Style-based image retrieval and inference of stylistic classes

James M. Hughes

May 31, 2011

Abstract

The availability of large collections of digital images via the Internet has necessitated the development of algorithms for searching for and retrieving relevant images, given a particular query. Traditionally, these algorithms have described images in terms of associated metadata (such as tags), as well as with content-based features and statistical descriptions. Furthermore, while the digitization of vast archives of cultural artifacts has made these objects more accessible both to the general public and to researchers, the increase in their availability requires more efficient means of locating objects relevant to a particular query. In the case of works of art, one obvious means of describing these objects is by statistical features derived from their digitized versions. These features could then be used to facilitate a *style-based image retrieval* system that returns stylistically relevant images with respect to a query image (or query string). I propose a system that implements similarity-based image search using a query image provided by the user, along with the ability to automatically learn the stylistic categories that describe style relationships between images from feedback on search results provided by the system.

Introduction

Increasingly, large collections of various types of images are being made accessible via the World Wide Web [1, 2]. Searching for images of a particular class is an important problem in machine learning that has received a great deal of attention [3, 4]. To this point, research has focused primarily on retrieving images based on object classes and the content present in images, known as content-based image

retrieval (CBIR) [3, 5]. In my project, I will extend some of these ideas to the problem of *style-based* image retrieval (SBIR), or retrieving images based on low-level features that describe identifiable characteristics of images, but are not directly related to the content in the images. Although one can describe natural images according to this set of characteristics, the natural domain for SBIR is art, since art images are routinely (if not primarily) described according to their stylistic variations. With increasing digitization of cultural artifacts and the availability of these digital representations via in the Internet [6, 7], there is an obvious cultural and scientific usefulness in describing the meaningful variations in style among works of art.

Human perception of artistic style is a combination of a number of salient visual characteristics in works of art, from the use of particular colors to the extent to which brushstrokes or pen strokes are visible. Style is also informed by a number of external factors, such as prior knowledge of the historical trends in art making. Nevertheless, the agreement of both experts and people in general in defining regularities among the styles of different artists suggests that there is a common framework for understanding style. Recent studies have focused on determining statistical features that are capable of discriminating between the styles of various artists [8, 9, 10, 11, 12, 13, 14, 15, 16]. However, these studies have failed to focus on exactly *what* features are contributing to stylistic distinctions, at least at a level beyond a very specific statistical feature. To illustrate the point: although we can tell a van Gogh from a Vermeer, we do not at present have a good sense of exactly what makes a van Gogh a van Gogh and a Vermeer a Vermeer. I will develop a model that is capable of combining statistical features for the task of style-based image retrieval, while at the same time automatically learning the salient statistical features that contribute to stylistic distinctions.

Style-based image retrieval

I define the problem of style-based image retrieval as follows: given a large collection of images T and a user-specified input image I, the system should retrieve (i.e., return) a set of images D_t that are stylistically relevant according to some category C_k. Systems exist that model global stylistic characteristics (i.e., in which the relevant category C_k is not considered) [17] and in which users, rather than categories, are modeled [18]. A further distinction of the current model is that it aims to understand important statistical categories as functions of the relationship between images according to their features, rather than model distinctions based directly on the features. This enables the model to learn for example that "color" is an important aspect of stylistic variation, rather than any *particular* color.

Ideally, the relevant stylistic categories would be learned from user interactions with the system by incorporating feedback from the user. However, due to time constraints and the complexity of such an approach, I have chosen to train the model on predefined categories (which could be used to bootstrap the system for use in recommending images before online learning begins) and evaluate the system using traditional metrics and others that are more applicable to the image recommendation context.

As a practical example, consider a user providing an art image I for which he or she wishes to find stylistically similar images. At the first stage, the system will provide a sample of possible relevant images according to the stylistic classes that it currently knows about. The user will then indicate among the returned results which are relevant to the query image I. As the user selects more relevant images, the query is refined, incorporating information it has about the most likely stylistic category C_k, according to the selections made by the user. The statistical definition of the chosen category C_k will then be updated according to the set of relevant images the user chose, enabling the system to incorporate experience in deriving its representations of stylistic categories, although, as stated, taking user feedback into consideration is not implemented in this project.

Objectives

My objectives for this project are as follows:

1. Design and implement a style-based image retrieval model that learns categorical distinctions (described below)

2. Validate this model using simulated training data, organized according to predefined stylistic categories

The similarity-based image prediction model

As stated above, a user inputs an image I into the system in order to find stylistically relevant images among the set of images $T = \{T_1, \ldots, T_N\}$ in the database. Each image T_i is described according to set of features $\phi^i = \{\phi^i_1, \ldots, \phi^i_M\}$, which may be scalars, vectors, or collections of vectors. Critically, we define a similarity function $\kappa_j(\phi^x_j, \phi^y_j)$ that describes the similarity between feature j for images X and Y, and this value will be a scalar for any feature j.

Initially, we must decide what set of images $D_0 \subseteq T$ to display in response to the query image I. Since at this point we have no information about what stylistic category the user intends to search with respect to, we can choose the set of images D_0 based on two different approaches: we can simply choose a set of "likely" images according to the current set of categories $C = \{C_1, \ldots, C_K\}$, with a fixed number of images (say, 10) chosen according to each category; or, we could select a set of images according to the marginal probability of each image T_i in the database. Both of these approaches will be described in more detail below; it should suffice at this point to recognize that whatever strategy we adopt for displaying images, we will eventually decide on some subset D_0.

Among the images D_0 that are displayed, the user may select any number of them as being "relevant" to the query image I in whatever way the user chooses. We call the entire set of relevant images chosen by a user up to and including time t $S_t \subseteq D_0 \cup D_1 \cup \ldots \cup D_t$. Once the user has indicated *some* relevant images to the query image I, we can begin to leverage this information. Consider the following function, which describes the likelihood of observing an image T_i, given a particular stylistic category C_k (with its associated parameter β_k) and query image I:

$$P(T_i|C_k, I) = f_{\beta^k}(T_i, I)^{L_i}(1 - f_{\beta^k}(T_i, I))^{1-L_i}, \tag{1}$$

where $f : \mathbb{R} \to [0, 1]$ to form a valid probability and L_i is the label of exemplar T_i. In this model, I use the logistic sigmoid function for f, and so the likelihood above is modeling the probability that the image T_i is in- or out-of-category with respect to the query image I under the model C_k, where the label L_i is determined by user feedback (which defines a logistic regression model on user feedback [19]). The function f has the following form, given β^k, T_i, and I:

$$f_{\beta^k}(T_i, I) = \frac{1}{1 + \exp\left(-\left[\beta_0^k + \sum_{j=1}^{M} \beta_j^k \kappa_j(\phi_j^i, \phi_j^I)\right]\right)}.$$

Using this formulation for an individual image, we can construct the likelihood of observing a set of relevant and irrelevant images D_t at an arbitrary iteration t of the search (specified by the user as feedback):

$$P(D_t|C_k, I) = \prod_{T_i \in S_t} P(T_i|C_k, I).$$

If we assume a uniform prior on all categories C_k (although this assumption is not necessary and is motivated at this point strictly by convenience), then we can infer

the posterior probability of each class C_k via Bayes' Rule:

$$P(C_k|D_t, I) = \frac{P(D_t|C_k, I)P(C_k)}{\sum_{C_l \in C} P(D_t|C_l, I)P(C_l)},$$

where $P(I)$ has been omitted as it is taken to be constant. Without a more complex form for the prior $P(C_k)$, the maximum *a posteriori* choice for C_k will be the same as the one determined by maximum likelihood. Nevertheless, it is instructive to consider the full posterior over C_k, since it highlights areas in which the complexity of the model could be increased, for example by estimating a "prior" over C_k by simply using the fraction of times C_k was predicted to be the relevant stylistic category. Another possible extension would be to consider the conditional $P(C_k|I)$, which was omitted from the current model, since we assume no dependence of C_k on I, but which could be used to estimate a category according to some function of the images currently in the database, before any relevant images S_0 are selected by the user.

Once a search has been completed, we can update the parameters β^k associated with category C_k by considering the images chosen by the user as relevant (S_t) and the set $D = D_0 \cup D_1 \cup \ldots \cup D_t$ of possible relevant images shown to the user. For each image $T_i \in S_t$ we define an associated target variable L_i, which we set equal to 1, indicating that the relevant images in the set S_t, as chosen by the user, are "in-category" images with respect to I and the class C_k. Furthermore, we say that all images $T_i \in D_{out} = D \setminus S_t$ are the "out-of-category" images, where $\setminus$ denotes the set minus operation, and concomitantly these images have associated target variables $L_i = 0$. We can now reconsider Equation 1 in this conetxt:

$$P(T_i|C_k, I) = \sigma\left(\beta_0^k + \sum_{j=1}^{M} \beta_j^k \kappa_j(\phi_j^i, \phi_j^I)\right)^{L_i} \sigma\left(\beta_0^k + \sum_{j=1}^{M} \beta_j^k \kappa_j(\phi_j^i, \phi_j^I)\right)^{1-L_i}, \quad (2)$$

where σ is the logistic sigmoid function. Given the relevant images chosen by the user (and the disregarded images D_{out}), we can now update the parameters β^k according to the error function defined by the negative log-likelihood of the relevant/disregarded images, given our set of "labeled" images $T_i \in S_t \cup D_{out}$ and associated target classes L_i [19]:

$$E(\beta^k) = -\ln P(L|\beta^k) = -\sum_{T_i \in S_t \cup D_{out}} \Big[L_i \ln y_i + (1 - L_i) \ln(1 - y_i)\Big],$$

where $y_i = \sigma\left(\beta_0^k + \sum_{j=1}^{M} \beta_j^k \kappa_j(\phi_j^i, \phi_j^I)\right)$. Taking the gradient of the error function

with respect to β^k, we obtain an update rule for the weight vector:

$$\frac{\partial E}{\partial \beta_j^k} = \sum_{T_i \in S_t \cup D_{out}} (y_i - L_i)\kappa_j(\phi_j^i, \phi_j^I).$$

Thus we can adjust the weights β^k iteratively using gradient descent, with each new input I and given the chosen relevant images S_t and those D_{out} disregarded by the user.

Choosing the set of display images

As mentioned above, there are several alternatives for choosing the set of images D_t to display at any iteration t of a particular query. Of particular interest is the method with which one might choose D_0, the initial set of images displayed to the user in response to a query image I. We may choose for example n of the most probable images for each of the existing categories C_k, or we may choose the overall most probable images by marginalizing over the categories in the following way:

$$P(T_i|I) = \sum_{C_k \in C} P(T_i|I, C_k)P(C_k).$$

In order to compute the posterior over C_k, this quantity must be computed, so the increased computational cost of this approach is negligible.

Image features

I spent a great deal of time collecting images from Bing Image Search [2] in order to provide some base stylistic classes that could be used to evaluate the efficacy of the model (e.g., I searched for images matching the queries "Abstract art," "Cubism," "Impressionism," and "Renaissance art," among others). Critically, the images returned for each query were stylistically quite diverse, although they did in some sense fit their categorical description. This highlights the fact that stylistic periods are in many ways more historical artifacts than concrete descriptions. For example, the art of Jackson Pollock is quite different from that of Mark Rothko, although both are Abstract Expressionists.

For this reason, I concentrated on identifying stylistic categories that were more visually consistent. These categories may be individual subsets of an artist's work, separated according to style. For example, I took 79 paintings by Picasso and grouped them according to their common stylistic category. I also created a dataset from

drawings by three artists (see below) that have relatively good in-category stylistic similarity, and further expanded this to seven different stylistically consistent sets of images by six artists (see also below). Datasets such as this provide a "simpler" ground truth against which the model can be tested.

In my experiments, I considered several image features (which will be indicated below by the tag specified here):

1. `color1`: Histogram intersection between RGB color histograms [20, 21]
2. `color2`: Absolute difference in entropy of color distributions [19]
3. `fourier1`: Hausdorff distance between slopes of log rotational average of patch-wise amplitude spectra between images [22]
4. `fourier2`: Hausdorff distance between KL-divergence of radial averages of patch-wise amplitude spectra [22]
5. `gabor1`: Histogram intersection between cluster proportions per image based on k-means clustering of patches according to correlation distance between basic statistics derived from a Gabor function decomposition of the patches (32 patches from each image, 4 clusters) [22]
6. `gabor2`: Histogram intersection between cluster proportions per image based on k-means clustering of patches according to correlation distance between per-filter energies from a Gabor function decomposition of the patches (32 patches from each image, 4 clusters) [22]
7. `line1`: Histogram intersection of line orientations for lines detected using the Hough transform on an edge image [23]
8. `line2`: Euclidean distance of basic statistics on line lengths for lines detected using the Hough transform on an edge image [23]
9. `line3`: Histogram intersection of patch-wise line density (i.e., number of edge pixels in a patch of fixed size)
10. `raw1`: Correlation between raw images (i.e., a downsampled version of each image) [20]
11. `gist1`: χ^2 distance between GIST features [20]

12. `slope1`: Euclidean distance between slope of the log of the rotational average of the amplitude spectrum for each image [15, 22]

In each case, if a distance was computed, then it was converted to a similarity via $\exp(-d(x,y)/0.5)$, where $d(x,y)$ is the distance between two sets of features x and y.

Experiments

The ultimate goal of this project is to develop a system that is capable of taking user feedback and learning a set of stylistic classes that describe the relationships between images determined by user-provided feedback. However, before this goal can be accomplished, it is necessary to validate the model and to demonstrate that 1) it can learn stylistic distinctions between images described by salient features and 2) that such distinctions actually appear in the data, at least using the features we have at present.

It is critical also to emphasize that the learned models β_k are not feature-based, but rather feature similarity-based. For this reason, a requirement of model validation is that it is capable of learning the commonalities among similar images according to their features. Since the weights we learn in the logistic regression model directly refer to the individual feature similarities (although the method for calculating these similarities is effectively hidden from the model), we can interpret large (relative) weights as placing strong emphasis on the corresponding feature. This suggests a means to initially validate the model: create images that vary according to a particular stylistic feature (or set of features) and evaluate the degree to which the model is capable of "recognizing" which features are important and which are not.

Methodology

In all experiments below, I trained models in the following fashion. first, I held out 25% of the overall data for testing, and kept the remaining 75% for training. Given a particular stylistic class that I was attempting to model (e.g., "Picasso"), I trained the model using each true image from the corresponding class as a target image and the similarities to all other images as regressors in the model. That is, I treated the entire training set as pseudo-feedback with in- or out-of-category labels given by the true labeling. For example, if the class I was attempting to model was "Picasso," then all Picasso images would be considered in-category and all other

images out-of-category. Similarities $\kappa_j(\phi_j^i, \phi_j^I)$ were computed separately, as per the description above, for each target image T_i. These regressors were stacked into a single matrix, as were the corresponding labels. In this way, I simultaneously used all possible training data (for a particular class) to learn a model to distinguish one class of images from the rest. This is not necessarily how training would proceed in the online setting, but it is instructive in the context of model validation.

Evaluation criteria

In order to evaluate the performance of a given model or set of models, I concentrated on several criteria that describe various aspects of the performance of the model(s):

1. Training and testing error for classifying images as in- or out-of-category
2. Visual inspection of the learned weights
3. Best model prediction using log-likelihood (i.e., the method above that would be used to select the optimal C_k, given the observed data)
4. False positive vs. true positive prediction rates on testing datapoints
5. The fraction of possible in-category images predicted in the top 10 most likely images, according to each model, which simulates recommending images to users (the intuition being that we want in-category images to be highly likely and thus predicted before out-of-category images)

Except for the initial two experiments (which used only the first two criteria), all of these criteria were used to evaluate the learned models in the experiments described below.

Experiment 1 - Distinguishing based on color feature

To validate the model and demonstrate that it is capable of learning which features are important in a highly controlled setting, I created two sets of random images, each of which contained variation along a particular feature.

The first set of random images I created were random RGB images that varied according to their color histograms (i.e., "color" was the perceptually salient dimension of stylistic variation). I created three "stylistically distinct" subsets of random images by first creating, for each image, three 256x256 pixel uniform random images using Matlab's `rand` function. Each of these represented the red, green, and

blue channel of the random image, respectively. In order to emphasize one of the three color components, I multiplied the noise image in either the red, green, or blue channel by a factor of 3. Effectively, this makes images whose (for example) red component at a particular pixel is on average three times larger than the corresponding green or blue component. This will cause images to appear more red, green, or blue, depending on the category. Figure 1 shows an example of a random image from each of the three groups. I created 75 random images from each category. Critically, since the images are noise images, they should not meaningfully vary according to any other stylistic feature.

I trained a model to distinguish the images in this experiment using the procedure described above. In this particular experiment, we need to learn only one β^k, since we are only attempting to model one distinction (i.e., color). The learned model is shown in Figure 2; clearly, the model has learned that "color" is the most important distinguishing feature. Furthermore, this model achieved 0% training and testing error (i.e., it was able to perfectly distinguish the images based primarily on their color distributions). Note that in this experiment I did not use the color entropy feature, since I added this feature later.

Experiment 2 - Distinguishing based on slope feature

I performed an experiment similar to the one above using images that varied according to a different feature, namely the slope of the log rotational average of the amplitude spectrum. Shown below are two images, one noise image with a flat amplitude spectrum, and another whose frequency spectrum has been modulated so that it contains more low-frequency than high-frequency information (and the frequency response falls off as $1/f^{1.5}$, where f denotes frequency). This will affect the slope of the line fit to the log of the rotational average of the amplitude spectrum in a consistent way, so this feature should stand out as being important. Specifically, the unmodulated image should have slope of roughly 0, while the modulated images should have a slope of roughly -1.5. Figure 3 shows an example of a uniform random image and a random image whose frequency spectrum has been modulated.

The learned weights from training a model as described above are shown in Figure 4. Clearly, the `slope1` feature, which refers to the slope of the log of the rotational average of the amplitude spectrum, is the most dominant feature, but other features, particularly those that are sensitive to frequency structure in the image (e.g., Fourier- and Gabor-based features) also show discriminative power. This is not surprising, since the uniformity of the frequency structure of the noise in both types of images is present (at least at random) in all parts of the images; thus, patch-based methods

should recover differences similar to those that the global slope feature is able to distinguish. As before, I achieved training and testing error of 0%.

Experiment 3 - Distinguishing drawings by three artists

In order to create a more realistic setting in which to test the model, I took drawings from three artists that were fairly stylistically consistent. Figure 5 shows one drawing each by Pieter Bruegel the Elder, Raymond Pettibon, and Rembrandt van Rijn. These drawings are from wildly different periods in art (Bruegel lived in the 16th century, Rembrandt in the 17th, and Pettibon in the 20th), and are distinguished in subject and medium, to some extent. This dataset included 46 drawings by Bruegel, 29 by Pettibon, and 20 by Rembrandt.

I trained models as before, holding out 25% of the images from each class and keeping the remaining 75% as training data. I trained a model to distinguish each artist's works from all the rest. The learned models are shown in Figure 6.

Clearly, the most important features appear to be `gist1` (for all classes) and variously the `color1`, `color2`, `fourier2`, and `line3` features. I achieved a respectable training error (below 15% for all models, as shown in Figure 7) and overall testing error rates of 3%, 5.6%, and 16.4% for each of the three classes, respectively, using the corresponding model. Furthermore, the corresponding models produced good tradeoff between false positive and true positive rates, as shown in Figure 8. Thus it is clear that the learned models were good at separating the three classes of drawings.

I evaluated the performance in other ways more germane to the task of recommending stylistically similar images as well. As described above, I also considered the ability of the model to predict in-category images among the top 10 most probable images, given a testing image as a query image.

The performance of the model with respect to this metric is shown in Figure 9. The testing error indicated that the model corresponding to the query image results in good overall performance, and the model scores shown in Figure 9 indicate that in-category images were highly consistently placed in the top 10 most probable images for all three models, though less strongly for the Rembrandt model. In a real setting, however, we would not know *which model* was the optimal one for an input query, so it is also instructive to consider how often the model corresponding to the class of the testing (query) image was in fact the correct model. For this dataset, the model achieved 0% error in choosing the correct model (based on log-likelihood of the training images treated as observations), which suggests that the model has captured relevant structure in these images.

Experiment 4 - Distinguishing works of art in seven styles

In order to create a more complex comparison (in particular, one that includes more stylistic classes), I augmented the drawings dataset with works of art by several other artists to create a dataset with the following composition:

- 46 drawings by Bruegel
- 29 drawings by Pettibon
- 20 drawings by Rembrandt
- 14 paintings/drawings by Charlotte Capsers, which include multiple copies of the same setting for comparison
- 12 paintings by Odilon Redon
- 13 portrait paintings by Rembrandt
- 19 paintings by Vincent van Gogh

In this case, the imbalance in data is even more pronounced, and it will be shown that this creates overfitting problems in training.

As before, I trained a model to distinguish each of the seven groups above from the rest. The learned models are shown in Figure 10. Interesting, there seem to be a variety of different features that are important across the classes, which suggests that this set of features is fairly diverse, although it is likely that some redundancy exists. Even so, the `gist1` feature seems to be dominant across models, which may be account for by the fact that in many cases, the groups are fairly stylistically distinct. These models resulted in (fairly) good training and testing error, as shown in Figure 11. However, looking at the trade-off between false- and true-positive rates (Figure 12), it is clear that, while testing error is fairly low overall, this is largely due to the fact that most negative exemplars are correctly identified, but many (if not most) positive exemplars are *not* correctly identified.

In order to mitigate the problems of what appears to be underfitting with respect to the in-category datapoints most likely caused by the major imbalance between the amount of positive and negative data available during training, I implemented a version of the model that allows for variable weighting of misclassification of the datapoints, so that misclassifying a positive exemplar carries a much larger penalty. Specifically, misclassification was weighted in the following manner:

$$w_i = \begin{cases} \frac{m^+}{m^-}, & \text{if } L_i = 1 \\ \frac{m^-}{m^+}, & \text{if } L_i = 0 \end{cases},$$

where m^+ is the number of in-category exemplars and m^- is the number of out-of-category exemplars. This transforms the likelihood function, which alters the gradient of the log-likelihood function in the following way:

$$\frac{\partial E}{\partial \beta_j^k} = \sum_{T_i \in D_t} w_i (y_i - L_i) \kappa_j(\phi_j^i, \phi_j^I).$$

From this it is clear that, if the number of positive exemplars is smaller than the number of negative exemplars, then an equivalent misclassification will result in a higher penalty for an in-category exemplar than for an out-of-category exemplar.

Using this strategy, I trained seven new models to distinguish the drawings above, as shown in Figure 13. The training and testing errors are generally higher for these models (see Figure 14), but the false versus true positive tradeoff on the testing set is significantly improved. Figure 15 shows the tradeoff for each model (indicated by a colored symbol) with respect to all testing images from the category indicated in the title of the figure. For example, for all Bruegel drawings in the test set, I look at the tradeoff between false and true positive rates when predicting the category for the training images (using the testing image as a query image). In most cases, the tradeoff is much better than the one shown in the equivalent figure for the unweighted model, and the true positive rate is typically much higher, though this often comes with a somewhat higher false positive rate as well.

It is interesting to consider the predictive capabilities of these models, as mentioned above, in the image retrieval setting. For the testing images, the weighted models were able to accurately predict the best model (based on log-likelihood) with an error rate of only 13.9%. Furthermore, the models were also able to achieve a high rate of success at placing in-category images within the top 10 most probable images with respect of the query image and corresponding model, as shown in Figure 16. Nevertheless, performance was poor for some classes, such as "Redon." Interesting, despite having worse performance with respect to category a labeling, the unweighted models performed overall *slightly* better at predicting in-category images among the top 10 results. The scores for each testing image are shown in Figure 17 for comparison.

Experiment 5 - Distinguishing the styles of Picasso

I tried a similar experiment using images from Picasso's various stylistic periods. The dataset I used consisted of

- 8 paintings from the Blue period

- 8 paintings from the Rose period
- 25 paintings from the early Cubist period
- 9 paintings from the Analytical Cubist period
- 11 paintings from the Synthetic Cubist period
- 8 paintings from the Interwar period
- 4 paintings from the post-World War II period
- 6 paintings from the Late period

Unfortunately, the results for this dataset were not as good as those for the previous experiments. Both training and testing error for weighted models (necessary due to the extreme data imbalance) were quite high, and the tradeoff between false- and true-positive rates for the testing data was quite unfavorable in all cases, which suggests that the models learned were highly overfit to the training data and possess little ability to generalize to new exemplars.

Image recommendation and style-based image retrieval

The core goal of my project was to provide a platform that could be used for two tasks: first, recommending images that are stylistically similar to a query image; and second, automatically learning salient stylistic classes from user input. The latter goal has been achieved in this project using simulations rather than actual user input, and the former goal has been touched on in the results. Specifically, I have given, for the tasks described, some measures of the quality of the recommended images and the ability of the learned models to "detect" the correct stylistic class. However, up to this point I have not given any specific examples of the success (or failure) of the model to recommend images that appear to be stylistically similar to the query.

Here I present several examples of query images and the 5 most probable images according to the learned models, using testing images from the "seven styles" experiments shown above. Figure 18 shows the returned images according to the Bruegel model, using a Bruegel landscape drawing as input. Figure 19 shows returned images using a scene with people as a query image. In both cases, all five drawings are by Bruegel, and most are not only stylistically but also semantically related to the query image (four of five drawings in each example are semantically related to the query image). This is not due to semantic information as such being capture by the model (since none is), but rather because of shared statistical features between images of

particular types. However, the model seems able to abstract away from these, too, since it correctly predicts Bruegel drawings, regardless of content, as the most likely images.

Figure 20 also shows an interesting recommendation result, in which the most likely images given the Caspers model returned four images that were copies of the query image. This dataset had several base images that were copied by the artist, but retain objects and of course stylistic similarities with the original. Another test using a Rembrandt drawing and the Rembrandt drawings model yielded predicted images, four out of five of which were also from the Rembrandt drawings dataset (see Figure **??**). The one non-Rembrandt image contained a scene similar to the query image. A final example (shown in Figure 22) shows the most probable images using the van Gogh model, using a van Gogh painting as a query image. This example shows a less successful result (only three of the top five images were by van Gogh), but in some ways the non-van Gogh images returned do possess stylistic similarities with the query image.

Conclusions

I have implemented a model that is capable of learning stylistic classes from inputs and I have demonstrated that this model is capable of learning predefined stylistic classes in a simulated setting. Clearly, the next step is to extend this to actual perceptual data and a much more diverse set of images. I have also shown that the models learned capture important features of the input images and are able to recommend similar images with fairly good consistency for several classes. In some cases, however, the "correct" in-category images were not among the most probable images. In these cases, though, it seemed that a paucity of data was likely to blame. Categories in which there was a large number of exemplars tended to have better rates of prediction for in-category images, and the relationship between these quantities was somewhat strong (on 7-category dataset with weighted model, Pearson's r between size of each group of images and prediction scores on test images was 0.58 at $p = 2.3 \times 10^{-4}$). The results I obtained so far indicate that there is promise in this model, but, as stated, it is important to apply it to the arguably much more difficult task of modeling human perception of artistic style. In future, I also plan to explore the discriminative power of this model for categorical distinctions other than between artists or periods in an artist's work, for example whether the model can distinguish between landscapes and non-landscape works of art, regardless of medium or the time period in which the works were created.

References

[1] "Google image search," in *http://www.google.com/images*.

[2] "Bing image search," in *http://www.bing.com/images*.

[3] N. Vasconcelos, "From pixels to semantic spaces: Advances in content-based image retrieval," *IEEE Computer*, 2007.

[4] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Transactions on Multimedia Computing, Communication, and Applications*, 2006.

[5] Y. Chen, J. Li, and J. Z. Wang, *Machine Learning and statistical modeling approaches to image retrieval*. Kluwer Academic Publishers, 2004.

[6] "Google art project," in *http://www.googleartproject.com/*.

[7] S. Hockey, "The history of humanities computing," in *A Companion to Digital Humanities*, Blackwell Publishing, 2004.

[8] R. P. Taylor, A. P. Micolich, and D. Jonas, "Fractal analysis of Pollock's drip paintings," *Nature*, vol. 399, June.

[9] J. M. Hughes, D. J. Graham, and D. N. Rockmore, "Quantification of artistic style through sparse coding analysis in the drawings of Pieter Bruegel the Elder," *Proceedings of the National Academy of Sciences*, vol. 107, pp. 1279–1283, 2010.

[10] R. Taylor, R. Guzman, T. Martin, G. Hall, A. Micolich, D. Jonas, B. Scannell, M. Fairbanks, and C. Marlow, "Authenticating pollock paintings using fractal geometry," *Pattern Recognition Letters*, vol. 28, pp. 695–702, 2007.

[11] L. van der Maaten and E. Postma, "Identifying the real van Gogh with brushstroke textons." White paper, Tilburg University, February 2009.

[12] I. Berezhnoy, E. Postma, and H. van den Herik, "Authentic: computerized brushstroke analysis," pp. 1586–1588, 2005.

[13] J. C. R. Johnson, E. Hendriks, I. Berezhnoy, E. Brevdo, S. Hughes, I. Daubechies, J. Li, E. Postma, and J. Wang, "Image processing for artist identification," *IEEE Signal Processing Magazine*, vol. 37, July 2008.

[14] S. Bucklow, "A stylometric analysis of craquelure," *Computers and the Humanities*, vol. 31, pp. 503–521, 1998.

[15] D. Graham, J. Friedenberg, D. Rockmore, and D. Field, "Mapping the similarity space of paintings: image statistics and visual perception," *Visual Cognition*, 2009.

[16] D. Graham, J. Friedenberg, D. Rockmore, and D. Field, "Efficient visual system processing of spatial and luminance statistics in representational and non-representational art," vol. 7240, pp. 1N1–1N10, 2009.

[17] J. Z. Wang, J. Li, and G. Wiederhold, "Simplicity: Semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 9, 2001.

[18] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos, "The bayesian image retrieval system, pichunter: Theory, implementation, and psychophysical experiments," *IEEE Transactions on Image Processing*, 2000.

[19] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 1st ed., October 2007.

[20] C. W. et al., "Categorizing art: comparing humans and computers," *Computers & Graphics*, vol. 33, 2009.

[21] A. Barla, F. Odone, and A. Verri, "Histogram intersection kernel for image classification," 2003.

[22] T. Pouli, D. Cunningham, and E. Reinhard, "A survey of image statistics relevant to computer graphics," *Computer Graphics Forum*.

[23] J. Illingworth and J. Kittler, "A survey of the hough transform," *Computer vision, graphics, and image processing*, 1988.

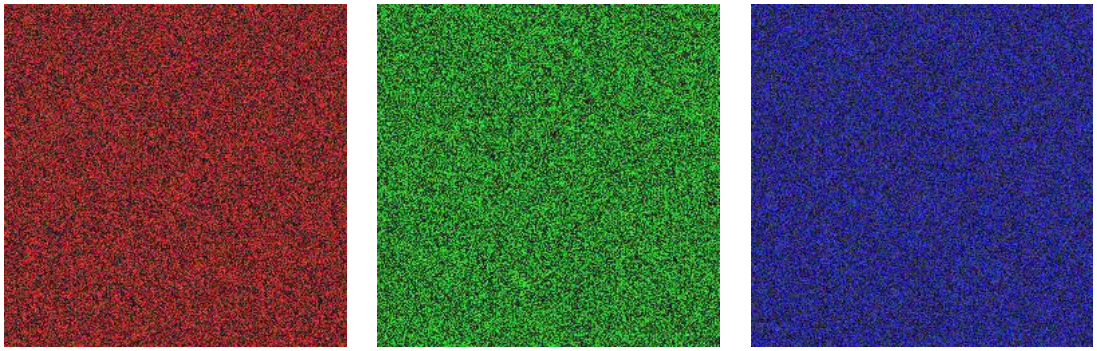

Figure 1: Random images from the red, green, and blue classes. Each was created using uniform random noise to represent each color channel, except that the noise in the channel corresponding to the desired class was on average three times larger in magnitude than for the other two channels.

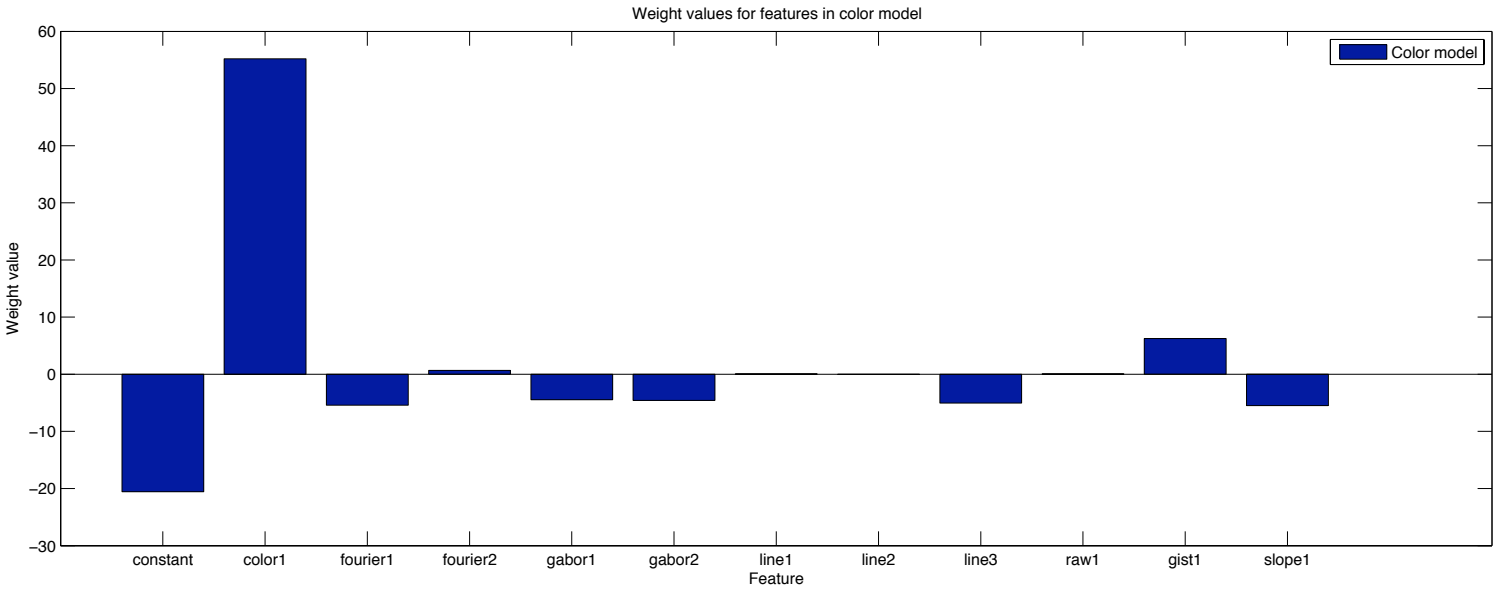

Figure 2: Learned model in color experiment. The color feature ('color1') is clearly the most important feature, indicating that the model has identified the correct distinguishing feature.

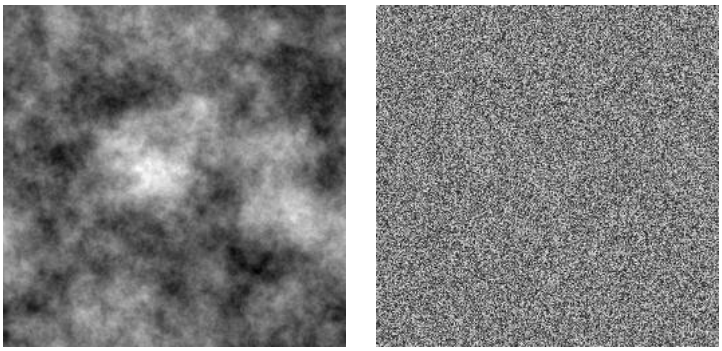

Figure 3: Random noise images. Although both images were created using 256x256 uniform random noise, the amplitude spectrum of the left-hand image has been modulated so that amplitude falls off roughly as $1/f^{1.5}$ across frequency f.

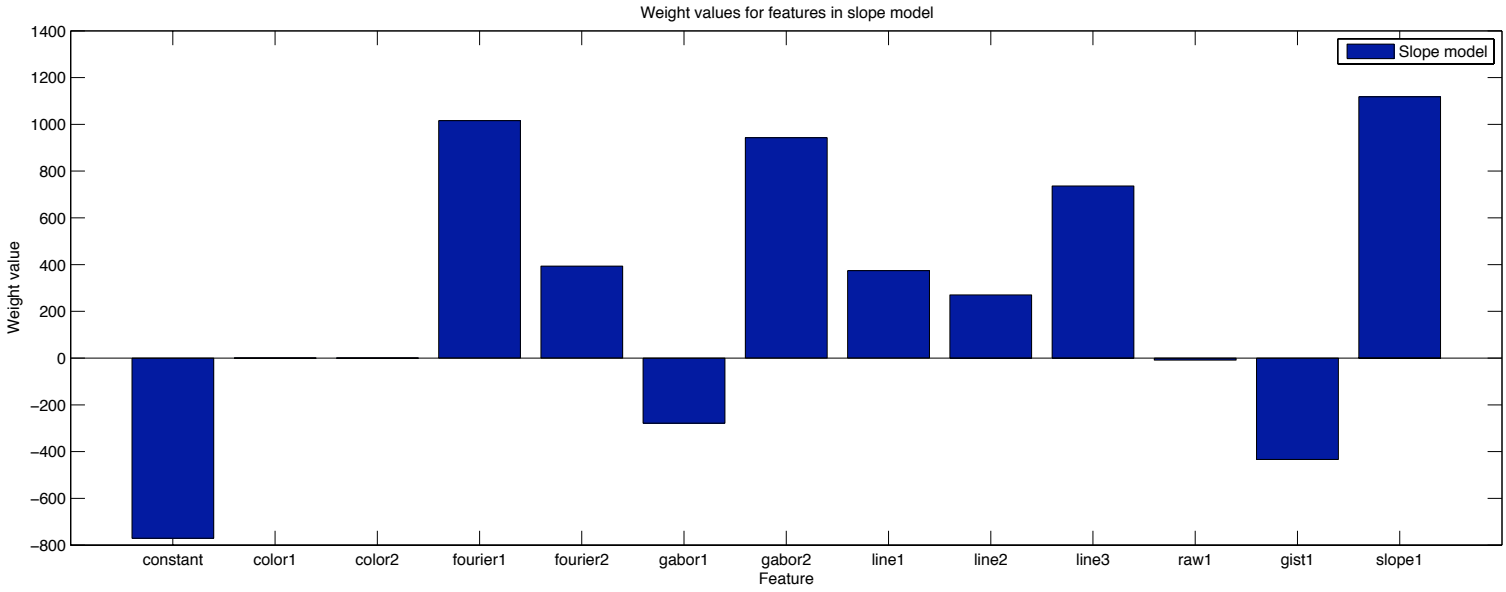

Figure 4: Learned model in slope experiment. The slope feature ('slope1') is clearly the most important feature, although several other features contribute to the distinction between the frequency-modulated noise and the unmodulated noise.

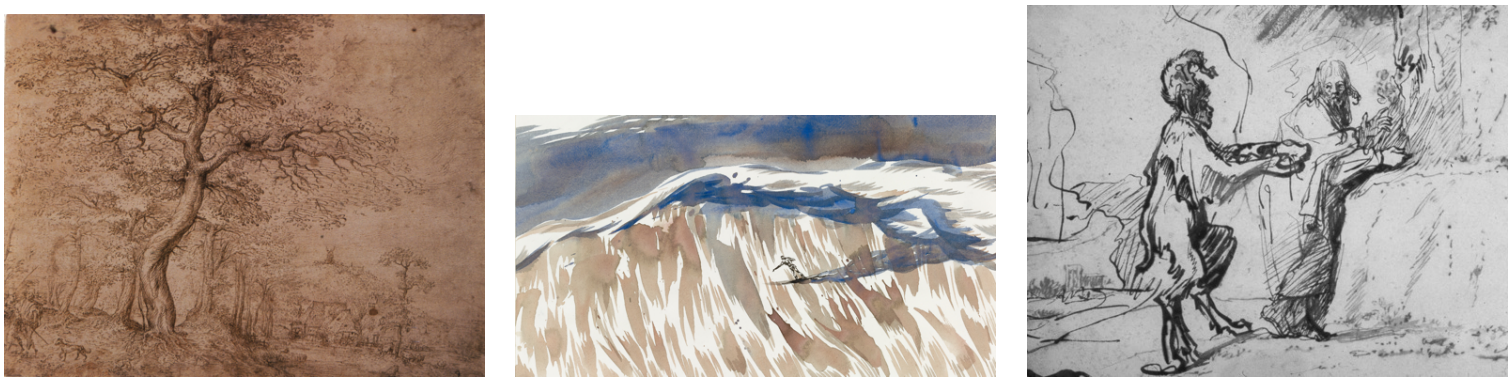

Figure 5: Drawings from each of the three artists in the drawings experiment. From left to right: Bruegel, Pettibon, and Rembrandt.

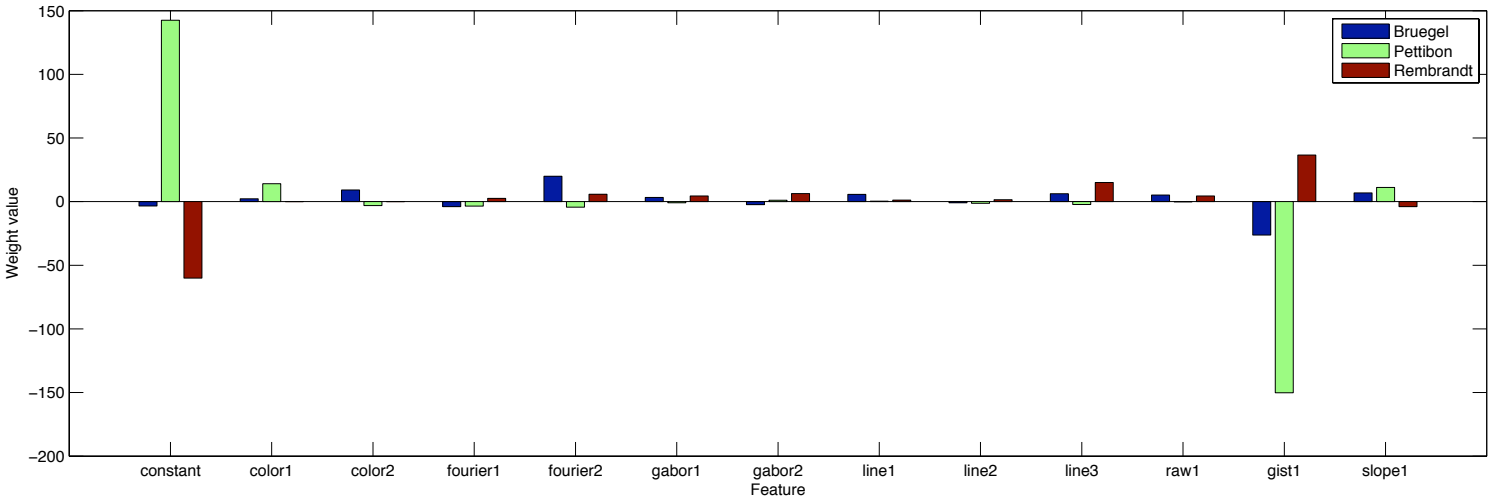

Figure 6: Learned models in drawings experiment. Several feature appear prominently, though the 'gist1' feature appears dominant in almost every model.

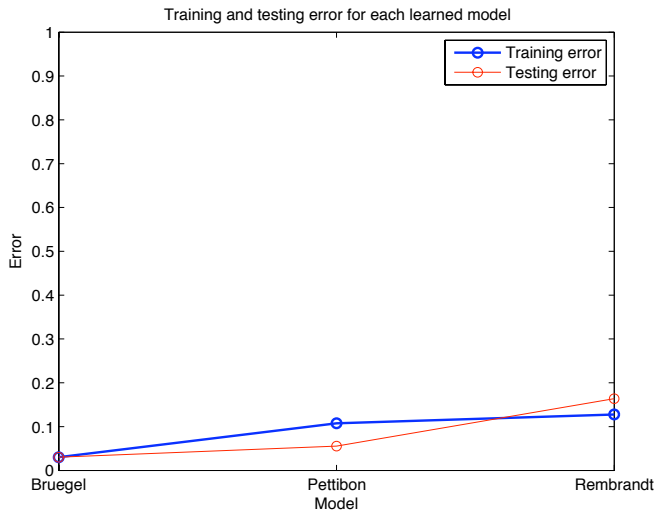

Figure 7: Training and testing error for each model (testing error represents average over all testing images) in the drawings experiment.

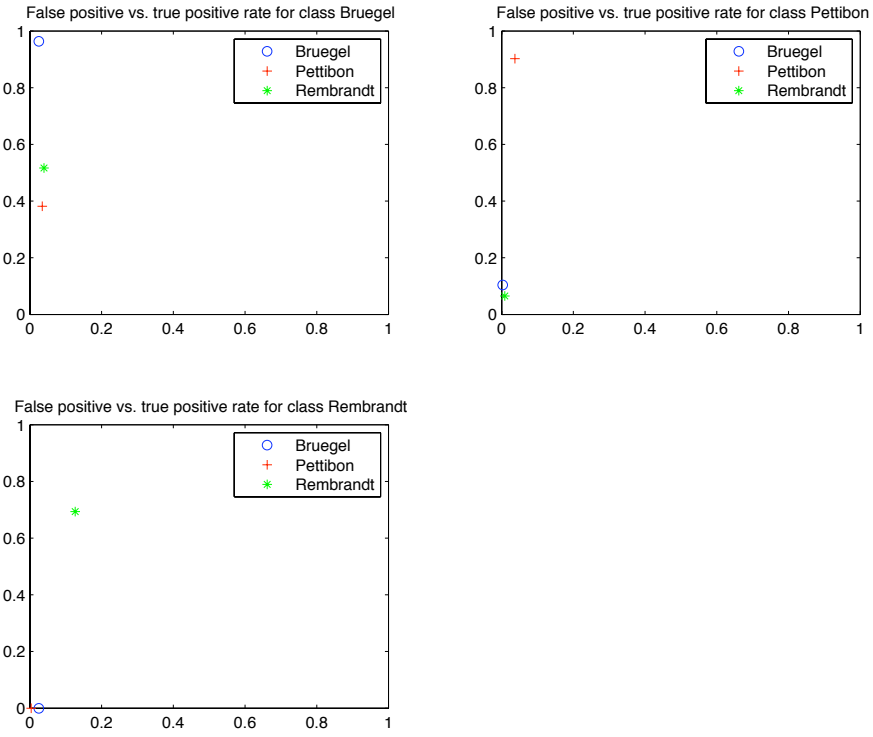

Figure 8: False- and true-positive rates for each category of images indicated at the top of each plot; the plotted symbols indicate the false and true positive rates for the corresponding model. In each case, the model corresponding to the image category showed the best tradeoff (e.g., the Bruegel model was optimal for Bruegel drawings, as shown by the open blue circle in the first plot).

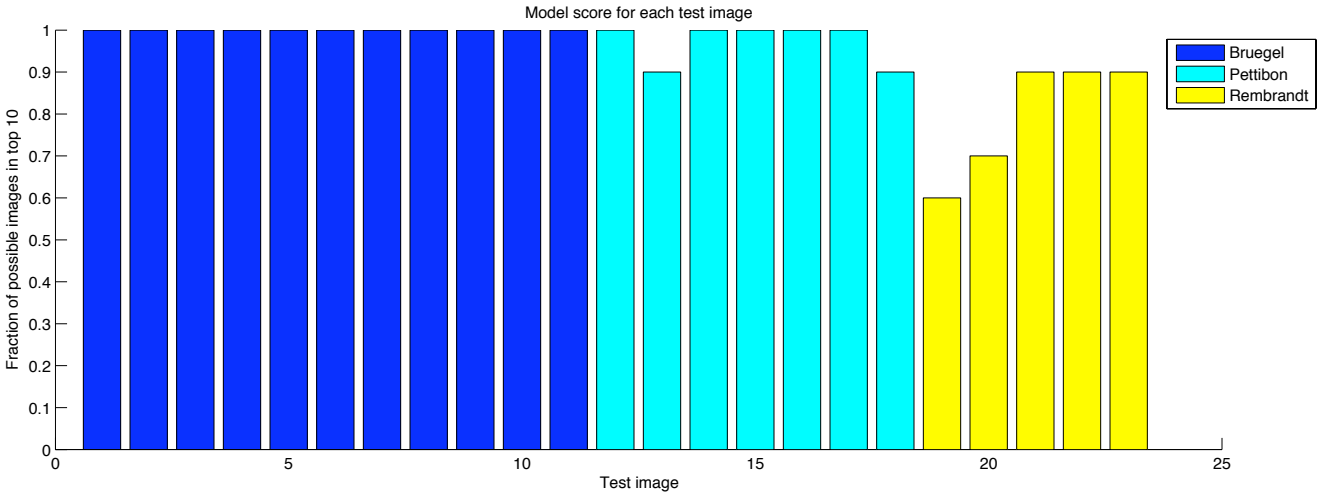

Figure 9: Model scores for testing images in drawings experiment. The vertical axis represents the fraction of (possible) training images predicted among the top 10 most probable images, given the testing (query) image. Clearly, each model was very good at associating the testing exemplars with the correct images.

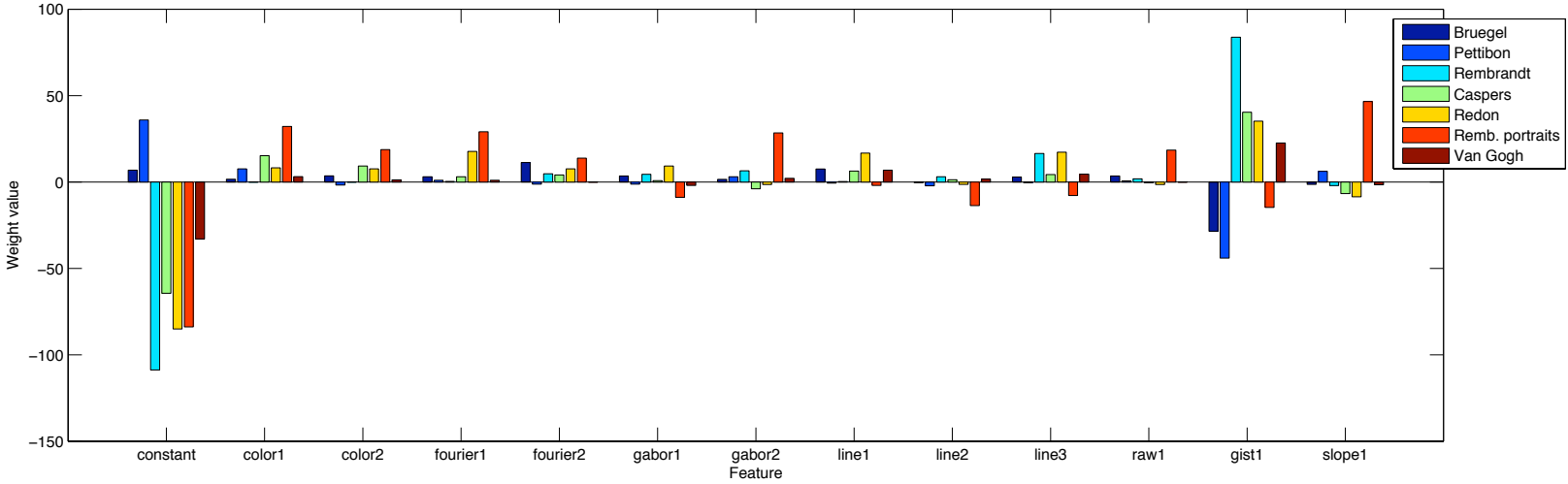

Figure 10: Learned models in seven styles experiment. Several features appear prominently, though once again the 'gist1' feature appears dominant in most models.

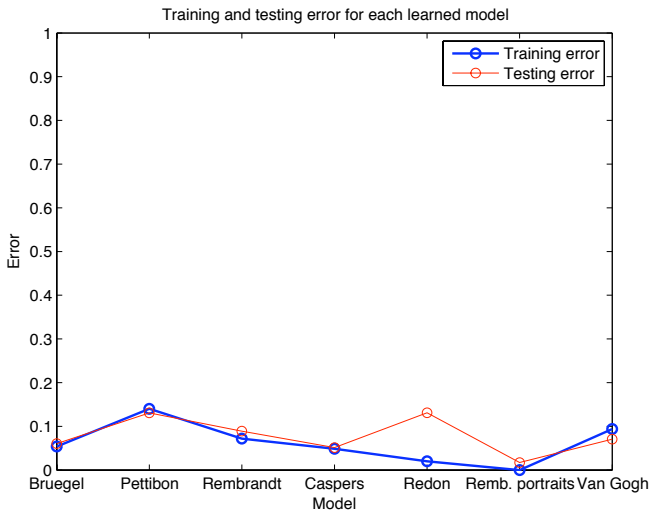

Figure 11: Training and testing error for each model (testing error represents average over all testing images) in the seven styles experiment.

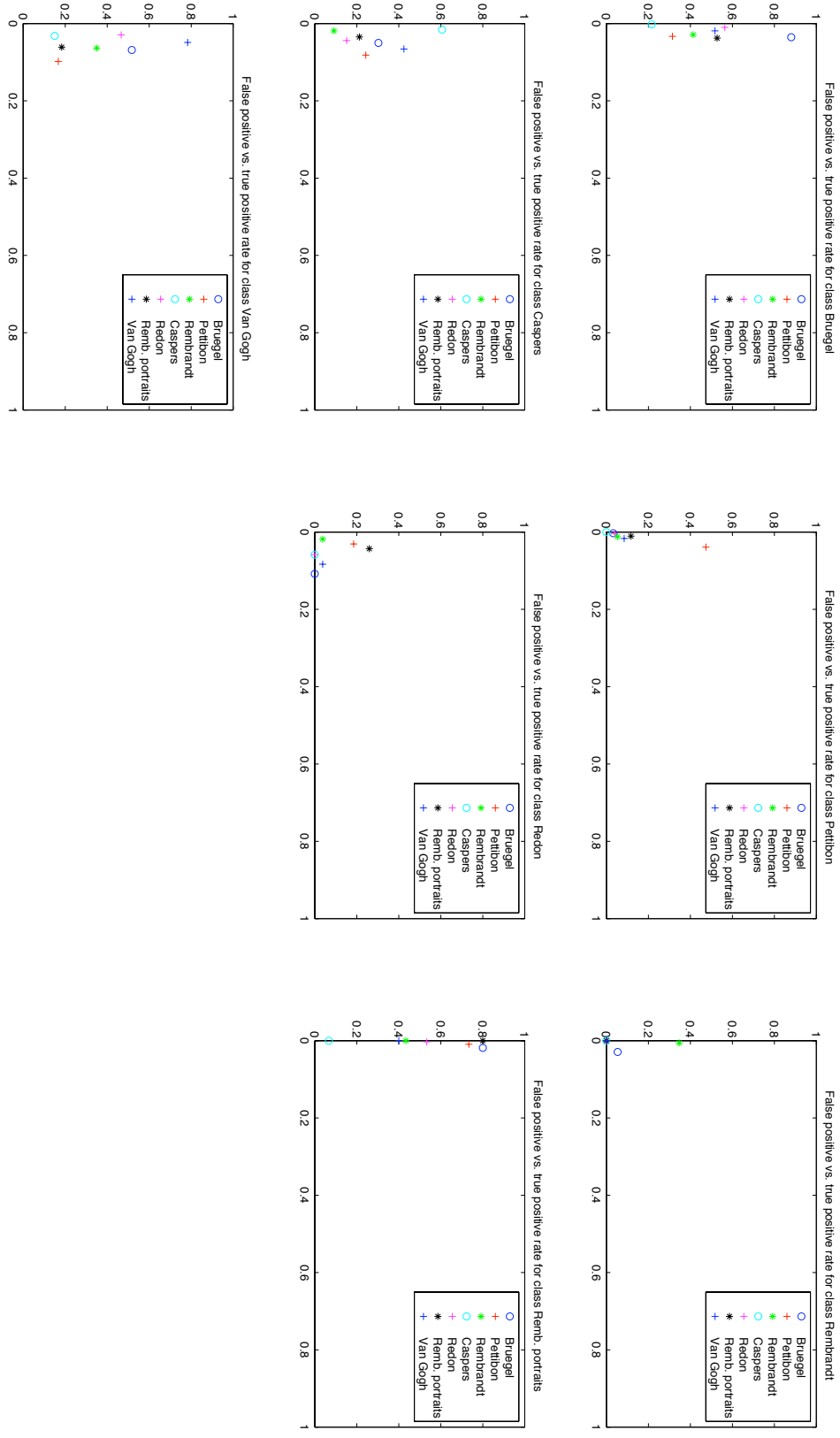

Figure 12: False- and true-positive rates for each category of images indicated at the top of each plot; the plotted symbols indicate the false and true positive rates for the corresponding model. Using an unweighted logistic regression model, the generalization performance of all models (except the first) is relatively poor.

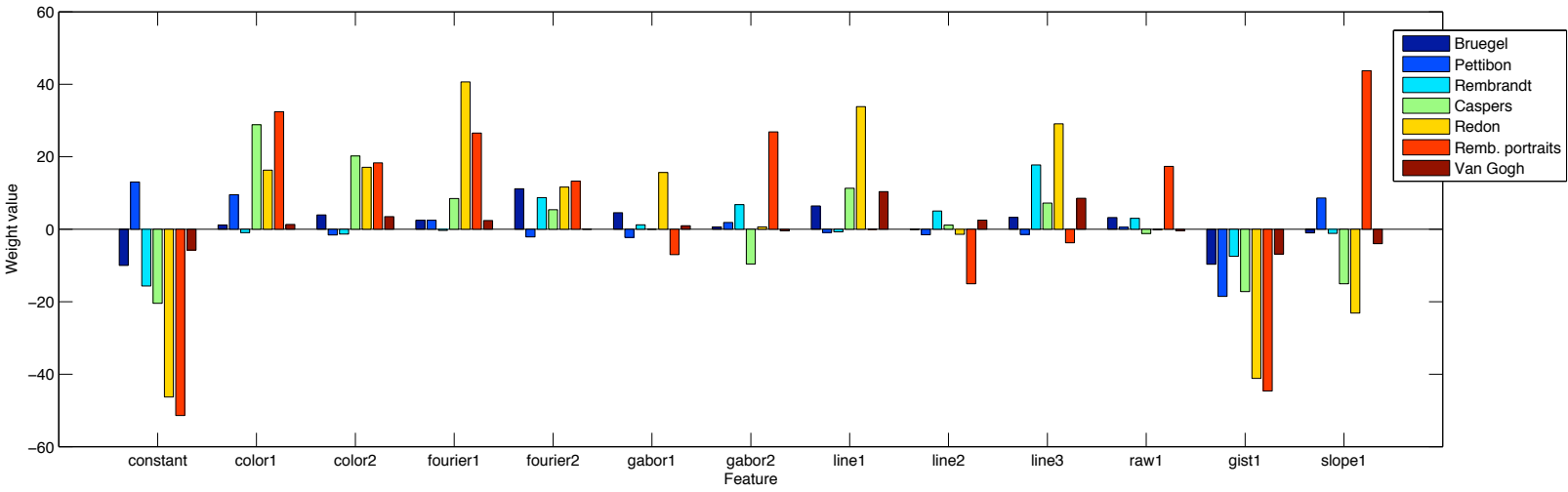

Figure 13: Learned models in seven styles experiment using weighted logistic regression.

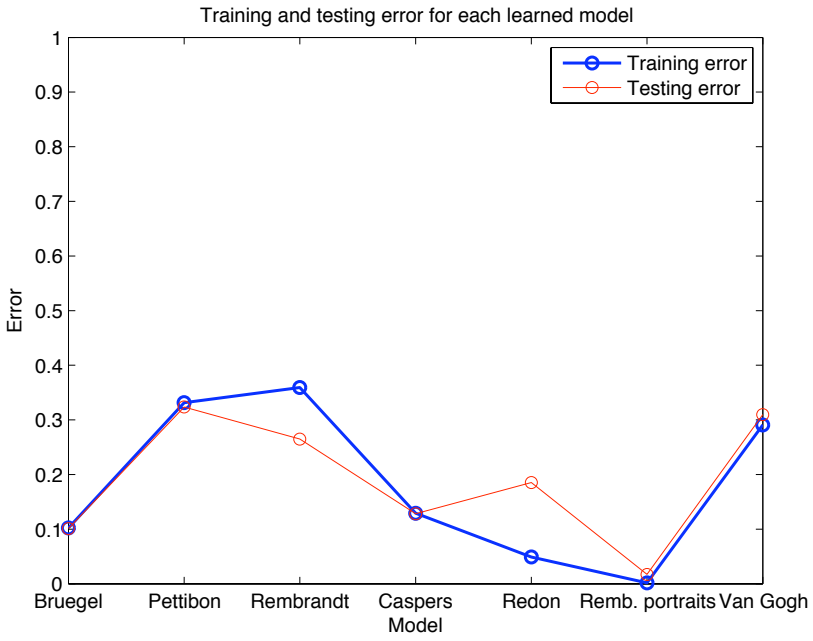

Figure 14: Training and testing error for each model (testing error represents average over all testing images) in the seven styles experiment using weighted logistic regression.

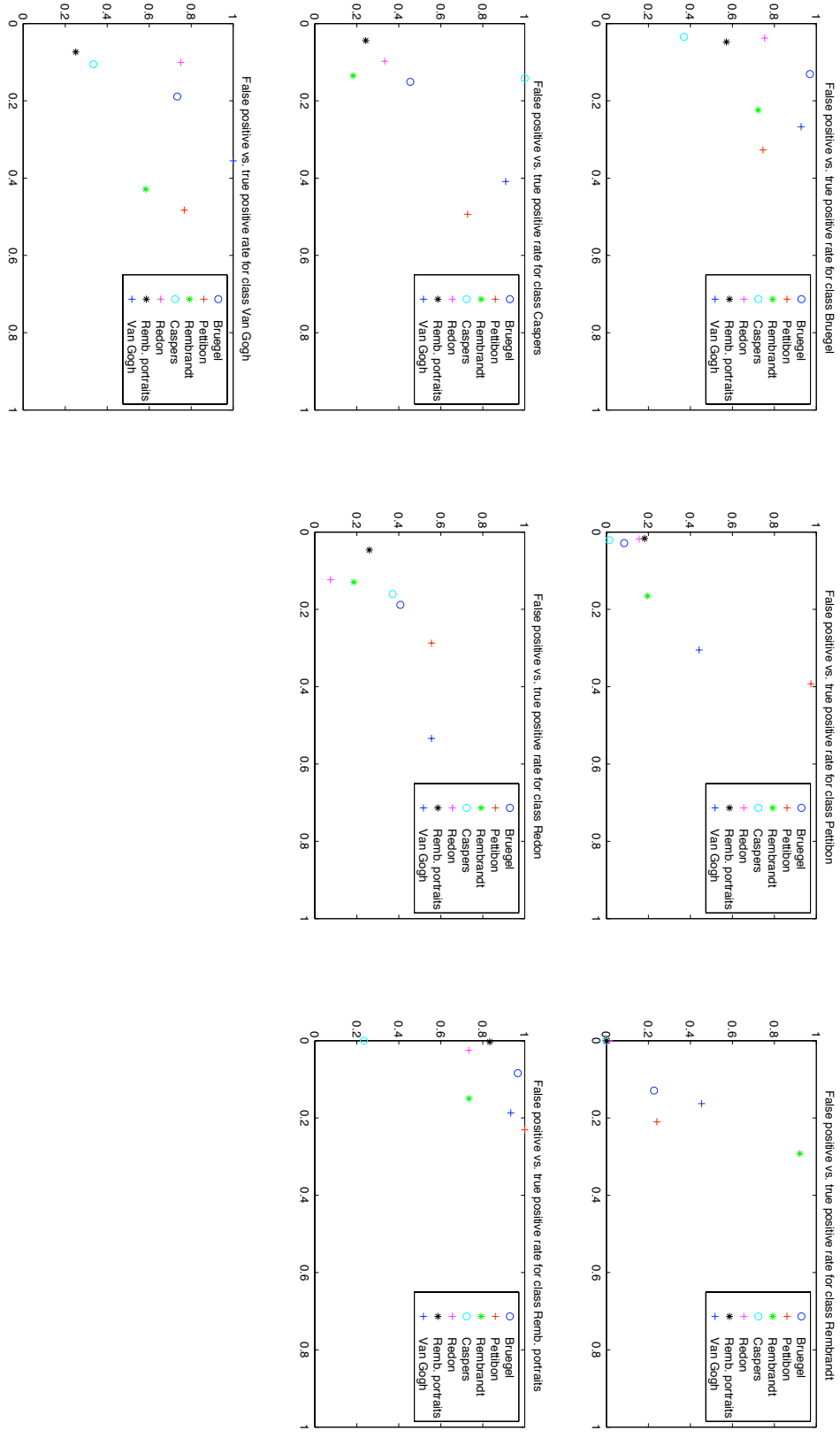

Figure 15: False- and true-positive rates for each category of images indicated at the top of each plot; the plotted symbols indicate the false and true positive rates for the corresponding model. Using an weighted logistic regression model, the generalization performance of all models is significantly better.

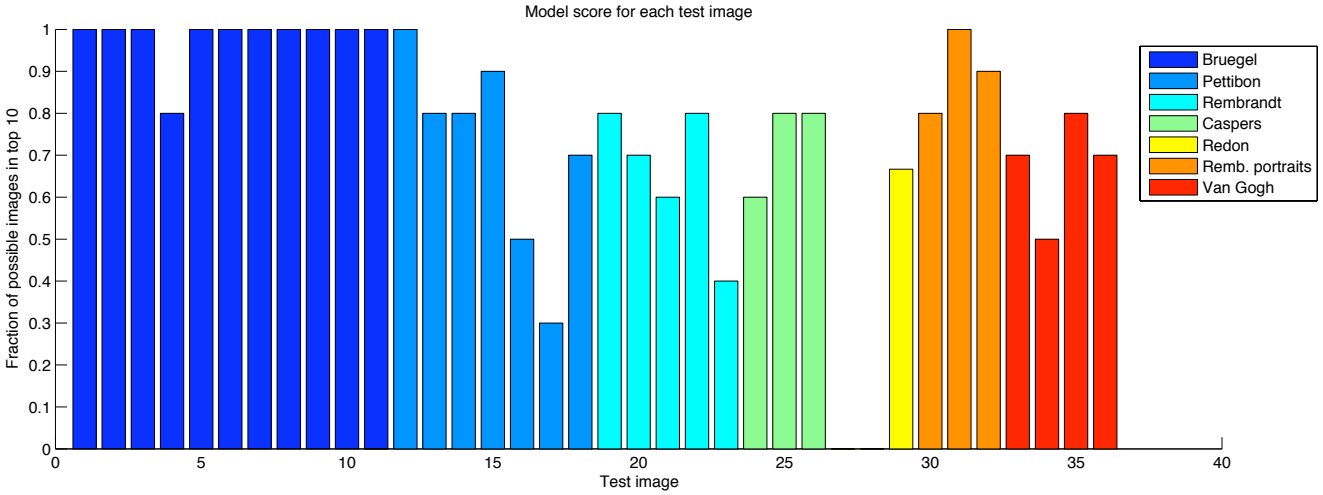

Figure 16: Model scores for testing images in seven styles experiment using weighted logistic regression. The vertical axis represents the fraction of (possible) training images predicted among the top 10 most probable images, given the testing (query) image.

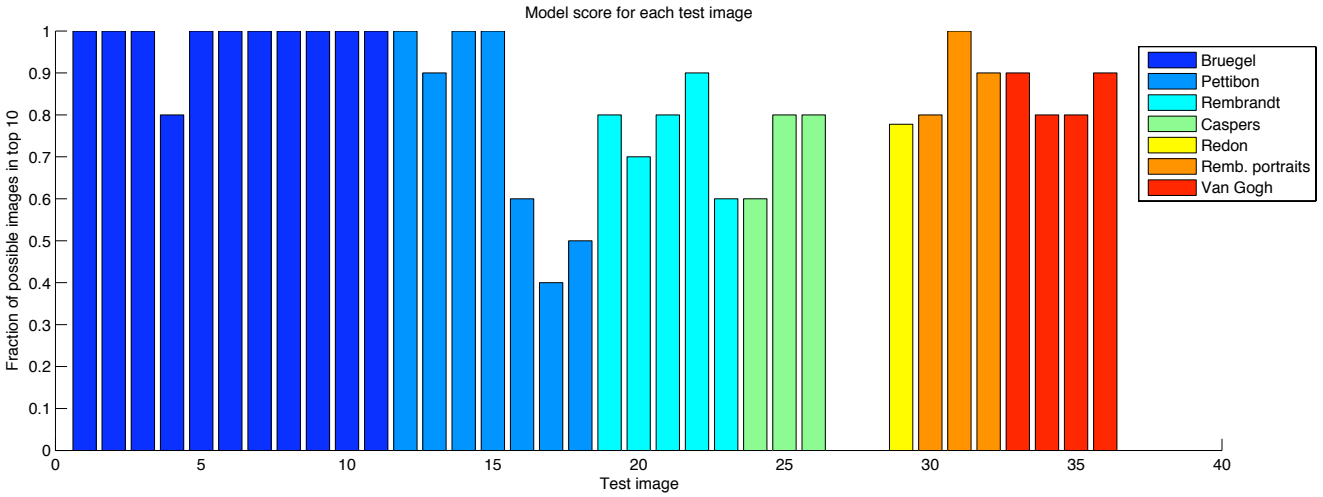

Figure 17: Model scores for testing images in seven styles experiment using unweighted logistic regression. The vertical axis represents the fraction of (possible) training images predicted among the top 10 most probable images, given the testing (query) image.

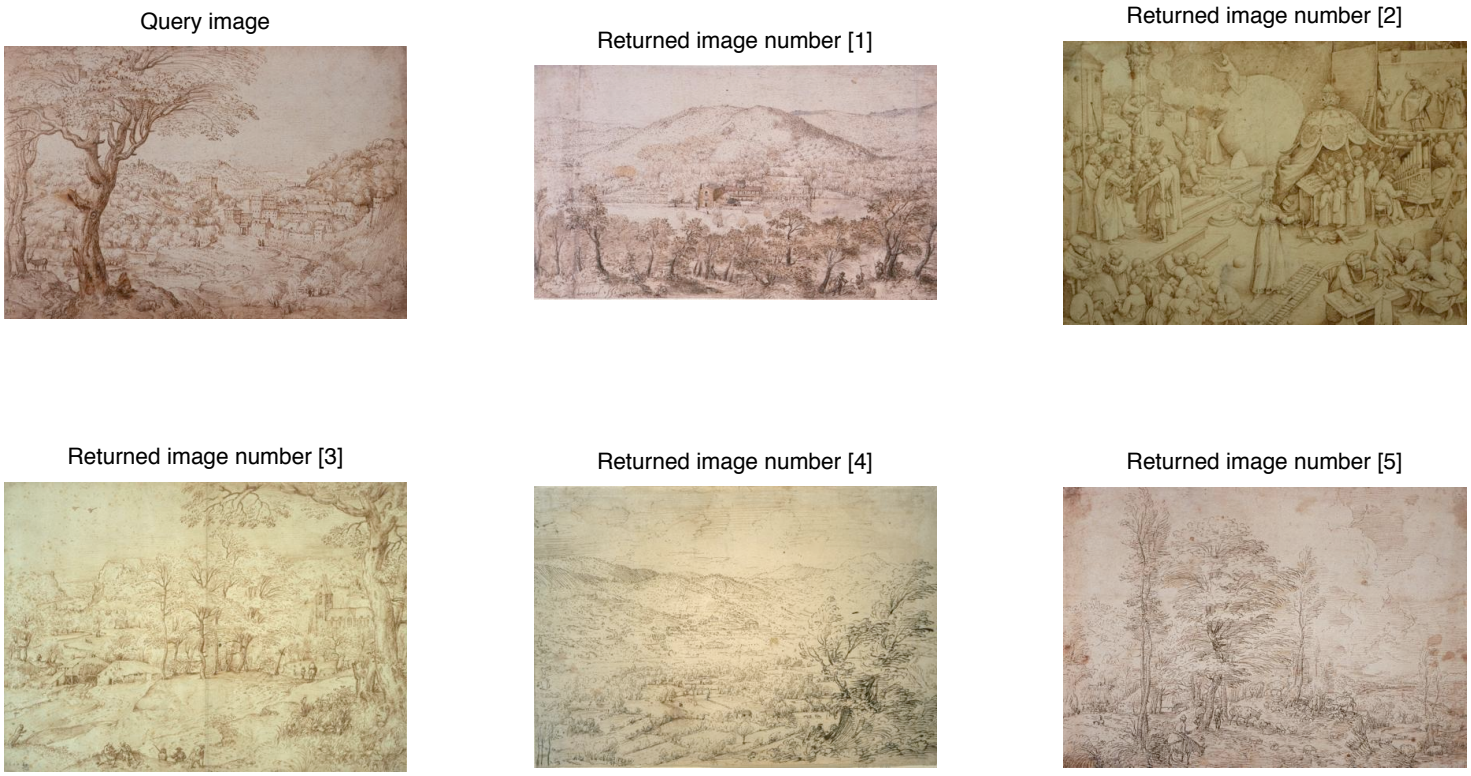

Figure 18: Five most probable images according to learned Bruegel model in "seven styles" experiment, using a Bruegel landscape drawing as the query image. Four of the top five images are also landscapes (and all are drawings by Bruegel), indicating that some meaningful information has been captured by the model.

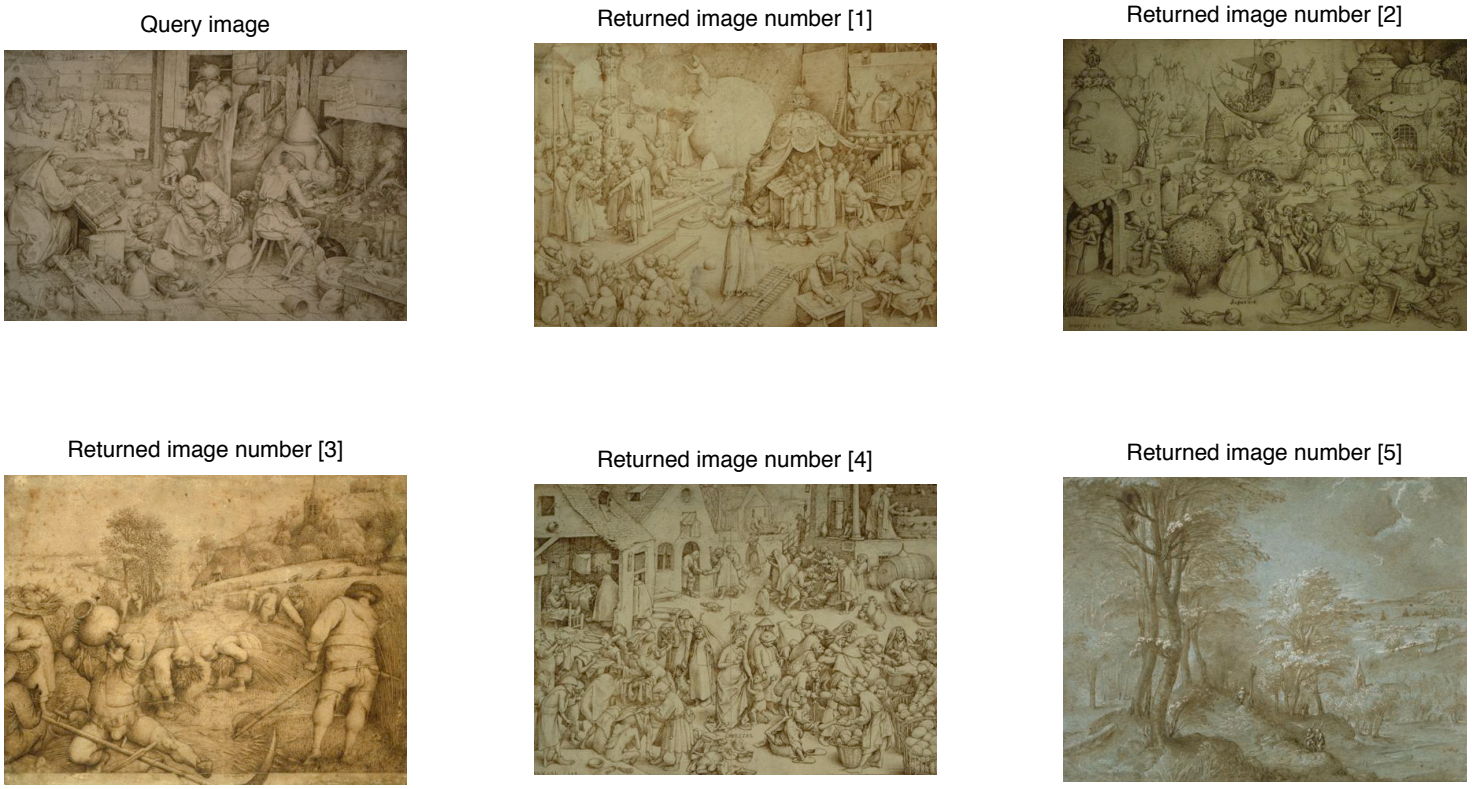

Figure 19: Five most probable images according to learned Bruegel model in "seven styles" experiment, using a Bruegel drawing of a scene with people as the query image. The top four of five images are also scenes with people (all returned images are drawings by Bruegel), and the fifth is a landscape. Once again, it seems that meaningful information has been captured by the model.

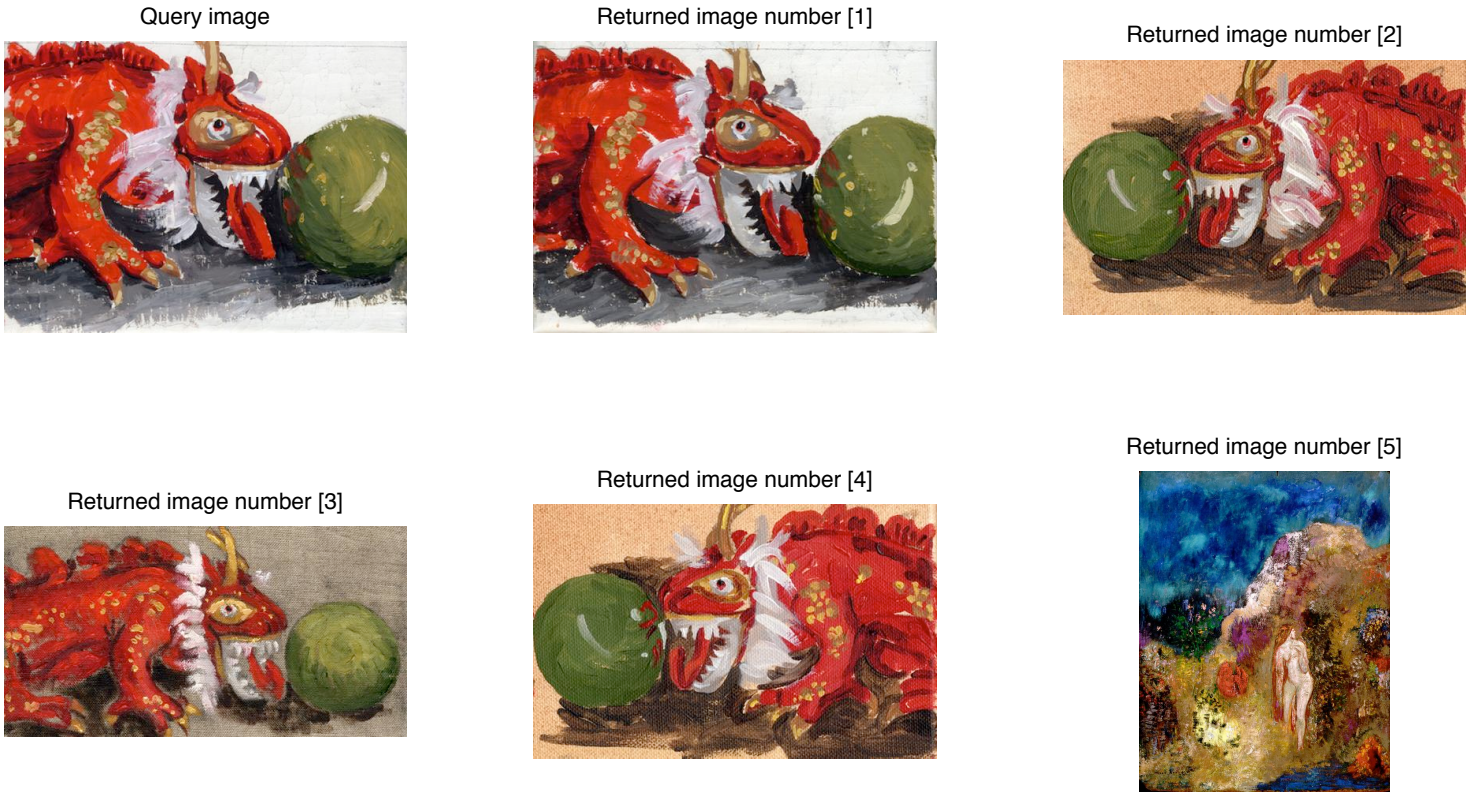

Figure 20: Five most probable images according to learned Caspers model in "seven styles" experiment, using a painting by Caspers as the query image. Four of the top five images are also by Caspers, and are indeed copies of the query image.

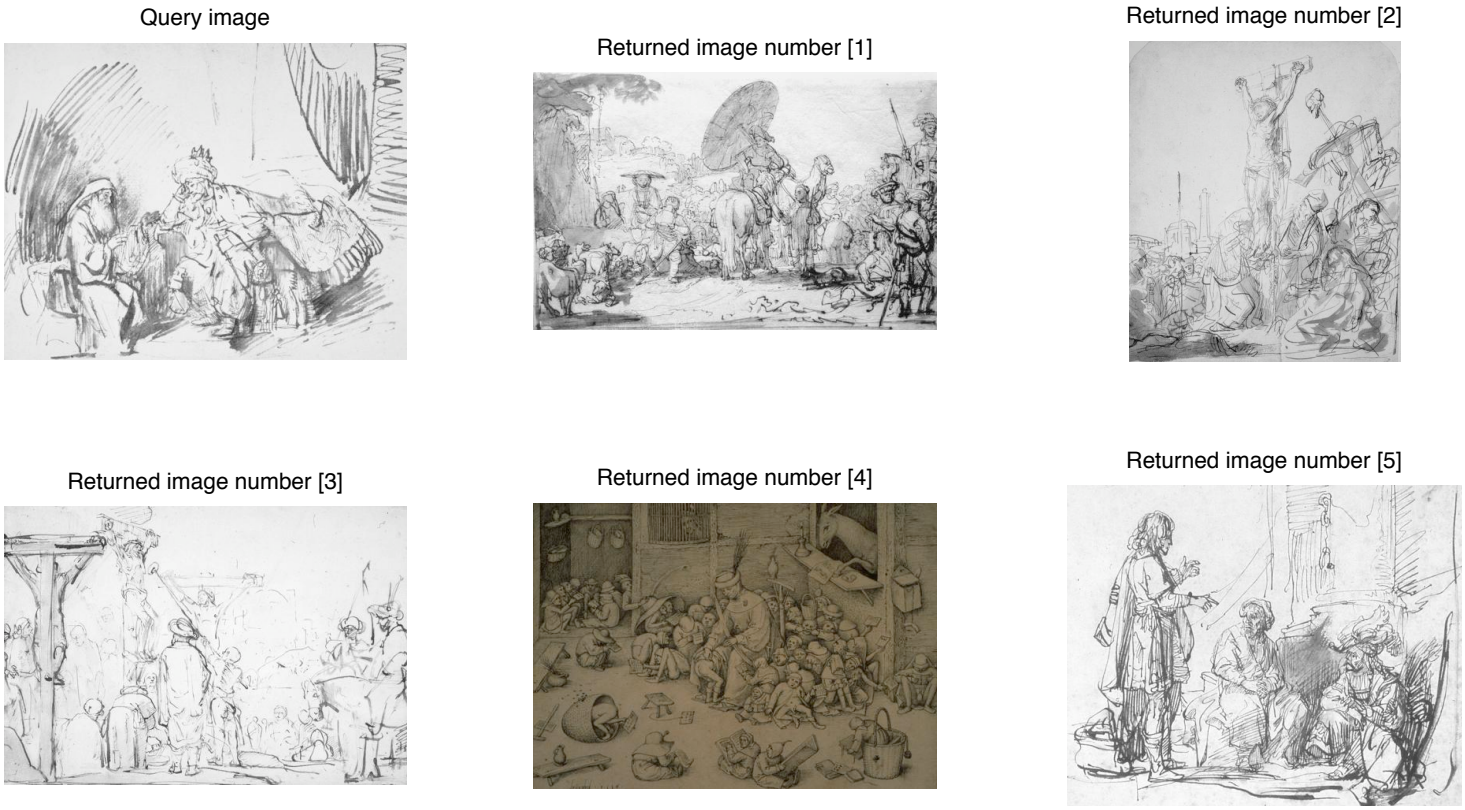

Figure 21: Five most probable images according to learned Rembrandt drawings model in "seven styles" experiment, using a drawing by Rembrandt as the query image. Four of the top five images are also from the Rembrandt drawings set, and the one non-Rembrandt drawing is a Bruegel drawing that contains a scene arguably similar to the query image.

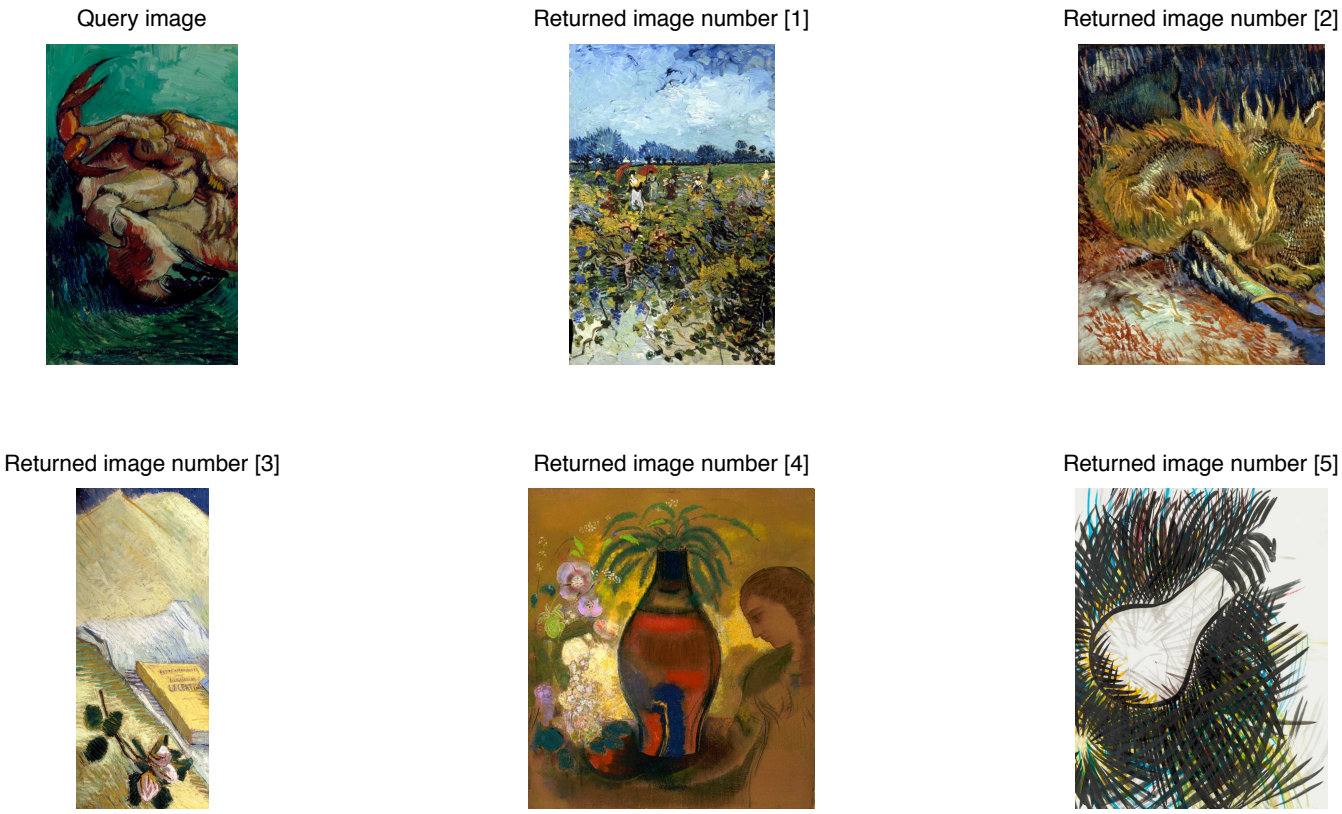

Figure 22: Five most probable images according to learned van Gogh model in "seven styles" experiment, using a painting by van Gogh as the query image. Three of the top five images are also by van Gogh, but the non-van Gogh images show some stylistic similarity according to the types of brushstrokes present in the images.