Automated Identification of Artist Given Unknown Paintings & Quantification of Artistic Style

Nicholas Dufour, Kyle Griswold, Michael Lublin
{ndufour, kggriswo, mlublin}@stanford.edu

December 13, 2014

1 Introduction

Certain painters have visual styles that make them immediately identifiable, even to a layperson. In this paper, we examine whether artists' features that are obvious to the human eye can be encoded using modern visual features and learned by existing machine learning algorithms. Our goals here are threefold:

1. Show that our featureset captures information about artistic style (i.e., properties consistent within artist and across paintings) by demonstrating it can be used for significantly above chance prediction of painting authorship.
2. Show that the features are similar to those humans use for the same task by demonstrating that algorithmic confusion between two artists is predictive of human assessment of similarity.
3. Use differences in classification accuracy as a function of hyperparameters to study properties of artistic style.

Classifying paintings by artist is a task usually left to experts, but recently the machine learning literature has begun to approach this problem—especially as it pertains to forgery detection [6].

2 Data

Our dataset consisted of 752 paintings from 6 different artists, with each painting labelled with its artist. Table 1 gives an overview of our dataset, including the number of paintings per artist. Paintings were scraped from online sources (Wikipedia, Google, etc) under Fair Use. Any images that included elements not chosen by the artist (i.e., picture frame, monochromatic borders, captions) were discarded. Data were divided into training (64%), dev (16%), and test (20%) sets.

Table 1: Summary of Artists Used

Artist	Style	Birth	Country	N Images
Alfred Sisley	Impressionism	1839	Britain	137
Hans Holbein	Northern Renaissance	c. 1497	Germany	116
John Millais	Pre-Raphaelitism	1829	Britain	121
Claude Monet	Impressionism	1840	France	135
Rembrandt	Dutch Baroque	1606	Netherlands	108
Vincent van Gogh	Post-Impressionism	1853	Netherlands	135

These artists were chosen because they represent several important artistic styles. However, we also include multiple painters from the same style so that our classification objective is not simply to classify paintings by genre, but to distinguish among the artists of a genre.

3 Methods

Figure 1 summarizes our feature extraction pipeline visually. A detailed explanation follows (for image sources, see[1][2]).

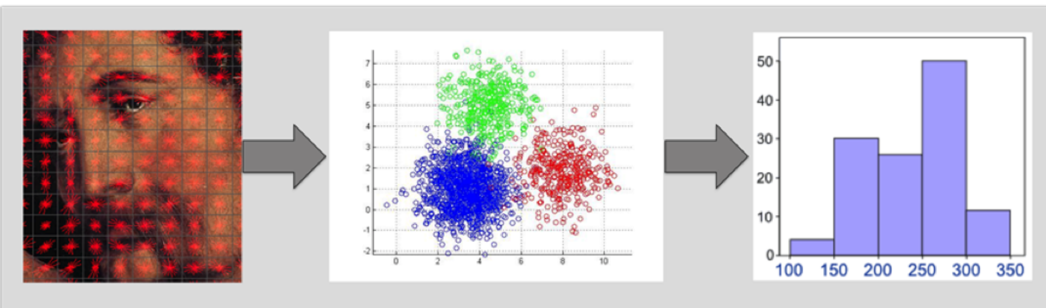

Figure 1: The three steps of feature extraction are shown: HOG descriptors are computed, clustered via k means, and then the image is assigned catenated spatial pyramid histograms. The first image is the author's own work (*Adam and Eve*, Holbein, 1517, with HOG descriptors overlayed). The second and third images are from web sources.

[1] http://www.mathworks.com/matlabcentral/fileexchange/screenshots/6432/original.jpg
[2] http://www.mathsisfun.com/data/images/histogram-heights.gif

3.1 Feature Extraction

Images are converted to grayscale and then downsampled using linear interpolation. Grayscale was chosen because it allows classification based purely on image structure as opposed to color composition, given that color composition is not natively considered in the computation of our features and would greatly expand the featurespace.

3.1.1 HOG Features

HOG features were selected as the primary source of image features, given their robustness in a variety of contexts [2, 5]. The image is divided into a set of 2x2 pixel cells. Two 1-dimensional gradient mask kernels are applied to each pixel, which permits calculation of gradient intensities:

$$[-1, 0, 1] \text{ and } [-1, 0, 1]^T$$

Each pixel in the cell then casts a vote for each of the 8 evenly-spaced (from 0 to 360 degrees) gradient "channels" (i.e., orientations). Gradients are normalized locally within 8x8 pixel blocks, each of which contains 16 cells. While other cell and block sizes have shown differential performance on certain tasks (i.e., human recognition), given the abstract nature of the task there was no clear method of selecting them besides testing on the dev set, which would have greatly expanded our hyperparameter space. Blocks were spatially contiguous but nonoverlapping, and image downsampling was such that each image yielded an array of 64x64 HOG descriptors.

3.1.2 k-Means Clustering

HOG descriptors were transformed from continuous measures into discrete labels via k-Means Clustering, which essentially transforms them into visual "words." k-Means Clustering proceeds by constructing k non-overlapping sets of HOG descriptors H_i where each HOG descriptor set has an average point μ_i in descriptor space such that:

$$\arg\min_{H} \sum_{i=1}^{k} \sum_{hog \in H_i} \|hog - \mu_i\|^2$$

i.e., the sum of the squared L^2 norm for every HOG descriptor to the mean point for each set, in every set of descriptors, is minimized. k was treated as a hyperparameter and tuned using a dev set.

3.1.3 Spatial Histograms

The final image-level descriptor X was produced by dividing the image into quadrants and subquadrants n times, such that 4^n quadrants were produced in total, arrayed in a uniform grid over the image. A histogram of k-Means labeled HOG features in each subquadrant was computed as a vector, resulting in a matrix $X \in \mathbb{N}^{n \times n \times k}$ such that $X_{i,j,m}$ is the number of times a HOG descriptor was labeled m in subquadrant (i, j). This matrix was then flattened into a vector by stacking the histograms for each subquadrant on top of each other. The final feature vector V for each image was thus a vector of positive integers where $|V| = k * 4^n$.

3.2 Machine Learning Algorithms

Three algorithms were tested. Each was considered as a separate experiment, given their disaparate qualities.

3.2.1 k-Nearest Neighbor (kNN)

k-Nearest Neighbor is the simplest classification method used; each training example is assigned a point in feature space. For a test example x, the algorithm merely finds the k closest points to x, and combines their labels according to some function (in the case of classification, taking the plurality).

3.2.2 Random Forests (RF)

Random Forests have previously been shown to be successful at various vision tasks [3, 1]. RF trains a set of T decision trees (the number of which is treated and tuned as a hyperparameter) of some maximum depth (treated and tuned as a hyperparameter) on random subsets of the data. If we let $p_t(x) = a$ be the prediction a from the set of artists A of tree b for unseen example x, then computing the predicted classification is merely taking the mode:

$$\arg\max_{a \in A} \sum_{t=1}^{T} \mathbb{1}[p_t(x) = a]$$

3.2.3 Multi-Class Support Vector Machine (SVM)

Mutli-Class SVMs are the most mathematically complex method employed. This method constructs a series of hyperplanes in high (in our case, infinite) dimensional space, and uses the position of the test datapoints relative to these hyperplanes to classify them. For our purposes, we used a Radial Basis (i.e., Gaussian) kernel, given by

$$K(x_1, x_2) = \exp(-\gamma |x_1 - x_2|^2)$$

where γ is treated and tuned as a hyperparameter. Our SVM was amongst multiple classes, and implemented a "one-against-one" [4, 7] method, in which $N \times (N-1)/2$ classifiers are constructed for all binary pairwise combinations of the N classes (in this case, $N = |A|$. Each binary classifier solves the dual problem

$$\min_{\alpha} \frac{1}{2}\alpha^T Q\alpha - \sum_{i=1}^{|\alpha|} \alpha_i$$

s.t. $y^T\alpha = 0$ and $0 \leq \alpha_i \leq C, i = 1, ..., |\alpha|$ where $Q_{ij} = K(x_i, x_j)$ and C is an upper bound on values of α. Individual decisions are made according to (for testing example x^*):

$$\mathrm{sgn}(\sum_{i=1}^{n} y_i\alpha_i K(x_i, x^*) + \rho)$$

Where ρ is an intercept term. Each of the $N \times (N-1)/2$ classifiers makes such a classification, which are combined according to a majority rule, similar to the RF.

4 Results

4.1 Classification

Each classification method was trained on the training set. Hyperparameters (number of clusters, maximum depth of decision trees, etc.) were optimized using the dev set. The results reported here correspond to the performance of the indicated classification method on the unseen test set using hyperparameters found to be optimal via the dev set.

All classification methods performed remarkably well, and far above chance (one-tailed z-test $p < 0.001$ for all methods). Hit rates for each classifier are summarized in Figure 2.

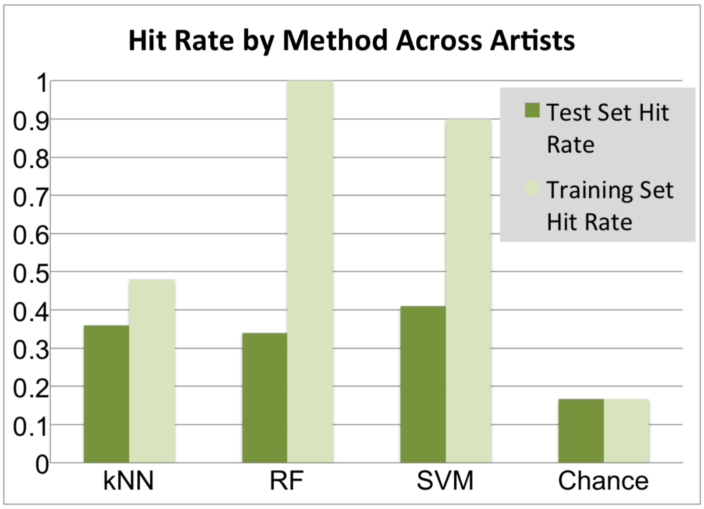

Figure 2: Hit Rates For Each Model

The SVM performed best with a hit rate of 41%, followed by kNN (36%) and RF (34%). Random chance is also shown, at 17%. Additionally, SVM and RF perform extremely well on the training set—likely due to overfitting. Though much work was done in the feature extraction and hyperparameter optimization steps to reduce variance, we can see that variance remains high for the RF and SVM classifiers (this is discussed more in the Future Work section).

A confusion matrix for the SVM classifier is depicted in Table 2a; Table 2b summarizes this data into counts of true positives, false positives, true negatives, and false negatives for each artist, as well as calculates the classification acccuracy for each artist.

Table 2a: Confusion Matrix.

N=143	Alfred Sisley	Hans Holbein	John Millais	Claude Monet	Rembrandt	van Gogh
Alfred Sisley (27)	14	0	0	7	2	4
Hans Holbein (22)	1	12	6	1	2	0
John Millais (23)	0	9	6	3	0	5
Claude Monet (26)	10	1	1	10	0	4
Rembrandt (19)	4	1	2	2	8	2
Van Gogh (26)	5	1	4	5	2	9

Table 2b: Confusion Matrix Statistics.

Artist	True Positives	False Positives	False Negatives	True Negatives	Accuracy
Alfred Sisley	14	20	13	96	.77
Hans Holbein	12	12	10	109	.85
John Millais	6	13	17	107	.79
Claude Monet	10	18	16	99	.76
Rembrandt	8	6	11	118	.88
Vincent van Gogh	9	15	17	102	.78

Our methods are substantially more effective if we classify artists one-vs-one (detailed in Table 3).

Table 3: Maximum 1 vs. 1 Hit Rate (%)

	Holbein	Sisley	Monet	Millais	Van Gogh	Rembrandt
Holbein		95.9	90.0	71.2	84.4	76.9
Sisley	95.9		80.9	91.8	83.3	79.6
Monet	90.0	80.9		88.0	88.4	78.0
Millais	71.2	91.8	88.0		88.9	75.0
Van Gogh	84.4	83.3	88.4	88.9		84.4
Rembrandt	76.9	79.6	78.0	75.0	84.4	

These results allow us to quantify how similar artists are (according to our features) by looking at how frequently the paintings of artist i are classified as artist j. We can see that artists with similar styles are confused more frequently than artists with disparate styles, thus lending more support to the hypothesis that our featureset captures information about artistic style.

4.2 Coreference Analysis

Coreference frequency (number of websites mentioning both artist i and artist j) did not correlate significantly with algorithmic confusion ($r = 0.04$, $p = 0.81$), although this could be artifactual due to difficulties in obtaining coreference frequencies.

4.3 Analysis of Artistic Style

There are stylistic interpretations of the hyperparameters used in our feature extraction which can be used to study properties of artistic style. The number of clusters in our feature extraction relates the amount of textural information preserved in the feature vector, because more clusters allows the model to distinguish between more classes of HOGs (which correspond to specific methods of painting and other such textural information). In addition, the number of spatial levels in the feature extraction relates the amount of spatial information preserved in the feature vector, because more spatial levels gives more information about the location of different elements in the painting. Therefore, the relevance of textural and spatial information to style by artist can be computed by averaging the optimal values of these hyperparameters across pairs of artists. The results of this analysis are detailed in Figure 3.

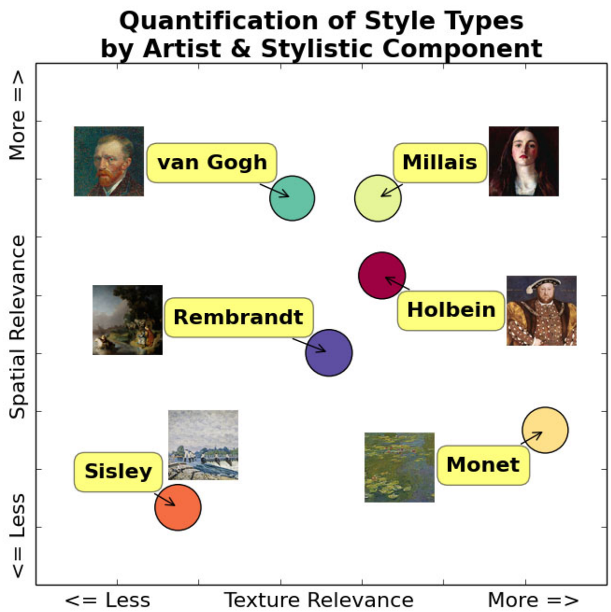

Figure 3: Textural and Spacial Relvence by Artist

The spatial and textural information found here is largely in line with what one would expect from these artists. This analysis also allows us to formalize notions of similarities and differences between artists. For example, the primarily portraiture artists Millais and Holbein both have relatively high levels of spatial and textural importance, as both the spatial information (i.e. where the different elements of the human body are in the portrait) and textural information (i.e. information about the brushstrokes themselves) play important roles in identifying their works. Another interesting example is Monet. Monet painted a variety of scenes (such as landscapes, portraits, etc), so one would expect that spatial information would not be very relevant in identifying his paintings, whereas he had a very intricate and distinctive style, so one would expect textural information to be very indicative of his work. Indeed, these are the results seen in our analysis.

5 Conclusions

HOG features are a significant part of what makes a painter's work identifiable as their own – the more than double increase in the probability of correct identification over random chance shows this. Furthermore, HOG features are also a part of what makes a work's style identifiable – as shown by artists who worked with similar styles having their paintings misclassified as each other's more often than other artists.

Perhaps the most exciting aspect of this work is its utility in quantifying aspects of artistic style. While we make no suggestion that artistic style may be boiled down entirely to numeric values, it is not unreasonable to suggest that some portion of it can be expressed as such. Further, by dividing the image up into variable numbers of cells and using variable numbers of clusters, the method affords fine control over the amount of spatial vs. textural information preserved in the final feature vector. This permits careful exploration and comparison of how artistic style is encoded in art.

6 Future Work

There are two major directions that this work can go in the future to continue to improve classification accuracy. The first is to improve the classification accuracy of our existing models by continuing to reduce the variance of the RF and SVM classifiers. This can be done by collecting more data to increase the size of our training and test sets, and also by continuing to shrink the feature set by experimenting with more values of our models' hyperparameters. Second, we can experiment with different types of features to see if other features capture more information about artistic style. For example, we can experiment with features involving colors, introduce deep learning generated features, experiment with the 2D Fourier Transform, or try Scale-Invariant Feature Transform (SIFT) features.

In addition, we can conduct further analyses relating our models' accuracies with the accuracies of expert humans to better understand the similarities between our features and the features humans use to distinguish between artists. Another possible direction would be to apply our techniques to the domain of forgery detection.

References

[1] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[2] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[3] Juergen Gall, Nima Razavi, and Luc Van Gool. An introduction to random forests for multi-class object detection. In *Outdoor and Large-Scale Real-World Scene Analysis*, pages 243–263. Springer, 2012.

[4] Stefan Knerr, Léon Personnaz, and Gérard Dreyfus. Single-layer learning revisited: a stepwise procedure for building and training a neural network. In *Neurocomputing*, pages 41–50. Springer, 1990.

[5] Ivan Laptev. Improving object detection with boosted histograms. *Image and Vision Computing*, 27(5):535–544, 2009.

[6] Gungor Polatkan, Sina Jafarpour, Andrei Brasoveanu, Shannon Hughes, and Ingrid Daubechies. Detection of forgery in paintings using supervised learning. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 2921–2924. IEEE, 2009.

[7] Ting-Fan Wu, Chih-Jen Lin, and Ruby C Weng. Probability estimates for multi-class classification by pairwise coupling. *The Journal of Machine Learning Research*, 5:975–1005, 2004.