Automated Art Comparison Using Neural Networks
Project Progress

Andrew Ward, Ryan Burke

Introduction

In the Internet Age, the presence of digitized fine art galleries has risen steadily, bringing unique masterpieces to the fingertips of everyone with a network connection. Simultaneously, machine learning has broken new ground in the realm of image classification and object recognition. Some of these techniques have been applied to online art collections in order to provide large-scale classifications of art based on style and genre [1]. However, we believe there are both features and learning techniques that have gone unexplored. Furthermore, our goal is not to classify individual art pieces; the goal of our project is to compare two paintings and determine their similarity. Ultimately, we hope to create an automated method of determining whether any two paintings share the same artist, genre, or style. This type of system can be used to determine artistic influences and relationships, as well as provide a recommendation system based on similarity.

Related Work

Saleh and Elgammal tackled the problem of predicting a painting's artist, genre, and style using a similarly labeled dataset to ours [1]. In this study, they applied metric learning to classify individual pieces of art using features obtained by unsupervised learning, and features that represented which objects were in an image. In a different study, Gatys et al. trained a convolutional neural network to perform a content invariant style transformation on an input image [2]. In other words, the neural network was trained to learn a style from one painting, and then transformed a different image to reflect the style the neural network had learned. This worked remarkably well, which demonstrates the efficacy of neural networks in more abstract feature recognition. Ivanova et al. addressed a similar art classification problem, but did not use machine learning methods [3]. The features they extracted consisted of MPEG 7 descriptors that were quantized in a tiled fashion across the image.

Data

The training and test sets for this project are labeled image datasets from the "Painter by Numbers" Kaggle competition [4]. This includes over 10 gigabytes of image files, ranging from well-known art to paintings submitted by anonymous artists. In addition, we are supplied a metadata file that includes the artist name, painting title, style, genre, and date the work was completed.

Problem

The goal of this project is to create a system that, when given two paintings, is able to determine whether the two paintings were:

1. created by the same artist,
2. painted in the same style,
3. painted of the same genre.

Specifically, the input will be a collection of image files, and a CSV file with image filename pairings. The output will be three binary vector, with each element corresponding to one comparison of two paintings, i.e. one row in the input CSV. Additionally, we hope to return a “similarity” score for each pair, if the model allows. This would aid in a recommendation system and in uncovering relationships between artists, styles, and genres.

Methods

Preprocessing:

We will first preprocess our data set to extract useful model features from the paintings. For this, we will experiment with a variety of digital image processing techniques, including L*a*b* color histogram, GIST, graph-based visual saliency, Meta-class binary features, deep convolutional neural net, content classifiers, HOG2x2, SSIM [5]. Example features include general gradients, prominence of areas with fixation potential, and image texture. Since there is a wide range of possible features to use, we plan to determine the most relevant features using principal component analysis (PCA).

Model:

Because of the clear differences in the categories of artist, style, and genre, we envision our automated system will have a different model that addresses each of our three goals. From the successful work of Gatys et al. [2], we believe that a deep neural network will be a useful model for style.

Baseline:

For our baseline model, we reduced each image to 8 values: the mean and variance of the entire image and RGB channels individually. We then found the absolute sum of the difference between the 8 values for the images. Sums below a threshold (calculated by averaging the mean positive and negative comparisons) are categorized as by the same artist/genre/style. The results are shown in Table 1.

	Genre	*Style*	*Artist*
Sensitivity (TPR)	.618	.518	.538
Specificity (TNR)	.533	.557	.726
Precision (PPV)	.103	.042	.008

Table 1: results of baseline predictor in genre, style, and artist comparison

Oracle:

The oracle for our problem would be an art aficionado with access to a database of all noteworthy art. The oracle will be able to categorize by genre and style with 100% accuracy, but won’t be perfectly accurate at identifying paintings by the same artist, since the database includes art by unknown artists.

Progress

<u>Preprocessing:</u>

L*a*b* color histogram, GIST, SSIM, and HOG have been implemented for this progress report.

L*a*b* color histogram: The L*a*b* color histogram is found by converting an image to the CIELAB color space, and computing histograms for the L, a, and b channels. This data provides more direct insight into the luminance of an image, while maintaining all color information. In implementation, the L*a*b* color histograms are separated into 4, 14, and 14 bins for the L, a, and b histograms respectively. The histograms are also normalized such that they sum to one, to account for different image sizes.

GIST: The GIST algorithm is computed by convolution with a series of filters that extract qualities such as openness, naturalness, and expansion. The standard GIST algorithm is used to collect an additional 512 features for each image.

SSIM (Structural SIMilarity): SSIM uses the means, variances, and covariances of two subsections of a test image and a reference image to compute similarity for that subsection. Similarity of each subsection is computed, resulting in a matrix of similarity scores. This SSIM algorithm is implemented by scaling the test image to the same size as the reference image.

HOG (Histogram of Oriented Gradients): HOG computes histograms of subsections of an image and creates a histogram describing their orientations. These histograms can be used to match subsections of larger images to given templates. HOG is utilized by finding the histograms of both images (regardless of image sizes) and finding the maximum similarity between the two by shifting the overlapping image subsections.

<u>Model:</u>

A simple neural network has been implemented for this progress report. The neural network was implemented in R using the “nnet” package, and contains only one hidden layer, which uses 10 nodes. We plan to expand this to a multiple hidden layer convolutional neural network, which will require much more extensive tuning of the number of nodes in the hidden layers and the decay parameters for backpropagation. Currently, our workflow consists of extracting image features in MATLAB, and passing these features in matrix form into R, where the features and labels are used to train a separate neural network for each of the three categories we are trying to predict. However, due to R’s limit of 100 features in a data frame, we were not able to use all of the features we extracted in the neural network. In fact, we were only able to use 50 features, as we included features from each of the two images we were comparing, and we chose only features from the L*a*b* color histograms due to their simplicity. Moving forward, we plan to include a feature selection phase which ranks variables by importance in order to use only the most useful features to train our models. In addition, because of the 100-feature constraint and the lack of control in parameter tuning in R, we are exploring a Python convolutional neural net implementation.

Preliminary Results

We trained three separate neural networks, one each to classify whether two paintings had the same genre, style, and artist, on features we extracted from the two paintings. The results are shown in Table 2. Promisingly, the simple, untuned neural network model outperforms the baseline mode in every category except for sensitivity in Genre.

	Genre	*Style*	*Artist*
Sensitivity (TPR)	.302	.529	.652
Specificity (TNR)	.853	.815	.828
Precision (PPV)	.153	.216	.188

Table 2: results of rudimentary (untuned) neural network predictor in genre, style, and artist comparison

This decrease in performance in genre likely comes from the fact that, due to the time taken to train the neural network, we used an extremely small subset of the images (155 images out of 79,433), and cross-compared them with each other. Therefore, it is likely that many genres, styles, and artists were poorly represented; there are 43 genres in the entire dataset, and only 28 were represented in our sampling, 10 of which were represented by only one picture. We have planned to allow ample time for network training once the model has been completed.

References

[1] Saleh, B. and Elgammal, A. "Large-scale Classification of Fine-Art Paintings: Learning The Right Metric on The Right Feature." arXiv preprint arXiv:1505.00855v1 (2015)
[2] Gatys, L et al. "A Neural Algorithm of Artistic Style." arXiv preprint arXiv:1508.06576v2 (2015)
[3] Ivanova, K. et al. "Features for Art Painting Classification based on Vector Quantization of MPEG-7 Descriptors." ICDEM 2010. *Data Engineering and Management,* Springer (2010)
[4] URL https://www.kaggle.com/c/painter-by-numbers
[5] Karayev, Sergey, et al. "Recognizing image style." arXiv preprint arXiv:1311.3715 (2013)