Artist Identification with Convolutional Neural Networks

Nitin Viswanathan
Stanford University
nviswana@stanford.edu

Abstract

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.

1. Introduction

Artist identification is the task of identifying the artist of a painting given no other information about it. This is an important requirement for cataloguing art, especially as art is increasingly digitized. One of the most comprehensive datasets, Wikiart, has around 150,000 artworks by 2,500 artists [30]. Artsy has a growing collection with the aim of making all art easily available and accessible online [1]. As these collections grow, it becomes increasingly important to be able to efficiently label and identify newly digitized art pieces. A reliable way to identify artists is not only useful for labeling art pieces but also for identifying forgeries, another important art historical problem.

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 (see Figure 1) have painted in multiple styles and changed their style over time.

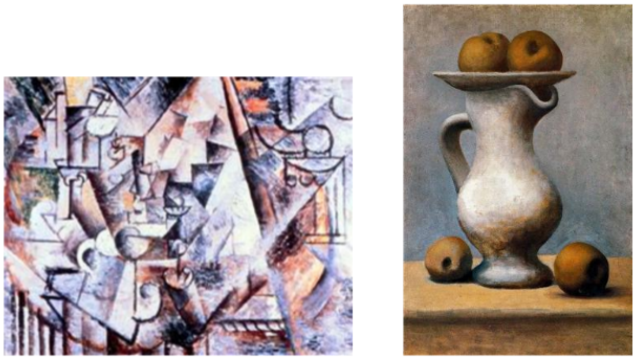

Figure 1. Both of these paintings were created by Pablo Picasso, but they have vastly different styles and content.

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 paper 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

2. Related Work

As mentioned previously, artist identification has primarily been tackled by humans. The Wikiart dataset, consisting of about 150,000 artworks by 2,500 artists, is collected and labeled by volunteers [30]. Artsy's Art Genome Project is led by experts who manually classify art [1].

Most prior attempts to apply machine learning to this problem have been feature-based, aiming to identify what qualities most effectively distinguish artists and styles. Many generic image features have been used, including scale-invariant feature transforms (SIFT), histograms of oriented gradients (HOG), and more [3, 16, 19, 20].

[26], another prior work using generic features, is one of the few works that explicitly tries to identify artists with a large and varied dataset. Most prior work uses datasets with less variety and smaller label spaces; for example, [22] uses art from only one museum and [12] looks at only 5 different artists. Our dataset consists of 17,000 paintings total from 57 different artists and is significantly larger than what most prior work has used.

Previous work has also explored using features that are more specific to art, such as identifying distinct brushstrokes, although this work is primarily done in the context of style identification and not artist identification. For example, [9] develops a new method of automatically extracting brushstrokes by combining edge detection and clustering-based segmentation. It applies these features on a variety of van Gogh paintings to capture his individual style. Other work examines brushstrokes across many different painted portraits to distinguish style [24].

Prior work has primarily used SVMs to identify artist and style given these features. However, other classification methods such as k-nearest neighbors and hierarchical clustering have been experimented with as well [12, 19].

Recently, CNNs have made great advances on many image recognition tasks, for example achieving a top-5 classification error of 3.6% on the ImageNet dataset [7]. This accuracy is higher than than human performance on the same dataset [25]. When applied to art, CNNs have been successfully used to decompose a painting into style and content components and to transfer style from one painting to another, implying that a neural network is capable of capturing the style of paintings [6, 11]. However, this work does not explicitly identify or label the style or artist. Some previous work has applied CNNs to the problem of style identification and shown promising results [14]. Other work has generated features using neural networks and then used SVMs as the classifier to identify style [2, 27]. There has not been notable exploration using CNNs for artist identification.

3. Dataset

3.1. Overview

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. Figures 1 and 2 have some sample images from this dataset. The images vary widely in size and shape. Every image is labeled with its artist in a separate .csv file.

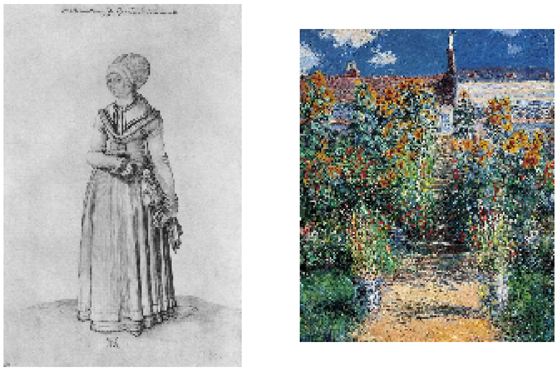

Figure 2. Work by Albrecht Durer on the left and Claude Monet on the right.

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. This dataset is significantly larger than those used in prior work.

3.2. Preprocessing and Data Augmentation

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

Input size	Layer
3x224x224	3x3 CONV, stride 2, padding 1
32x112x112	2x2 Maxpool
32x56x56	3x3 CONV, stride 2, padding 1
32x28x28	2x2 Maxpool
1x6272	Fully-connected
1x228	Fully-connected

Table 1. Baseline CNN architecture (ReLU and batch normalization layers omitted for readability).

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.

4. Methods

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:

$$L_i = -\log(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}})$$

Where L_i is the loss for example i in the training minibatch, f is the score for a particular class calculated by the network, j is one of the possible classes (of the 57 artists), and y_i is the correct class for example i. 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.

4.1. Baseline CNN

We train a simple CNN from scratch for artist identification. Table 1 shows the architecture of this network. As the name implies, this network serves as a baseline for comparison with the other approaches. Every layer in the network downsamples 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.

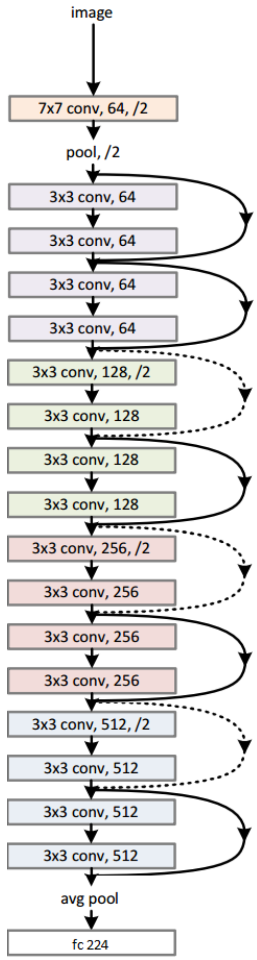

Figure 3. Resnet-18 with a fully-connected layer for artist identification, based on [7].

4.2. 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. Our network architecture is visible in Figure 3. We used the 18-layer version of ResNet in order to allow for faster training and to reduce the memory requirements.

4.3. 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.

We start with a pre-trained network to test whether or not a feature representation from ImageNet is a valuable starting point for artist identification. Some artists, for example Renaissance painters, used shapes and objects that you would expect to find in ImageNet since they usually painted lifelike scenes. However, other artists such as Cubists did not paint scenes as directly representative of the real world.

5. Experiments

5.1. Setup

All of our models and experiments are implemented in PyTorch [23]. We used [29] to set up the ResNet-18 architecture and to obtain weights for ResNet-18 pre-trained on ImageNet. We used [5] as a guide for replacing the fully-connected layer of ResNet-18 with one suitable for artist identification, and for performing transfer learning on the pre-trained ResNet network. All experiments were performed on Google Cloud using a machine with 8 vCPUs and an NVIDIA Tesla K80 GPU, and 80GB storage capacity.

5.2. Implementation Details

We trained all of our models using an Adam update rule [15]. We explored using SGD with momentum, but obtained better results with Adam. For the two networks trained from scratch, we started with the default Adam parameters of learning rate $= 10^{-3}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. We observed the accuracy and loss for both the training and validation datasets over the training epochs and decreased the learning rate by a factor of 10 if improvement slows down significantly. We initialized the weights of the convolutional layers of our networks based on [8], as the methodology from this work is best practice when initializing networks that use a recitified linear activation function.

For training the ResNet with transfer learning, we first held the weights of the base ResNet constant and updated only the fully-connected layer for a few epochs. We performed this step using the same default Adam parameters described previously. After network performance stopped improving, we allowed weights throughout the entire network to change but lower the learning rate to 10^{-4}. This allows for some change throughout the entire network to better fit our dataset.

We experimented with varying levels of L2 regularization on all networks but did not see significant changes on validation set performance so in the end we used no regularization during training.

5.3. Evaluation Metrics

Using the scores generated by our networks, we report top-1 classification accuracy (the fraction of paintings whose artists are identified correctly), precision, recall, and F1 scores. We also report top-3 classification accuracy, which considers a painting correctly classified if the correct artist is in the top 3 highest scores generated by a network. We compare our networks against each other as well as against [26], which reports SVM classification accuracy using pre-defined features.

Precision and recall are defined as:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

The F1 score, a weighted average of precision and recall, is defined as:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

In addition to quantifying the accuracy of our networks, we conduct a variety of experiments to qualitatively evaluate their performance and to understand their learned representations. We use a variety of techniques including saliency maps, gradient ascent for class score maximization, and filter visualizations [18].

6. Results and Discussion

6.1. Quantitative Analysis

Table 2 compares the accuracy of the different CNNs across our key metrics. We can see that both networks that were trained from scratch do not perform as well as the SVM approach from prior work and that the network that started out pre-trained on ImageNet performs significantly better, with a 20% greater absolute (not relative) Top-1 training accuracy. When looking at Top-3 classification accuracy, we see that all networks perform better than their Top-1 accuracies. Our best network achieves a Top-3 classification accuracy of almost 90%, meaning that it can narrow down the artist to three in the vast majority of cases. Of the networks trained from scratch, ResNet-18 outperforms the baseline CNN by a large margin, indicating that increasing network depth does help improve accuracy. Network performance as ranked by test set accuracy is consistent with performance on the other reported metrics as well. Although there is variation in the F1, precision, and recall scores, they tend to track closely with classification accuracy.

Figure 5 shows how the training and validation accuracies of our networks changed during training. We report only training and validation accuracy; the trends we describe in this section also occurred in the loss function as well but are not shown here to allow for clearer images. We see that in the networks trained from scratch, training and validation accuracies track very closely together, indicating

Model	Top-1					Top-3	
	Train Acc	Test Acc	F1	Precision	Recall	Train Acc	Test Acc
Baseline SVM [26]	∗	0.578	∗	∗	∗	∗	∗
Baseline CNN	0.437	0.422	0.360	0.416	0.366	0.653	0.623
ResNet from scratch	0.516	0.511	0.503	0.516	0.511	0.737	0.710
ResNet transfer learning	**0.907**	**0.777**	**0.771**	**0.777**	**0.774**	**0.973**	**0.898**

Table 2. Artist identification results summary. ∗ = results not reported.

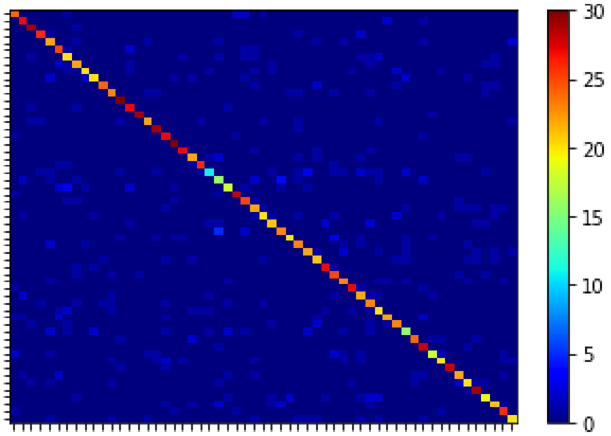

Figure 4. Confusion matrix for top-1 classification accuracy on the test dataset.

little overfit. This could mean that performance of these networks would improve with more training epochs and more training data as we generally expect to see a small but noticeable gap between training and validation/test accuracy in a fully-trained network. The pre-trained network does have a gap between training and test accuracy, indicating that adding more training data and further training might not add as much value.

Despite many hyperparameter tweaks as learning slowed over time, the baseline CNN improved relatively slowly and peaked at round 40% classification accuracy. ResNet-18 when trained from scratch however, shows clear and definite improvements throughout the training process. Each time that learning slowed and the learning rate was decreased by a factor of 10, we saw a clear and immediate bump in accuracy. This happened multiple times, which is why there are multiple plateaus and upward jumps following them in the accuracy graphs. In the ResNet-18 with transfer learning, we see one notable jump after epoch 5, and a smaller jump at epoch 11. The first jump at epoch 5 corresponds to not only a learning rate decrease but also to allowing the entire network to change instead of only the final fully-connected layer. We see a significant increase in training and validation accuracy at this pint, indicating that the network begins adapting much better to an art dataset. The second jump around epoch 11 corresponds only to lower the learning rate, and there is a noticeable but smaller improvement here as well.

We can see that although not explicitly given a feature representation of paintings, our best network was able develop one that significantly outperforms existing approaches. These results not only show the overall potential of CNNs in artist identification but also the value of starting with an ImageNet-based feature representation.

We next examine the artists that our best network had more trouble classifying correctly. Figure 4 shows a confusion matrix, which was calculated using Torchnet for PyTorch [28]. Each row represents the true artist and each column represents a predicted artist; we omit the labels on the matrix for clarity since there are 57 artists in our dataset. Ideally, we want as many predictions as possible to be concentrated on the diagonal as this means that the network correctly predicted the true artist. The maximum possible value for a cell in the confusion matrix is 30, since there are 30 paintings per artist in the dataset.

We can see that for most artists, the diagonal entry is yellow or red, indicating that most paintings are classified correctly. One artist in the middle, Henri Matisse, is light blue; only 11 of his 30 paintings in the test dataset were correctly attributed to him. It does not appear that Matisse is being strongly confused for any other artist in particular, as there are not other artists in his row with bright colors. Instead, his paintings have been attributed to a wide variety of artists. Matisse is most confused for Martiros Saryan; the network predicted Saryan as the artist for 3 of Matisse's paintings. This is not an entirely unexpected result, as Saryan's style was heavily influenced by Matisse [4]. Additionally, Matisse experimented and painted in a wide variety of styles, ranging from Fauvism to Post-Impressionsim to Modernism, increasing his overlap with other artists [21]. These results point towards our network building a representation of artistic style and having trouble if an artist has a wide variety of styles and influenced others.

6.2. Qualitative Analysis

We now explore a variety of visualizations on our networks to better understand their representations. With the exception of first-layer filter visualizations, all of our experiments were conducted on our best-performing network

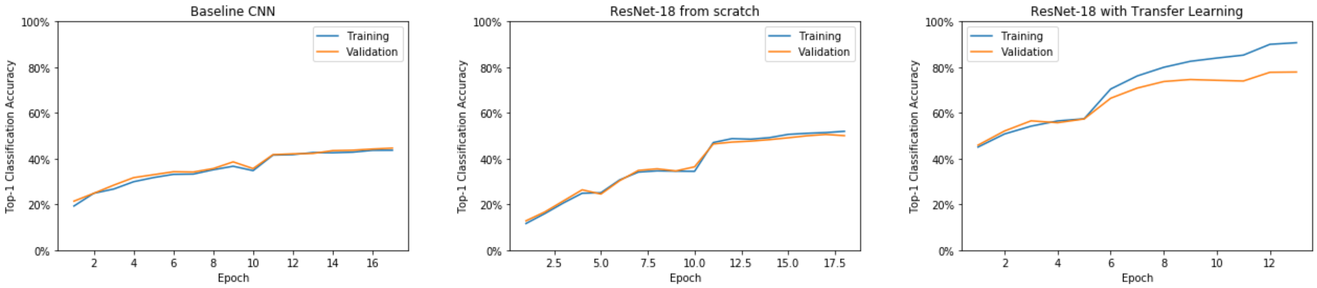

Figure 5. Accuracy by epoch when training our three different networks.

based on ResNet-18 with transfer learning.

Saliency maps allow us to visualize which pixels in an image contribute most to the predicted score for that image [18]. We implement saliency maps for artist identification to determine what parts of paintings are used to predict artists [17]. In Figure 6, the painting on the right has a fairly well-defined saliency map focused around the person, including their face and arms. However, in the painting on the left, the saliency map highlights pixels from all over the image, not focusing on the person. This indicates that in some paintings, objects or people are where an artist's style is most apparent, while in other paintings, the style comes from all over. We examined saliency maps of many other paintings and saw that in most but not all, the important pixels were spread all over the image and not focused around objects or people in them. This indicates that the network is flexible in either focusing on certain parts of paintings where style is particularly evident, or looking everywhere if there is not one distinctive area or object in the image.

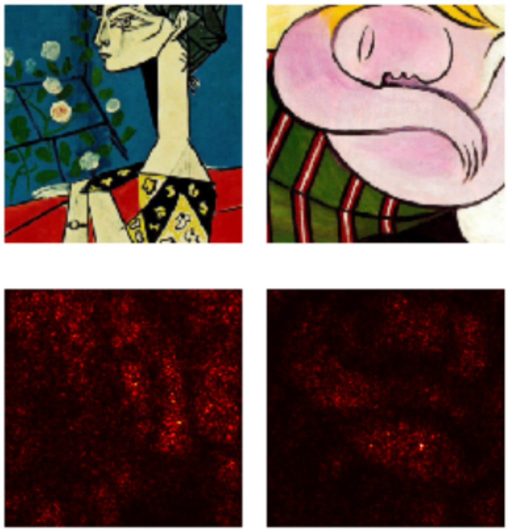

Figure 6. Two paintings by Picasso and their corresponding saliency maps.

To better understand our models, we also examine their filters to understand what might activate them. Figure 7 shows the filters in the first convolutional layer of ResNet-18 trained from scratch. Some of these filters appear to be splotches of color, making them more useful for understanding the colors of different artists. Others appear to be more shape-focused, helping the network identify the rough strokes and shapes of paintings. We omit the filters of the network based on pre-trained ResNet-18 as it's first-layer filters appear largely unchanged from the pre-trained version. These filters are also a mix of shape-focused and color-focused, further providing evidence that starting with ImageNet data is helpful for predicting artists as these filters align closely with those of the network trained from scratch.

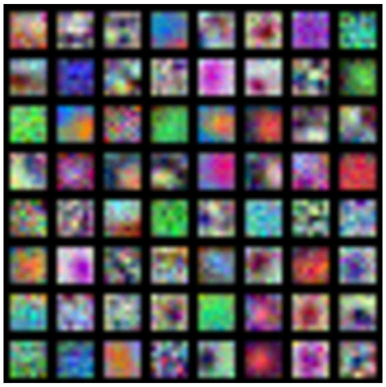

Figure 7. Filters from the convolutional first layer of ResNet trained from scratch.

We next try to understand the representation of the subsequent layers in our networks. We apply gradient ascent on our best network to create an image that maximizes the score of a particular artist [17]. This helps us understand what it views as the style of specific artists. Figure 8 shows two images that maximize the scores for Vincent van Gogh (left) and Pablo Picasso (right). Picasso's image has much bolder colors and sharper lines than van Gogh's, which makes sense given the difference between Picasso's more abstract Cubist style and van Gogh's softer strokes and post-

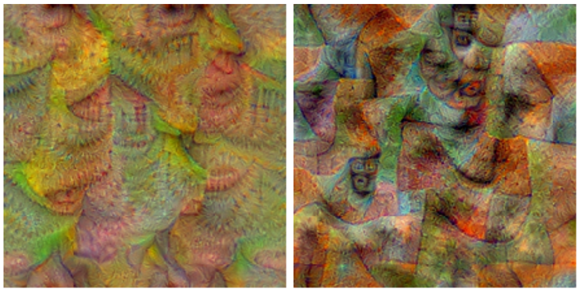

Figure 8. Class maximization images for van Gogh (left) and Picasso (right).

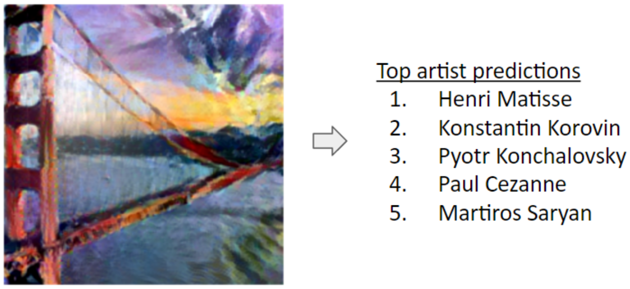

Figure 9. Predicted artists for an image of the Golden Gate Bridge with style transfer from Henri Matisse applied.

Impressionist style. We also see a few circle-like shapes appear in Picasso's image, which could be a commonly recurring structure in his paintings.

Our final experiment to understand the style representation stored by our network is to take style-transferred versions of regular images and run them through our network to obtain a predicted artist. Figure 9 shows an image of the Golden Gate bridge transformed to have the style of Henri Matisse from [10] and indicates the top 5 predictions for its artist. The network predicts Matisse as the most likely artist for this painting, indicating that the network is able to recognize Matisse's style independent of the specific content of the painting. We applied this method to evaluate style transfer from other artists as well and although the network does not always produce the highest score for the correct artist, the correct artist is usually in the top 5, further confirming this theory.

7. Conclusion

We introduce the problem of artist identification and apply a variety of CNN architectures to it to maximize classification accuracy, which has not been done in prior work. Our dataset consists of 300 paintings per artist for 57 artists over a wide variety of styles and time periods. Our best network, based on ResNet-18 pre-trained on ImageNet with transfer learning, outperforms traditional feature-based approaches by a significant margin. This network also outperforms training CNNs from scratch, indicating that features learned from the ImageNet training data are very relevant for artist identification as well. When asked to predict artist, our network created a representation of the style of paintings. We verify this through a variety of experiments looking at the underlying representation and how the network makes artist predictions.

For future work, we would like to dive deeper into the model representations and try to quantify how much of the predictions come from the style of an image versus the content. Our belief is that the networks create a representation primarily of the style of a artists, but there are likely some elements of content present as well - for example, Mary Cassatt almost exclusively painted young women, so the network might have learned a representation of the content of her paintings to help classify them. In order to do this, we plan to calculate the Gram matrices of various layers of our network (to use as a representation of style) and pass them into a separate CNN to obtain artist predictions. This network would still produce one score per artist, but instead of looking at the entire image as an input, it only looks at the information representing image style. We would then compare predictions from this new neural network with our best-performing network to understand how much of the artist predictions comes from the style versus other aspects of a painting like the content.

We would also like to expand our dataset and see how our network handles classifying more artists with fewer paintings for artist. We used 57 artists with more than 300 paintings, but our original dataset has 108 artists with more than 200 paintings. We would switch to using all available images for the artists we use in our dataset instead of using an equal number per artist. This would result in an unbalanced dataset, but if we expand to using more artists, we should still be able to take advantage of the larger sample sizes for certain artists and classify them with high accuracy.

References

[1] Artsy. https://www.artsy.net/about/the-art-genome-project.

[2] W. L. Bar Y., Levy N. Classification of artistic styles using binarized features derived from a deep neural network. *ECCV 2014*.

[3] Z. A. M. X. Bosch, A. Image classification using rois and multiple kernel learning, 2008.

[4] E. Brittanica. https://www.britannica.com/biography/Martiros-Saryan.

[5] S. Chilamkurthy. http://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html.

[6] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint*, abs/1512.03385, 2015.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *arXiv preprint*, arXiv:1502.01852, 2015.

[9] E. H. J. Li, L. Yao and J. Z. Wang. Rhythmic brushstrokes distinguish van gogh from his contemporaries: Findings via automated brushstroke extractions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012.

[10] J. Johnson. neural-style. `https://github.com/jcjohnson/neural-style`, 2015.

[11] J. Johnson, A. Alahi, and F. Li. Perceptual losses for real-time style transfer and super-resolution.

[12] J. Jou and S. Agrawal. Artist identification for renaissance paintings.

[13] Kaggle. https://www.kaggle.com/c/painter-by-numbers.

[14] S. Karayev, A. Hertzmann, H. Winnemoeller, A. Agarwala, and T. Darrell. Recognizing image style. *CoRR*, abs/1311.3715, 2013.

[15] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint*, arXiv:1412.6980, 2014.

[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, 2006.

[17] F.-F. Li, J. Johnson, and S. Yeung. Cs231n assignment 3, 2017. http://cs231n.github.io/assignments2017/assignment3.

[18] F.-F. Li, J. Johnson, and S. Yeung. Cs231n lecture 11, 2017. http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture12.pdf.

[19] T. E. Lombardi. The classification of style in fine-art painting. *ETD Collection for Pace University*, 2005.

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints., 2004.

[21] T. M. M. o. A. Magdalena Dabrowski, 2004. http://www.metmuseum.org/toah/hd/mati/hd_mati.htm.

[22] T. Mensink and J. van Gemert. The rijksmuseum challenge: Museum-centered visual recognition. 2014.

[23] PyTorch. https://github.com/pytorch.

[24] P. K. R. Sablatnig and E. Zolda. Hierarchical classification of paintings using face and brush stroke models. 1998.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li. Imagenet large scale visual recognition challenge. *arXiv preprint*, abs/1409.0575, 2005.

[26] B. Saleh and A. M. Elgammal. Large-scale classification of fine-art paintings: Learning the right metric on the right feature. *CoRR*, abs/1505.00855, 2015.

[27] A. H. S. J. C. S. Sharif Razavian, A. Cnn features off-the-shelf: An astounding baseline for recognition, 2014.

[28] TorchNet. https://github.com/pytorch/tnt.

[29] Torchvision. https://github.com/pytorch/vision.

[30] WikiArt. https://www.wikiart.org/en/about.