art classification using Computer Vision

Abstract

The aim of the project is to study paintings through the eyes of a computer. If Computer Vision is the science of making computers "see", then the ability to notice the nuances in a painting is a lofty and challenging goal.

I use various standard CV image filtering techniques along with some improvised feature generation to build a model that can differentiate between the paintings of different artists. It's a difficult goal given that even human recognition for the task is not perfect.

Motivation

Computer Vision can assist art historians in analyzing paintings – since it can manipulate a dataset much larger. It can also perform objective analysis and discover subtleties in an image not visible to the naked eye.
Apart from that it would be great development if CV algorithms could be used to understand paintings, which are inherently a very human intensive and nuanced task.

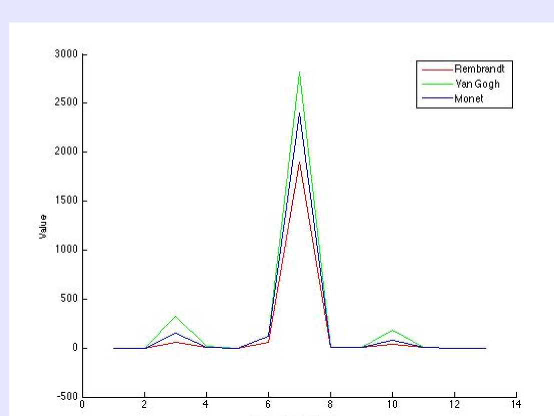
Texture

I have used Haralick features to determine the texture of the painting. Haralick features are computed using the gray-level co-occurence matrix of the image. For each gray level, the number of occurrences for each other gray level value is stored for various directions (horizontal, diagonal etc.). Then 14 different statistical measures such as correlation, moment, entropy, variance etc. are calculated from the image. It's one of the most effective metrics.

Linear – 67.69%
Chi-Squared – 71.79%

Confusion Matrix

8	2	0
0	11	5
2	2	9



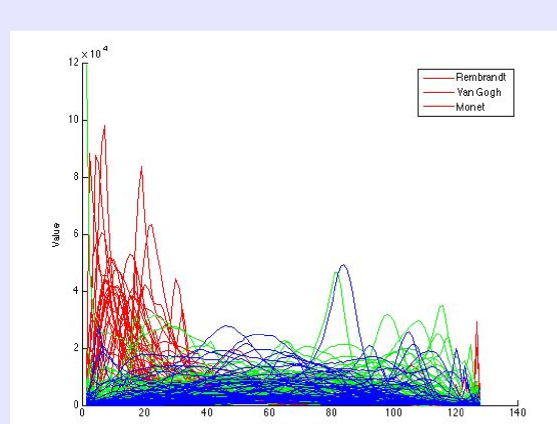
Intensity

While there was clearly some pattern in the intensity distribution of the paintings, classification was harder due to the noisiness of data – possibly due to the limited amount of data. Various statistical measures of the intensity such as the mean, STD, skewness, kurtosis, % bright and dark pixels, peak intensities, gradient co-efficients, number of values above threshold etc. were experimented with. In the end a subset was chosen for optimized results below.

Linear – 67.69%
Chi-Squared – 71.79%

Confusion Matrix

8	2	0
0	11	5
2	2	9



Dataset

The dataset used for the experiments consists of 3 different master painters – Rembrandt, Van Gogh and Monet. They were chosen devoid of and pre-conception of style. Rembrandt is know for luxurious brushwork, rich colors and a mastery of chiaroscuro (This was particularly helpful in classification given the drastic changes in intensity). Van Gogh is a post-impressionist painter, who painted with meticulous brushstrokes and bold colors. Monet too was an impressionist painter and drew many landscapes. The dataset consisted of 40 Rembrandt, 65 Van Gogh and 54 Monet paintings.

Conclusion

The project was able to gain some differentiation between the paintings, but it was far from perfect. Features characterizing. While I dabbled with various simple statistical measures to differentiate brushstrokes through edge analysis, it wasn't very successful, but is promising for future work.
Intensity distributions worked reasonably well with small statistical measures but were quite noisy. This could be owing to the limited amount of data. However, they showed promise.

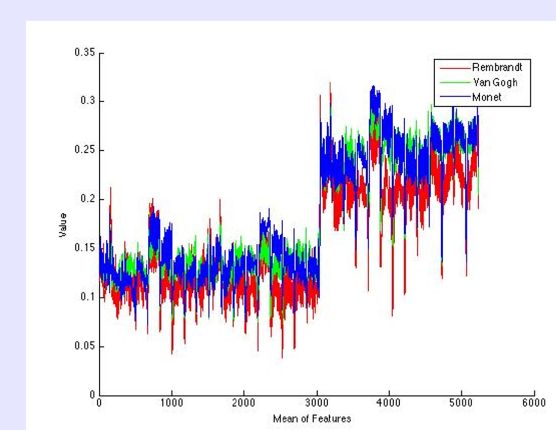
Histogram Oriented Gradient

I use HOG features for key-point detection within paintings. First, each image is resized to a 100 x 100 image since HOG is scale invariant. After that it calculates descriptors of size 13 x 13 x 31, given a cell size of 8. HOG gives the best performance of all the features and that's largely due to the fact that it is able to exploit the nature of the brush strokes of painters.

Linear – 73.84%

Confusion Matrix

8	1	1
2	12	2
0	3	10



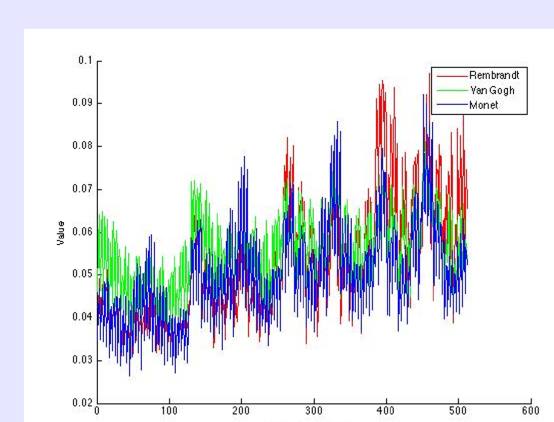
Shape (GIST)

The GIST descriptors are used to characterize the overall shape and spatial nature of the image. It is computed by first resizing the image to 256 x 256, and then over 4 different scales, with eight orientations per scale.

Linear – 66.64%
Chi-Squared – 67.68%

Confusion Matrix

9	1	0
1	9	6
1	2	10



Method

I first studied the dataset to look for traces of differentiating factors between the images. While at a cursory level, Rembrandt paintings were often characterized by deep intensity changes, there wasn't much difference between Van Gogh and Monet. All 3 painters used bright colors – Van Gogh had slightly characteristic brush strokes and Monet had bright distributions of colors in his paintings. Noticing these, features were used to capture the following aspects of the paintings.

Color | Texture | Intensity(Luminance) | Lines(Brush Strokes)

SVM and Kernels

SVM was used as the default Machine Learning algorithm since it provides good accuracy and thus allowed me to spend more time on the features. Also, it can easily be complemented with different kernel methods.
I used a linear SVM, a Chi Squared kernel, histogram intersection and RBF kernels for testing. Chi-Squared and Hist Intersection are known to perform well on image descriptor histograms. However, I found that the linear SVM and Chi-Squared kernel performed the best.

Testing Method

All the accuracies reported are an average over 5-fold cross-validation results. Given the limited size of data, 25% was used as the test set.

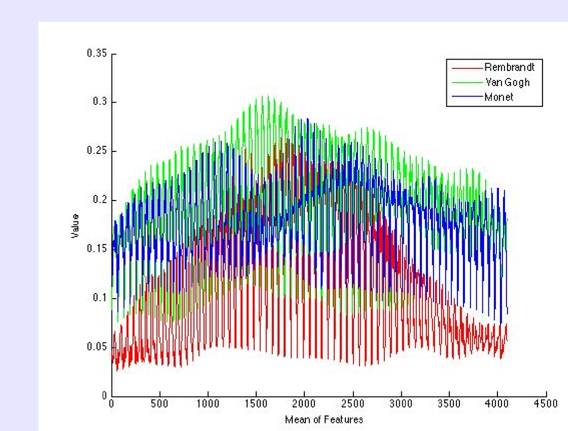
Saliency Map

Saliency maps are used to detect the most important points in an image. They try to model the points in the image that are most likely to draw visual attention. The implementation used is a simple one and based on the **Itti** algorithm. The main idea is to compute the dissimilarity of a pixel from its neighbors.
The image is first resized to 100 x 100, and the saliency map is calculated with a blurRadius of 4 and standard deviation of 4. 4 Gabor angles (0, 45, 90 and 135) are used.

Linear – 56.47%

Confusion Matrix

7	2	1
4	9	3
3	4	6



Other Features (Not very discriminatory)

There were various other features that I experimented with but have not been reported due to the insufficiency of their performance. Listing them as follows.

Edge Distribution

The number of connected pixels of length 4, 8 were computed. Then their total number and a histogram over the sizes of each of the connected regions was computed. However, it didn't prove to be discriminatory enough giving average performance.

Wavelet Transforms

Brief experimentation with wavelet transforms was carried out to find high frequency content, but wasn't successful.

Attempts to measure graininess using blurring by Wiener filters