We believe texture analysis will improve the overall result of our system since different paintings may have similar colors and features but may be of two different mediums. For example, oils paints are thicker than watercolor paints. Even though an oil painting of a city landscape contains the same features and colors as a watercolor painting of the same city landscape, the texture of the oil painting may be more rugged than the watercolor paintings.

We will use local binary patterns texture matching, a well-studied technique that is invariant to illumination as well as translation, the two primary ways our inputs will differ from our outputs. LBP matching works by forming an LBP descriptor, which is calculated by choosing n points on a circle of specified radius around a reference pixel and creating an n-bit number, where the ith bit is set to 1 if the ith value is greater than the reference pixel value, and 0 otherwise. This process is repeated for every pixel in the gray-scaled image, and thus the LBP descriptor is the same size as the original image. After the LBP descriptor has been created, a normalized histogram of the values is created. This process is repeated for every image in the training set.

In order to classify an image, the same process is repeated to create a normalized histogram of the LBP values. Then, a distance metric, such as chi-squared, is used to compare the test image histogram with the training data histograms. The most similar image—or the image with the lowest chi-squared error—is returned.

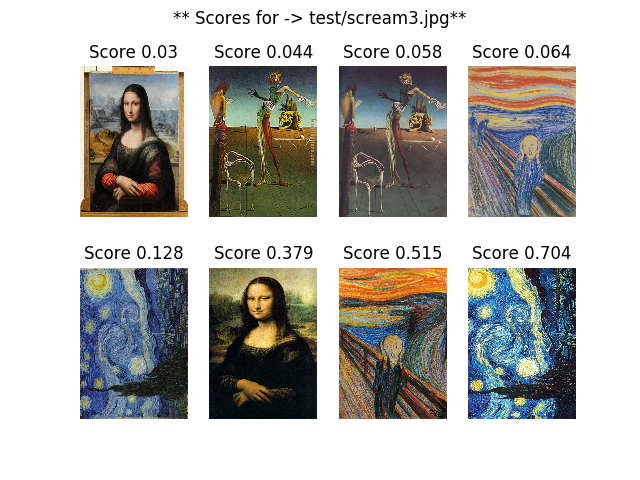
**Experiments and Results:**

The implementation for this experiment was based on [Bikramjot Hanzra’s Texture Matching using Local Binary Patterns](http://hanzratech.in/2015/05/30/local-binary-patterns.html) tutorial. The implementation uses several packages, including OpenCV and the local\_binary\_pattern algorithm in the skimage.feature package. The training dataset used was a collection of four famous paintings, all of which seem to have different textures. Two copies of each image were used in the training dataset in order to test this method’s sensitivity to angle, color intensity, and image size. As a pre-processing step, the images were rescaled to have the same number of pixels and dimensions.

The only user-defined parameter in the LBP algorithm is the radius of the circular patch. Since the purpose of this approach is to detect the fine-grained differences such as the medium of the painting—as opposed to coarser details like how thick lines tend to be in the painting—we iterated over several small values for radius ranging from 3 to 20 pixels. However, no radius value (including much larger ones) seemed to improve or drastically change the results of this technique.

As described in the approach section, in order to classify an input, we calculate its LBP descriptor and then use a distance metric—in this case, chi-squared—to compare the LBP descriptor against the training set. Our experiment returned the chi-squared error as well as the corresponding image. Below are examples of the results.

The subsample of results above clearly show that this method did not work very well, regardless of the query image and the radius used. We believe that the algorithm is doing no better than guessing. For example, in the trial below with radius = 3, we see that one version of *The Scream* has a chi-squared error of 0.064 while the other has an error of 0.515 from our query image.



**Texture**

Ultimately, this method was not able to yield consistent or impressive results. This makes sense given the very nature of paintings: most don’t have any sort of fine-grained, repeatable pattern in them, which is what texture detection depends on finding. Due to these results, we have decided that a texture-based classification approach is not feasible for this classification problem and is thus no longer being considered.