Computer Vision, Final Project, May, 2013
Professor Austin David Reiter

Masterpiece Matcher - Identifying Art Pieces by Era using Machine Learning

Cole Diamond, Nikita Nadkarni, Patrice Liang, Orlando Pineda

Abstract

We explored the idea of using machine learning in conjunction with computer vision to identify different art periods from images of various art pieces spanning four major epochs: Renaissance (1300 A.D. - 1602 A.D.); Neoclassicism (1602 A.D. - 1709 A.D.); Romanticism (1789 - 1880 A.D.) and Modern (1880 A.D. - 1945 A.D.). Using a database of 1700 images per period gleaned from *Web Gallery of Art*, we created three matrices from HOG, Dense SIFT, and RGB histograms features. Garnered from our computed feature set, we constructed One-Versus-All Support Vector Machines (SVM) with a linear kernel and Random Forests and compared classification accuracies. In order to boost the performance of our Random Forests, we varied tree depth and branching factor. Moreover, we employed a 70% / 30% training data to test data split in determining classification accuracy. Overall, Random Forests using RGB histogram features achieved the best observed classification accuracy of 47.3%, far exceeding our baseline of 25%. For the SVM models, Dense SIFT features performed the best with a classification accuracy of 40%. Our results indicate that utilizing Random Forests is fairly successful in correctly identifying art epochs.

Contents

Introduction

After researching several object recognition techniques, we hoped to apply classification algorithms to other fields that have yet to be thoroughly explored. For this project, we chose to apply computer vision algorithms in tandem with machine learning to the visual arts. Specifically, we examined the problem of art period classification. In 2010, Alexander Blessing and Kai Wen published a paper at Stanford University on the use of machine learning to identify paintings by artist vis a vis analysis of high definition images. We opted to apply an analogous technique to artistic periods, which we believe adds a different scope of use to the project. By recognizing artistic period, our project allows for a greater breadth of data to be analyzed, and can be used by a wider variety of people, such as artists, art historians, students and tourists. Additionally, the project could also be used for art authentication.

In our paper, we delineate our initial algorithmic approach to this problem, our methods for data collection, our processes, the features we tested and used, and our results and analysis of the project as a whole.

1. Methods

1.1 Data Collection

In order to obtain images to train our models and to conduct our testing, we scoured the web for an online database that contained a sufficient number of images. We found the Web Gallery of Art, which had around 40,000 images of paintings created between the years 1000-1900 A.D., meaning it would supply us with images for the following artistic periods: modern, renaissance, romantic, and neo-classical. In order to download all 40,000 images, we first downloaded a spreadsheet provided by the Web Gallery of Art, which included a date section and a url for more information about the image. We then converted the spreadsheet into a csv file. Using this csv file we wrote a web scraping script in python which downloaded the image using a reconstructed url(since the path to the image was different than the path to the information) and placed the image into the folder of the era that the image corresponded to, using the date field of the csv. The images were named as increasing integer values as the folders grew in size. This was done so that we could easily iterate through the images if needed.

The next step was to import the data into matlab, compute the necessary feature descriptors and store all of this data in a meaningful way for each image . We did this by creating a struct organized by era. For each image within an era, we scaled the image to 100x100 pixels and computed features. Moreover, we normalized each feature matrix to have zero mean unit variance to boost the decision tree accuracy.

Initially, we were too ambitious and attempted to create a struct that held all of the images that we downloaded, along with their computed descriptors. This created a file that exceeded our dropbox memory and exceeded the possible matrix dimensions when trying to feed it all into our SVMs. Instead, for the SVM stage, we used 500 images per era to train our SVMs.

For the random forest phase of our experiment, we split all of our data so that 70% would be used to train our model and 30% of our data would be used to test it. Instead of organizing the data as a struct we instead computed hog,dsift, and rgb histograms for each image and combined them into a giant feature descriptor matrix which would be used to train our models.

Figure 1. Database Images. Renaissance on the left, Neoclassicism on the right

Figure 2. Database Images. Romanticism on the left, Modern on the right

1.2 Feature Selection

After copious research and through consultation with the Blessing and Wen paper, we decided to use three different techniques to obtain our feature descriptors. We used Dense SIFT, HOG, and an RGB color histogram to compute the feature descriptor.

We used Dense Scale Invariant Feature Transform (Dense SIFT) to compute SIFT on a dense grid of locations at a fixed scale and orientation. According to the Blessing and Wen paper, Dense SIFT technique is particularly efficacious in the realm of landscape images and natural scene backgrounds. We also employed Histogram Oriented Gradients (HOG) which are primarily useful for human or animal shape detection in images, partly due to the density of the grid used to compute the descriptors and the high level of overlapping local contrast normalizations.

1.3 One-Versus-All Multi-class SVMs

Initially, we implemented the approach of using SVMs with a linear kernel, support vector machines, for training and testing on 70% and 30% of the data, respectively. The matrix rows were randomly permuted before partitioning. Then, we took the training data set and trained one vs. all SVMs for each of the 500 images for each of the four artistic periods. We built the training data matrix for each feature, which was later used to train each of the one vs. all SVMs.

Once the SVMs were trained, we evaluated the performance of each of the SVMs using the testing data and computed accuracy percentages for each of the features. We trained four one-versus-all SVMs, one for each period of classification, and labeled each painting in our testing set by selecting the period that yielded the maximum inner product between the painting's feature descriptor and the hyperplane of the period's SVM. We then chose the period which resulted in the largest value as the classification label. Then we compared our predicted label to the actual label, and computed the accuracy from the mean of the classification accuracies. It was at this point that we began to realize that the accuracy we were obtaining was not at a desirable level and began to explore other classification algorithms that could provide us with better performance.

1.4 Random Forests

Our second attempt for classification involved the use of Random Forests. As previously mentioned, we divided our data into two parts. 70% of our data was used to train our Random Forest and the remaining 30% was used to evaluate its performance. Since we had already computed the feature descriptors for SVM classification and stored that data externally, we were able to reuse this data for our Random Forests. Our approach in using Random Forests was as follows:

For each feature, we loaded our stored data structs containing the computed feature for all images in all periods. We then randomly chose 70% of the data to use for training our Random Forests. We trained the Random Forests using this 70% of the data and once the trees were trained we began testing the accuracy of our model. The accuracy was, just as before, computed by comparing the prediction of the Random Forest to the actual answer, which we had precomputed when selecting our training data.

Thereafter, we varied the split factor and depth of the decision tree in order to boost classification accuracy. After several preliminary tests, we observed little performance boost and noted increase in computation time from increasing the number of decision trees in our random forest. As a result, all random forests were trained using a total of 100 trees.

2. Results

2.1 One-Versus-All Multi-class SVMs

We first present the results obtained by our initial approach of using SVMs. Training the SVMs using 1700 images per period, with a total of 5400 images, we obtained a classification accuracy of 25% using HOG features. RGB features yielded an improved accuracy of 30%, and dense SIFT features yielded the highest accuracy of 40%.

Feature	Accuracy (%)
HOG	25%
DSIFT	40%
RGB	35%

Figure 3. SVM classification accuracies using HOG, DSIFT, and RGB features

2.2 Random Forests

Seeking to improve the performance of our classifier, we then decided to train and test our classifier using Random Forests in place of SVMs. We trained the Random Forests using various depths and splits and achieved overall improvements in accuracy for all feature descriptors. As with SVMs, the lowest performing feature with Random Forests is HOG, with a maximum accuracy of 42% with depth 11 and 10 splits. Dense SIFT yielded a maximum accuracy of 46.5% with depth 11 and 10 splits. RGB feature descriptors yielded the best results, with an accuracy of 47.3%.

RGB (% ACCURACY)	splits = 5	splits = 10
9	44.6	47.3
10	45.7	46.2
11	42.3	45.8
HOG (% ACCURACY)	splits = 5	splits = 10
9	39.1	39.5
10	40	39.5
11	41.9	42
DSIFT (% ACCURACY)	splits = 5	splits = 10
9	44.1	45.5
10	43	46.4
11	44.5	47

Figure 4. Random Forest classification results using 100 decision trees and RGB, HOG, and DSIFT features.

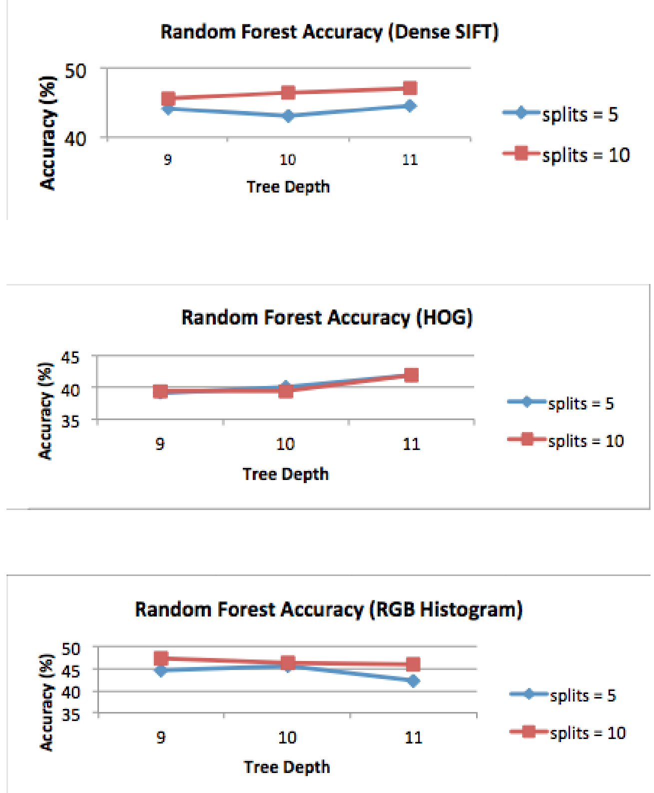

Figure 5. Graphs of Random Forest results with varying tree depths and splits using a) Dense SIFT, b) HOG, and c) RGB features.

3. Analysis

Though our initial approach of using SVMs yielded underwhelming classification rates, with the highest being 40% using dense SIFT features, we were able to improve upon this with Random Forests trained on RGB features, which yielded an accuracy of 47.3%. These results are not ideal for a classifier, but they provide meaningful information regarding training and feature performances. Most importantly, the classification accuracy far exceeds the baseline accuracy of random guessing, which is 25%.

3.1 Training performances

The SVM approach did not perform as well as we had speculated. This may be attributed to the fact that we did not calibrate the slack parameter c using k-fold cross-validation, since computation time for training our SVMs was expensive. K-fold cross-validation may have yielded better results for determination of the regularization parameter, since it is common for an artistic period to have smaller subsets of paintings that are similar to each other in style. For the purposes of this project, we chose to experiment with a second method (Random Forests) instead.

Since decision trees are able to maximize information gain amongst disparate features, the classification device may have been better able to detect the nuances in period-to-period art paintings. Random Forests are able to identify the most discriminative attributes in the data through averaging output labels across multiple decision trees. The superior performance of random forests over SVMs observed in this project indicates that there exists many subtle attributes in the paintings.

3.2 Feature performances

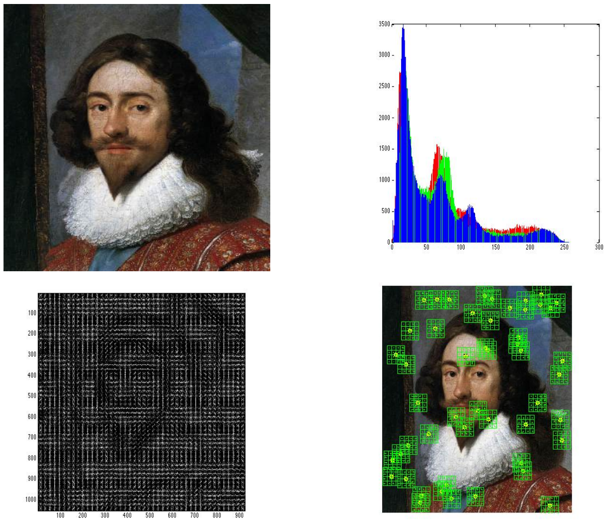

Figure 6. Different Feature Descriptors for Neoclassical Portrait. From top left to bottom right: Portrait, RGB Histogram, HOG, and DSIFT.

We used different features in the classification problem to determine whether certain features are more powerful than others in distinguishing artistic periods. In general, the dense SIFT and RGB feature descriptors yielded more accurate results than the HOG descriptors. This is expected, as HOG descriptors detect gradient orientation, an attribute that does is not necessarily characteristic of an artistic period.

Although the dense SIFT features yielded a higher accuracy than RGB features under SVM training with a linear kernel, the highest overall accuracy was obtained using RGB features under random forest training. This indicates color palette may be an attribute characteristic to different artistic periods. In other words, there may have been trends in color scheme or subject matter during each period.

Period	Accuracy (%)
Modern	46.65
Neoclassical	43.27
Renaissance	53.9
Romanticism	38.12

Figure 7. Classification accuracies within periods using RGB-trained Random Forests of 100 trees and 5 splits.

Another interesting point of comparison is performance within periods using the approach that yielded the best results, namely Random Forests trained on RGB features. As shown in Figure 7, classification within the Renaissance period had the best performance, with an accuracy of 53.9%. From this we can infer that color was most uniform in Renaissance paintings. This can largely be attributed to the fact that the Renaissance period was fairly limited to mythological and religious subject matters that tend to involve similar tones of primary colors.

The classifier performed least favorably for the Romanticism period, yielding a classification rate of 38.12%. As this period abounded in landscape paintings, we can infer that RGB feature descriptors are less successful at detecting similarities across paintings with nature as the primary subject matter.

4. Conclusion

We successfully demonstrated that paintings belonging to the same artistic period, namely the modern, renaissance, romantic, or neo-classical period, share enough similarities in features to be used to train a masterpiece classifier that classifies by period. In training, we found that Random Forests yielded superior results to SVMs. In terms of features, we found that the RGB features yielded the best results, suggesting that color is one of the most defining features of an artistic period. Finally, our results indicate that RGB features work best with classifying paintings from the Renaissance period, suggesting that color is most defined as a characteristic in the Renaissance period.

5. References

Blessing, Alexander, and Kai Wen. *Using Machine Learning for Identification of Art Paintings.* Stanford University, 2010. Web. Apr. 2013.

Dalal, Navneet, and Bill Triggs. *Histograms of Oriented Gradients for Human Detection.* LEAR, n.d., 2005 Web. Apr. 2013

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.

N.d. Comp. Emil Kren and Daniel Marx. N.p.: n.p., n.d. Web Gallery of Art. Web. Apr. 2013.