Predicting Artist Influence

Adam Baker, David Saadatnezhadi, Mitchell Schmidt, Wesley Siebenthaler

November 2019

Abstract

Our team found a dataset made up of 50 of the most influential artists of all time. We set out to attempt to create a model that learns to predict the artist based off the painting provided. First, six preprocessing steps were done before fitting the data to our models. The deep learning networks that we apply to our data were VGG16 and LeNet. These models were compared by their train accuracy and validation accuracy scores. After running the data through the deep learning algorithms, we discovered the VGG16 model to be the most optimal. The VGG16 model had the highest train accuracy at 0.279 and the highest validation accuracy at 0.280. One of our baseline models was a linear classifier, which had a training accuracy of 0.198 and validation accuracy of 0.174. We acknowledge that these scores are low, but when compared to simple guessing, the accuracy has improved by nearly five times.

1 Introduction

The history of paintings dates back to as early as 40,000 BCE. Since its first inception, paintings have progressed into an art form that has found its way into almost every culture throughout history. We found a dataset that is comprised of 50 of the most influential artists of all time. The dataset was obtained from Kaggle.com **[1]**. It contains 8,355 paintings with its accompanying artist. Our goal was to try and create a model that learns to identify the artists by analyzing the paintings. Through the use of the deep learning networks LeNet and VGG16, we hoped to make these predictions as accurate as possible. The remainder of this paper is dedicated to explaining the six distinct steps of our preprocessing, elaborating more into the models that we used, the results of how well the algorithms were fit to the data, and an analysis on the project as a whole.

2 Preprocessing

There were six preprocessing steps that needed to be completed in order for the data to be properly utilized in our model. First, we had to split the images into 36 parts. The divisor, 36, was chosen to reflect the number of training images to output categories in the MNIST dataset (which had 10 output categories and 60,000 training images). Since we had five times as many output categories, we chose to supply five times as many training images (300,000). We originally had 8,355 images, so we needed 36 training images from each original image (as

36*8355=300,786). This was done using the Image_Splitter Python library. As a part of this step, the resulting images were resized to 32x32x3 using the Python imaging library. The images below display our reasoning to splitting and resizing the original images.

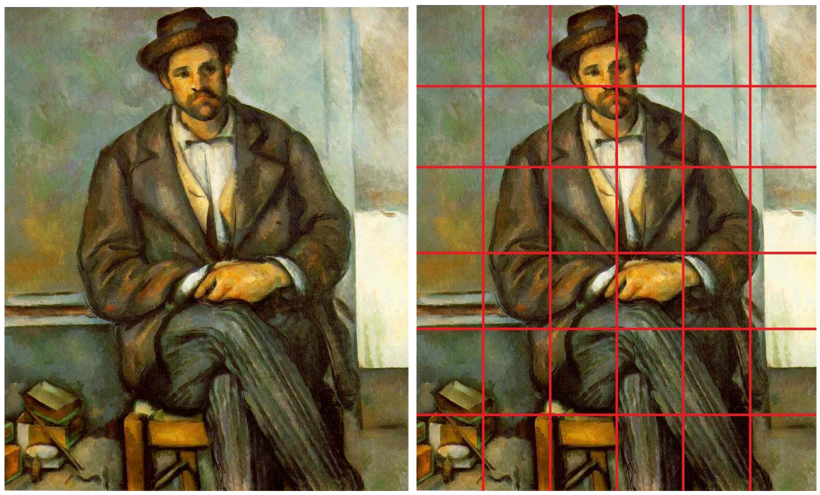

Image 1: Example picture (Paul_Cezanne_6.png) original image and with the 6x6 grid overlay.

Image 2: Subimage (0,2) of the original image and after resizing to 32x32x3.

Next, we needed to convert any grayscale images into RGB. This was done by taking the black and white channel and duplicating into two additional channels **[2]**. This was necessary so all our inputs were 32x32x3, so they could go into NPZ files and the respective networks. The files needed to be converted to NPZ so that the algorithms could correctly read and interpret our data.

We then one-hot encoded all of the images based upon the corresponding artist. Finally, we then shuffled the 300,780 images and then split them up into a training and testing set, using an 80-20 split. These two sets of images were saved as NPZ files, each with a set of 32x32x3 images, one-hot encoded labels, and corresponding label names.

3 Methods

Below are the methods we utilized to train and test to determine the artists' influences.

3.1 Simple Bias Classifier

The simple bias classifier that we implemented was simply random guessing that based on the most common artist in the dataset: Van Gogh. This resulted in an accuracy of 0.063.

3.2 Linear Classifier and Logistic Regression

To set a baseline for analyzing the performance of the other deep learning models, we fit a linear classifier and logistic regression for the training data.

3.3 VGG16

We applied transfer learning to the VGG16 network. We froze the weights on the convolution layers and removed the last three layers. We replaced these last few layers with a flatten layer, a dense layer with 50 outputs, a relu activation function, an additional dense layer with 50 outputs, and then a softmax activation function. There were a total of 28,200 trainable parameters in the final network with an 80-20 train test split.

3.4 LeNet

We trained a classic LeNet network to form a convolutional neural network to predict the artist influence. Our LeNet network has two convolutional layers, two max pooling layers, and finally a fully connected dense layer after the network had been flattened. There were a total of 1,652,120 trainable parameters in the network with a 80-20 train test split.

4 Results

The table below details the optimal hyperparameter(s), training accuracy, and validation accuracy of each technique described in the Methods section.

Table 1: Training and Validation Accuracies of each method

Method	Hyperparameters	Train Accuracy	Validation Accuracy
Simple Bias Classifier	None	N/A	0.063
Linear Classifier	None	0.198	0.174
Logistic Regression	Learning rate = 0.1 Epochs = 10	0.253	0.196
LeNet	Learning rate = 0.001 Batch size = 1000 Epochs = 100	0.273	0.200
VGG16	Learning rate = 0.001 Batch size = 1000 Epochs = 100	0.279	0.280

5 Conclusion

Predicting the influence of an artist on another artist's work is an advanced deep learning problem and with the right methods we were able to get a test accuracy rate of 0.280. The method used to create the best model was our VGG16 network. Before training the VGG16 model, data was preprocessed to form 32x32x3 miniature paintings from the larger, full-sized paintings. The painting images also had to be transformed from PNG files to NPZ files, as we are not equipped to deal with PNG files at this time. Following this preprocessing, we were able to train a VGG16 network with transfer learning to obtain a train accuracy of 0.279 and a validation accuracy of 0.280 after finding the optimal hyperparameters. Finally, from this model, we were able to find the most influential artist to be Van Gogh. However, this may be caused by the skew of the data as Van Gogh's paintings account for approximately 6 percent of the data.

In the future, we would look to adapt the models to include the specific structure of paintings. However, the network is set up to analyze digits rather than brush-strokes. We think that adding more convolutional layers to the model could provide a higher validation accuracy. In addition to this, a rotation of the training data set to increase the amount of images trained on and normalizing the number of images for each artist could improve the performance of the models. Finally, we think that finding different artists' artworks and then adapting our model to analyze the influences that the artist had could be an interesting next step. Deep Learning has very interesting applications to artwork and specifically paintings. Perhaps the next great era of art in human history could eventually be done purely through the use of deep learning and AI.

References

[1] Icaro. "Best Artworks of All Time." *Kaggle*, 2 Mar. 2019, www.kaggle.com/ikarus777/best-artworks-of-all-time.

[2] Arthur, Matthew. "Transform Grayscale Images to RGB Using Python's Matplotlib." *Medium*, Towards Data Science, 11 Feb. 2019, towardsdatascience.com/transform-grayscale-images-to-rgb-using-pythons-matplotlib-6a0625d992dd.