by-numbers.

Karmakar, P.; Teng, S. W.; Zhang, D.; Liu, Y.; and Lu, G. 2017. Improved tamura features for image classification using kernel based descriptors. In *2017 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2017, Sydney, Australia, November 29 - December 1, 2017*, 1–7.

Lyu, S.; Rockmore, D.; and Farid, H. 2004. A digital technique for art authentication. *Proceedings of the National Academy of Sciences* 101(49):17006–17010.

Noord, N. v.; Hendriks, E.; and Postman, E. 2015. Signal Processing for Art Investigation. *IEEE SIGNAL PROCESSNG MAGAZINE* 1053-5888(15):46–54.

Stanford. Using machine learning for identification of art paintings. http://stanford.io/2FMWNyx.

Viswanathan, N. Artist identification with convolutional neural networks.

Vyas, K. Bag of visual words model for image classification and recognition.

Yu, F. X.; Suresh, A. T.; Choromanski, K. M.; Holtmann-Rice, D. N.; and Kumar, S. 2016. Orthogonal random features. In Lee, D. D.; Sugiyama, M.; Luxburg, U. V.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc. 1975–1983.