Using Machine Learning for Identification of Art Paintings

Alexander Blessing
Stanford University
abless@stanford.edu

Kai Wen
Stanford University
kaiwen@stanford.edu

Abstract

Machine learning applications have been suggested for many tasks. We have investigated the suitability of applying machine learning to the problem of art identification, which we believe to be a new, but promising field. Our approach focuses on classifying works of seven different artists, by using a multi-class SVM with state-of-the-art features. Our results indicate that machine learning has good potential to classify art works. We conclude this paper by analyzing our results.

1. Introduction

As machine learning algorithms have advanced over the last decade there has been an increased interest in applying them to a variety of computer vision problems. We have identified art identification as a relevant and challenging activity that could benefit from machine learning algorithms.

Research on automated art identification is very sparse, and only recently has there been an increased interest in applying machine learning to the context of paintings. In 2008, Johnson et al. [1] presented an algorithm for verifying the authenticity of a particular painting, by considering very high-resolution images of Van Gogh paintings. In this paper, we want to examine how machine learning can be applied to the more general problem of identifying an artist's painting. We believe that this can have interesting applications for art historians and scientists, helping them to understand in how far different artists' works are similar, what some objective characteristics of an artist's paintings are, and how machines might be used to authenticate paintings.

In the next section we describe our approach to the problem. In particular, we outline how we obtained our data, what features in the paintings we considered and which classification algorithms we employed. We conclude by presenting and analyzing the results obtained.

Figure 1. Some of the paintings used as training examples. Works by Cezanne, Dali, Durer, Monet and Picasso

2. Method

The goal of this project was to classify a set of paintings, by labeling them with a predicted artist's name. This section first describes our methodology for obtaining training and testing data before exhibiting our selection of features and the deployed classification algorithm.

2.1. Data collection

We focused on a set of paintings from seven different artists: Cezanne, Dali, Durer, Monet, Picasso, Rembrandt and Van Gogh. The main reason for choosing these artists were due to their prolificacy as we anticipated to require a large number of training examples for successful classification. Moreover, we tried to include pairs of artists from both similar (e.g. Cezanne, Monet) as well as different genres (e.g. Durer, Dali).

To acquire the large number of data we created a script that searches for each artist's painting on Google Images and downloads it. Even though our initial approach was to only include high-resolution images ($\geq$ 2 MP), we quickly shifted towards data with less resolution (VGA) as we did not notice a notable decrease in the algorithm's classification rate.

In total, we acquired around 200 images per artist, with a total of over 1400 training examples. Figure 1 illustrates a small subset of images that were used for classification.

2.2. Feature selection and kernels

Our initial choice for the features were the pixels of the images and the color histograms. In particular, we normalized each image to 100x100 pixels when using the pixel features, and shrank the color space to 18-bit colors to reduce the number of features when using the color histograms. Furthermore, a linear kernel was used for training with a SVM.

As we will show in the next section, the accuracy of the basic features was not good enough. We therefore then studied more advanced features, inspired by the work of Xiao et al. on scene categorization [2]. The features we considered include GIST[3], HOG2x2[4], Dense SIFT[5], LBP (Local Binary Patterns)[6], Sparse SIFT histograms[7, 8], SSIM[9], Tiny Images (which is similar to the pixel features mentioned above)[10], Line Features[11], Texton Histograms[12], Color Histograms, Geometric Probability Map[13], and Geometry Specific Histograms[14]. Several different kernels were used for each set of features, including a χ^2 kernel, linear kernel, histogram intersection kernel and Radial Basis Function kernel. The performance for different combinations of feature sets and kernels were evaluated. Finally, weighted by the performance of different feature sets, the features were combined to further improve the accuracy.

2.3. Classification algorithm

In order to quickly implement an initial algorithm and get a better understanding of what features were most helpful, we initially used a Naive Bayes classifier to assign predicted labels to our testing data. After an initial iteration, we decided to use SVM for classification. In particular, we used the LIBLINEAR library [15], using a soft margin with a cost parameter of 1 and with the kernels mentioned in the previous section.

For both the basic features and the advanced features we initially only considered the two-class problem (i.e. the task of classifying two different artists). Encouraged by the performance of the advanced features, we later also considered the multi-class problem, i.e. the task of classifying paintings to one of several artists. For this case we employed a one-vs-all SVM, implemented by Xiao et al. in [2].

3. Results

Before presenting the results obtained with the advanced features, we begin by giving the results for the initial basic features.

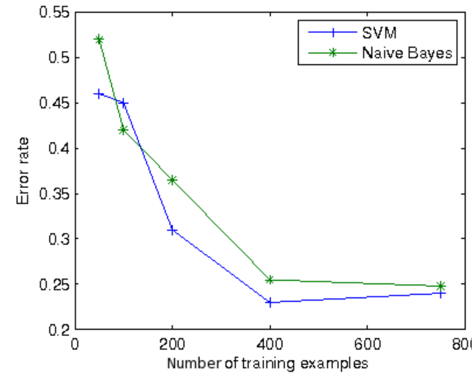


Figure 2. Error rates obtained using the color histogram features in 18-bit color space.

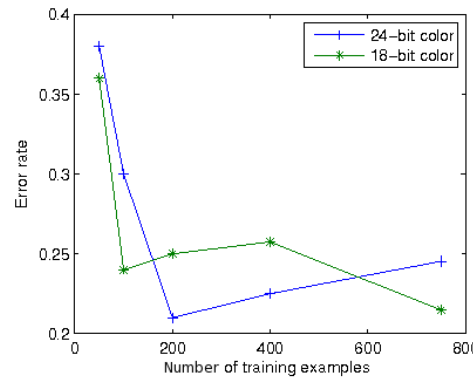


Figure 3. Error rates for the SVM, using the pixel features of normalized 100x100 images.

3.1. Basic features

We picked several training sizes, from 25 up to 375 images for each class (Van Gogh and Dali). We first used a Naive Bayes classifier on the color histogram features, then switched to an SVM and applied it to both the color histogram features as well as the pixel features. Furthermore, we used 10-fold cross validation to calculate the average error rate for both Naive Bayes and the SVM. Besides training with the original 24-bit color pixels, we also reduced the color space to 18 bits so that the feature space be smaller.

Figure 2 illustrates the result obtained by Naive Bayes and by the SVM on the color histogram features (using 18-bit color space), while Figure 3 illustrates the result obtained for the SVM on the pixel features.

The results in Figure 2 and 3 demonstrate that the basic features can achieve at most around 78.53% accuracy for a training set of 750 images. Figure 3 suggests that

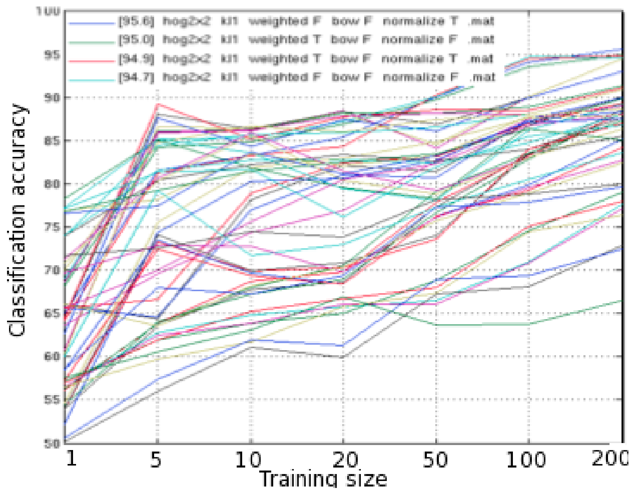


Figure 4. Classification rate for the Van Gogh/Picasso SVM, for increasing size of training data, and for different features. The legend on the top left lists the first couple highest-scoring features.

the model might be overfitted as the error rate increases for larger training sets, implying that the number of features might be too large and that it may be insufficient to hope achieving higher accuracy only by increasing the size of the training set. More advanced features should be introduced.

3.2. Advanced features

We first only considered the two-class problem. The following table lists the results we obtained using an SVM as described in the previous section, and using a weighted combination of all advanced features that yielded the best performance:

Artist Pair	**Classification rate** (%)
Van Gogh/Cezanne	95.9
Monet/Dali	95.2
Van Gogh/Picasso	94.6
Monet/Cezanne	94.5
Dali/Van Gogh	93.6
Monet/Van Gogh	92.1
Picasso/Cezanne	90.2

Figure 3.2 graphically displays the result obtained for the Van Gogh/Picasso problem, under varying training set size and for different features. In this example, the HOG2x2 features gave the best performance, with a classification rate of about 95% with 200 training images.

After training several two-class SVMs, we decided to approach the multi-class problem. In particular, instead of having only two classes of artists, the multi-class problem's goal was to assign training images to one of seven different classes. To solve the multi-class problem we trained a one-

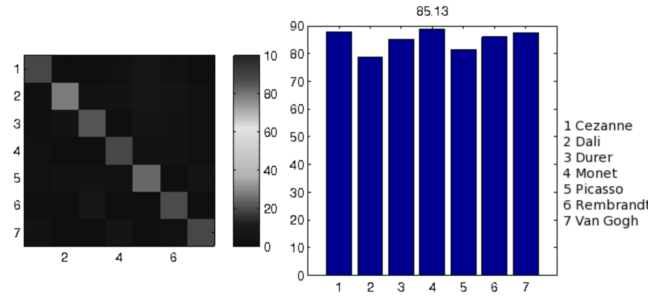


Figure 5. The confusion matrix and the accuracy per class by training with combined weighted features for the multi-class problem with 200 training examples per class.

versus-all SVM for each class and let the classifier with the highest output assign the label. Figure 5 displays the confusion matrix and the accuracy obtained per class, while Figure 6 presents the results for the multi-class problem. Once again, the highest-scoring feature was HOG2x2 with a classification rate of around 82%, while using a combination of all features weighted by the performances of each set of features resulted in an accuracy of 85.3%.

4. Analysis

The results we obtained for both the two-class and the multi-class problem are very promising. In particular, the overall accuracy for the multi-class problem with seven different artists was 85.13%, which is most likely higher than most people's performance. Moreover, it is interesting how

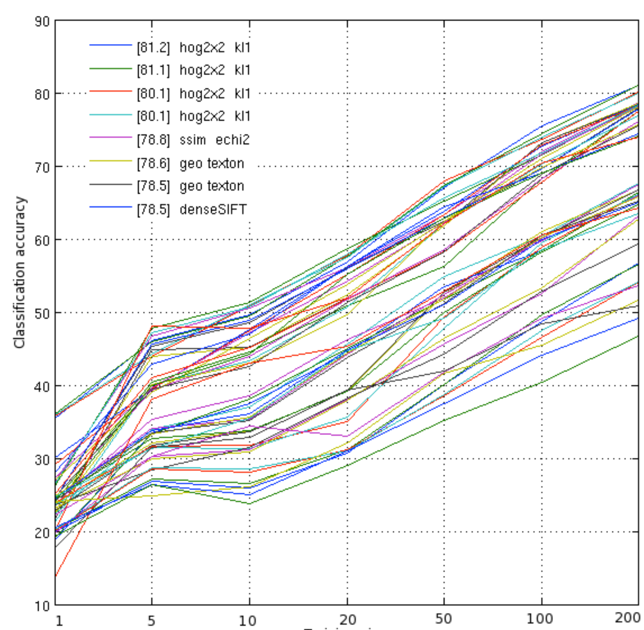


Figure 6. Classification rate for the multi-class problem, where a legend of the highest-scoring features is shown on the top left.

well the advanced features worked in the domain of art identification, while their performance for scene recognition (the domain they were originally used for) is much lower (around 38% [2]). Also, the graphical representation of the results obtained indicate that the performance is highly dependent on the number of training examples. In particular, the graph for the multi-class problem (Figure 6) depicts a monotonic increase in accuracy for increasing training size, so one can hope for even higher accuracy with more than 200 training images per class.

4.1. Similarity between artists

As pointed out in the introduction we think that the machine learning approach to art identification might be able to capture the similarities between different artists in an objective way. To analyze this problem we set out to compute similarities between a subset of the seven artists. In particular, we focused on Cezanne, Dali, Monet, Picasso and Van Gogh and built a total of ten two-class SVMs for all pairs from this subset. We then used the functional margin as the notion of similarity between two classes, where the functional margin is defined as usual as

$$\min_i y^{(i)}(\omega^T x^{(i)} + b)$$

A higher functional margin implies that the two classes are more easily separated and thus - by our definition - less similar to each other than two classes with a lower functional margin. After computing the margins for each pair we used hierarchical clustering and depicted the result by a dendrogram, as shown in Figure 7.

The dendrogram arguably captures some intuitive notion of similarity. For example, Monet and Cezanne - both attributed to (post-)impressionism - are deemed most similar. However, baring in mind that Picasso is considered a Cubist painter, the close relationship between Picasso and the impressionists is rather surprising.

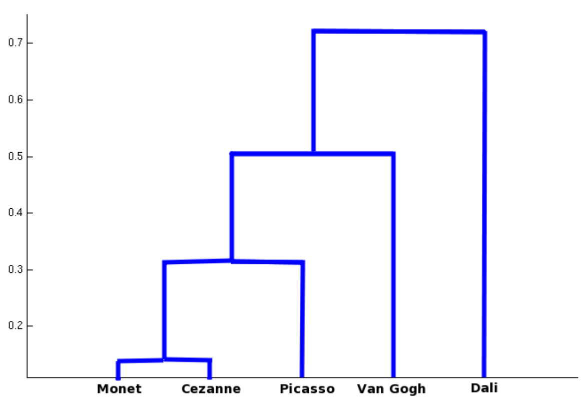
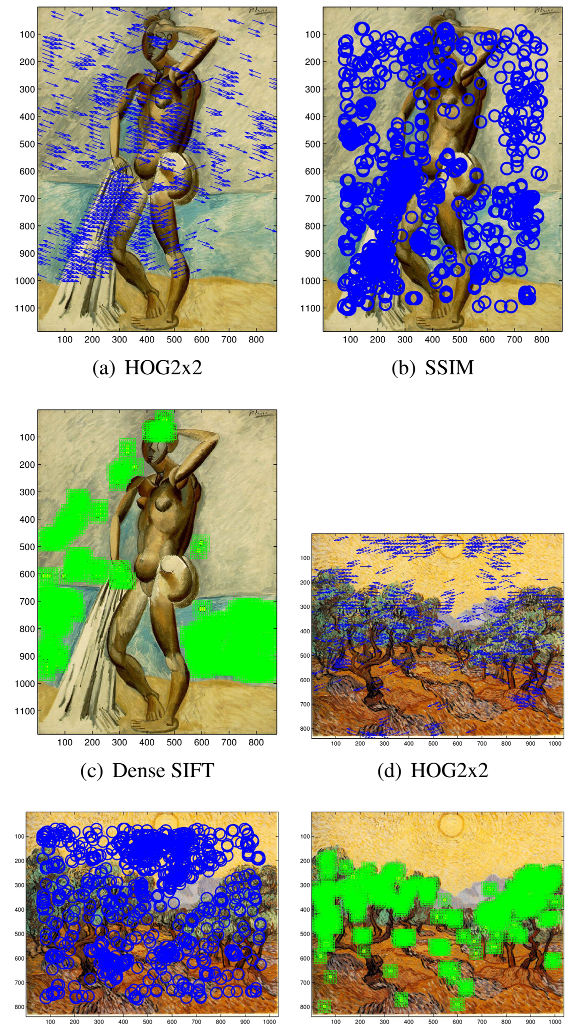


Figure 7. Hierarchical clustering performed on the distance tree obtained from all-pairs SVMs and using the functional margin as a measure of distance.



(a) HOG2x2 (b) SSIM (c) Dense SIFT (d) HOG2x2 (e) SSIM (f) Dense SIFT

Figure 8. Visualization of three different features on two images: Picasso's for the first 3 figures and Van Gogh's for the last 3.

While the validity of the distance tree might be disputable, it does reveal some interesting relationships. We believe that it would be worthwhile spending more time analyzing these similarities.

4.2. Feature performance

We think that an analysis of the performance of various feature set is able to tell us more about the styles for each artist. From Figure 6, HOG2x2[4], SSIM[9], geo texton (texton histograms for 4 geometric class: ground, vertical, porous, and sky)[14], and Dense SIFT[5] have the top performance. We first compare HOG2x2, SSIM, and Dense SIFT by visualizing the top 800 keypoints with the highest histograms for each feature set, as shown in Figure 8. The blue arrows give the most significant keypoints for HOG2x2 and their orientation gradient histograms, the blue circles illustrate the keypoints for SSIM with radius proportional the histogram, and the green blocks show the keypoints for Dense SIFT in which the gradient vectors are drawn for each of the 4x4 grids.

In Figure 8, it turns out that in the portrait images (Fig. 8(c)), Dense SIFT is not good at capturing the inside figure. In the landscape images (Figure 8(f)) Dense SIFT can capture trees, one of the most important elements, but not the sun. This suggests that Dense SIFT may be more suitable to classify artists who have more landscape paintings. SSIM is better, although still capturing many irrelevant areas. On the other hand, HOG2x2 demonstrates the most balanced keypoints on both types of images. Therefore, HOG2x2 has the best performance. However, if an artist can be classified more accurately with Dense SIFT, then this might be an indication that the artist prefers landscape images to portrait images.

5. Conclusion

We demonstrated a successful application of machine learning to artist identification. By using the combination of various computer vision features, we were able to achieve 85.13% accuracy in identifying images from a pool of seven artists. The results also helped us learn about the similarity between different artists. By comparing the performance of different features, we moreover found out that HOG2x2 has the best performance in general since it can capture more balanced keypoints over the images. Finally, we argued that the performances of the different features might imply the preferences and styles of the artists.

References

[1] C.R. Johnson, Jr., E. Hendriks, I.J. Berezhnoy, I.J., E. Brevdo, S.M. Hughes, I. Daubechies, J. Li, E. Postma, and J.Z. Wang. Image processing for artist identification. *Signal Processing Magazine, IEEE*, 25(4):37-48, 2008. 1

[2] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. SUN Database: Large-scale Scene Recognition from Abbey to Zoo. *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. http://groups.csail.mit.edu/vision/SUN/ 2, 4

[3] A. Oliva, and A. Torralba. Modeling the shape of the scene:a holistic representation of the spatial envelope. *Intl. J. Computer Vision*, 42:145175, 2001. 2

[4] N. Dalal and B. Triggs. Histogram of oriented gradient object detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005. 2, 4

[5] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 21692178, 2006. 2, 4

[6] T. Ahonen, J. Matas, C. He, and M. Pietikäinen. Rotation invariant image description with local binary pattern histogram fourier features. In *SCIA*, 2009. 2

[7] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust widebaseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761 767, 2004. 2

[8] J. Sivic and A. Zisserman. Video data mining using configurations of viewpoint invariant regions. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004. 2

[9] E. Shechtman, and M. Irani. Matching local self-similarities across images and videos. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007. 2, 4

[10] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: a large database for non-parametric object and scene recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(11):19581970, November 2008. 2

[11] J. Kosecka, and W. Zhang. Video compass. In *Proc. European Conf. on Computer Vision*, pages 476490, 2002. 2

[12] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. IEEE Intl. Conf. on Computer Vision*, 2001. 2

[13] D. Hoiem, A. Efros, and M. Hebert. Recovering surface layout from an image. *Intl. J. Computer Vision*, 75(1), 2007. 2

[14] J.-F. Lalonde, D. Hoiem, A. A. Efros, C. Rother, J.Winn, and A. Criminisi. Photo clip art. *ACM Transactions on Graphics (SIGGRAPH)*, 26(3), 2007. 2, 4

[15] Rong-En Fan et al.. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, 9:1871-1874, 2008. 2