First Capstone Project  
Artist identification from arts

September 7, 2017

# Overview

## Project Background and Description

Artist identification of fine art paintings is a challenging problem primarily handled by art historians with extensive training and expertise. Many previous works have explored this problem by explicitly defining classification features. We train Convolutional Neural Networks (CNNs) with the goal of identifying the artist of a painting as accurately and precisely as possible. Our dataset consists of 300 paintings per artist from 57 well-known artists. We train a variety of models ranging from a simple CNN designed from scratch to a ResNet-18 network with transfer learning. Our best network achieves significantly higher classification accuracy than prior work.

Additionally, we perform multiple experiments to explore and understand the learned representation of our networks. Our results demonstrate that CNNs are a powerful tool for artist identification and that when asked to identify artists, they are able to learn a representation of painting style.

## Project Scope

Artist identification is traditionally performed by art historians and curators who have expertise and familiarity with different artists and styles of art. This is a complex and interesting problem for computers because identifying an artist does not just require object or face detection; artists can paint a wide variety of objects and scenes.

Additionally, many artists from the same time period will have similar styles, and some such as Pablo Picasso have painted in multiple styles and changed their style over time. Previous work has attempted to identify artists by explicitly defining their differentiating characteristics as features. Instead of hand-crafting features, we train CNNs for this problem. This approach is motivated by the hypothesis that every artist has their own unique style of art and that we can improve upon existing artist identification methods by using a CNN to determine the best possible feature representation of paintings.

Our document has two key contributions:

* Train a neural network that significantly outperforms existing approaches for artist identification on a large and varied dataset
* Explore and visualize the learned feature representation for identifying artists

This notebook will purely be an exploratory and hopefully concise enough attempt to explain the idea as well as using different methods to extract meaningful relations out of it.

My stakeholders (Art lovers, Art industries, Computer Vision works working with Arts, Museums and government) will be greatly benefitted with my model and also common people can predict artist by looking at arts.

## High-Level Requirements

In order to train a CNN to identify artists, we first obtain a large dataset of art compiled by Kaggle that is based on the WikiArt dataset [13]. This dataset contains roughly 100,000 paintings by 2,300 artists spanning a variety of time periods and styles. The images vary widely in size and shape.

Every image is labeled with its artist in a separate .csv file. The vast majority of artists in the full dataset have fewer than 50 paintings, so in order to ensure sufficient sample sizes for training networks we use only the artists with 300 or more paintings in the dataset. Therefore, our dataset consists of 300 paintings per artist from 57 artists (about 17,000 total paintings) from a wide variety of styles and time periods.

We split this dataset into training, validation, and test sets using a 80-10-10 split per artist. As a result, the training dataset consists of 240 paintings per artist and the validation and test sets each have 30 paintings per artist. Although some artists have more than 300 paintings in the full dataset, we select an equal number of paintings per artist to ensure a balanced dataset for our experiments.

Because the art in the dataset comes in a variety of shapes and sizes, we modify the images before passing them into our CNNs. First, we zero-center the images and normalize them. Next, we take a 224x224 crop of each input image. While training the network, we randomly horizontally flip the input image with a 50% probability and then take a crop of a random section of the painting. This randomness adds variety to the training data and helps avoid overfit. For the validation and test images, we always take a 224x224 center crop of the image to ensure stable and reproducible results. We do not rescale paintings before taking crops in order to preserve their pixel-level details. Our hypothesis is that artist style is present everywhere in an image and not limited to specific areas, so crops of paintings should still contain enough information for a CNN to determine style.

Also, we hypothesize that in order to determine style, it is important to preserve the minute details that might be lost with rescaling. Given the large number of entries in the dataset and the processing that is needed before passing them into our CNNs, we do not load our entire dataset into memory but store it on disk and load minibatches one at a time. This slows down training due to requiring additional disk read time, but allows us to train using our entire dataset and to use larger crops of our paintings than would be possible otherwise, improving overall accuracy.

We develop and train three different CNN architectures to identify artists. Every network we use takes as input a 3x224x224 RGB image and outputs the scores for each of the 57 artists present in our dataset.

For all of our networks, we use a softmax classifier with cross-entropy loss.

This loss function ensures that our network is constantly trying to maximize the score of the correct artists of its training examples relative to other artists during training.

We train a simple CNN from scratch for artist identification. As the name implies, this network serves as a baseline for comparison with the other approaches. Every layer in the network down samples the image by a factor of two in order to reduce computational complexity, but the downside of this approach is that it might not allow sufficient exploration of lower-level features as the image details are quickly aggregated.

ResNet-18 Trained from Scratch Our next network is based on the ResNet-18 architecture but with a new fully-connected layer to allow for artist predictions. ResNets use residual blocks to ensure that upstream gradients are propagated to lower network layers, aiding in optimization convergence in deep networks [7]. We train this network from scratch to allow the network to learn features solely for the purpose of artist identification We used the 18-layer version of ResNet in order to allow for faster training and to reduce the memory requirements.

ResNet-18 with Transfer Learning Our final network is also based on ResNet-18 but starts with pre-trained weights from the ImageNet dataset. Like the previous network, we replace the final fully-connected layer with a new one to calculate a score for each artist in our dataset instead of a score for ImageNet classes.

## Deliverables

* Clear, documented code that will reproduce the entire analysis, and show what steps were taken in the analysis. Likely in a jupyter notebook format. Longer code could be provided separately and referenced in notebook.
* A slide deck presenting the results and implications of the analysis to the client.
* All deliverables will be in a GitHub repository.

<https://github.com/chatkausik/Second-Capstone-Project>