# Using CNN to Classify and Understand Artists from the Rijksmuseum

Tara Balakrishan* taragb@stanford.edu
Sarah Rosston* srosston@stanford.edu
Emily Tang* emjtang@stanford.edu

## Abstract

*As art museums become increasingly digitized, tools that can automatically classify artists based on paintings are needed to help both museum curators and visitors. As part of this push for digitization, the Rijksmuseum in Amsterdam has provided a public dataset consisting of art photographs and their corresponding artist labels. Previous work on this dataset includes the PigeoNET CNN, which achieves a 78% test accuracy for correctly classifying artists. In this paper, we use transfer learning by fine-tuning ResNet and VGG models that have been pre-trained on ImageNet, and demonstrate that we can achieve a classification accuracy of over 90% with our model. Specifically, when predicting the 10 most prolific artists in our dataset, we achieve a 90.75% test accuracy, and when predicting the 20 artists with the most images in our dataset, we achieve a 87.7% accuracy. In addition, we provide a thorough analysis of artist styles and visualize what our network learns about the images, through saliency maps and occlusion heat maps. Our goal is to understand why the network can predict artist names with such high accuracy, and learn what parts of each piece of art are most predictive of a particular artist.* [1] [2]

## 1. Introduction

The online presence of artwork has become increasingly important in the twenty-first century so it is important for art collectors, educators, and students alike to be able to view and analyze these important pieces of human artistic history. To support this movement, art museums including the Rijksmuseum are digitizing their collections to make them available for public consumption. However, museums boast hundreds of thousands of paintings and pieces of artwork, and digitization is not an easy task so a classifier that can predict the artist of a painting will be increasingly useful, both for museum curators and curious painting viewers. We hope that building an artist classifier will help curators label art objects automatically and allow visitors to more freely browse artwork. For example, a classifier could assist visitors in querying for art information with just a photo of the artwork, and potentially help identify cases of forgery.

Figure 1. Clustering of Paintings by Artist [18]

### 1.1. Dataset

As part of this push for digitizing art museums, the Rijksmuseum in Amsterdam has provided a public dataset consisting of 112,039 photographs of artwork [18] (shown in Figure 1). The dataset is provided as part of The Rijksmuseum Challenge, which consists of four visual recognition challenges: predicting artist, creation year, type and material. We will focus on the first challenge, predicting the artist based on the art photograph.

The Rijkmuseum dataset contains artwork by 12,641 unique artists, 21 of whom have over 1000 pieces in the collection, including both paintings and sketches (shown in Figure 2). It is important to note that 3,949 artists only have one piece of artwork in the dataset.

### 1.2. Problem Statement

The input to our algorithm is a photo of a piece of artwork. We then use a retrained ResNet CNN classifier[10] to output a prediction for which artist created the input artwork.

As our dataset contains an extremely long tail of artists with few or only one painting, we primarily focused on classifying artwork by the artists who have the most art in our dataset. We began with the 10 artists who created the most

[1] *equal coauthors

[2] Project Source Code: https://github.com/emjtang/geweldig

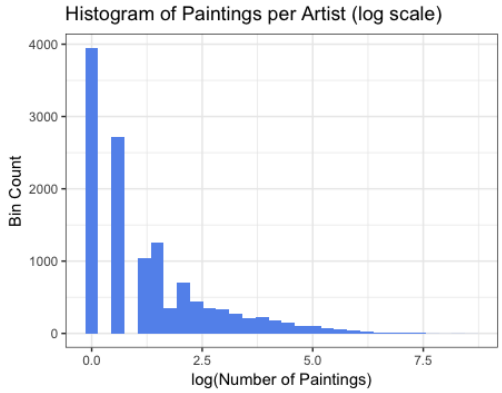



Figure 2. Histogram of Paintings per Artist


Figure 3. Ten Artists with the most artwork in the dataset, log scale

artwork currently held in the Rijksmuseum for the majority of our analysis. The ten artists in this group are George Hendrik Breitner, Jan Luyken, Reiner Vinkeles, Marius Bauer, Isaac Israels, Bernard Picart, Rembrandt Harmensz. Van Rijn, Johannes Tavenraat, Willem Witsen, and Simon Fokke (see Figure 3 for examples of artwork from each artist). We also compare our performance on a 20-class classification problem, to understand the drop in accuracy as we increase the number of classes.

## 2. Related Work

Initial attempts at analyzing artwork with computer vision often focus on feature extraction. This includes Johnson Jr., et al. (2008)[11] who use wavelet transforms to analyze brushstrokes and measure similarity using texture features to identify forged Van Gogh paintings. Using feature selection and a maximum likelihood classifier, they are able to correctly identify the forged paints. Agarwal et al. (2015) use SIFT, HoG, GIST, GCLM, and Color features for genre classification [3] and Cetinic and Grgic (2013) achieve a 75% accuracy on 20-class problem with light, color, texture, and composition features [8]. Saleh and Elgammal (2015) [20] also use data from Wikiart paintings to classify images by genre, style, and artist using a combination of what they call of low-level GIST features and high level CNN features and then use a feature vector of high and low level features for classification.

Khan et al. (2014)[15] also discuss Histogram of Gradients (HOG), SIFT, and GIST features for artist classification. Another approach based on feature extraction is using shape and color features by Saleh et al. (2008) [16] which achieves 62% accuracy on an eight-class classification problem. Arora and Elgammal (2012) [4] also use feature extraction with discriminative and generative models using Bag of Words with OSIFT and CSIFT and semantic features. In a more recent paper looking at classification by artist, van Noord and Postma (2015) [23] discuss the impact of resizing images as it may result in potential loss of information. They find that different image sizes are optimal depending on the artist.

We then turned to references which attempted any components of the Rijksmuseum challenge. This included a paper by van Noord, et al (2015) [23] which discusses pigeoNET, a CNN based on AlexNet that classifies Rijksmuseum paintings by artists if the artist has at least 256 images in the collection. PigeoNET reports accuracies across different subsets of the data ranging from 52% to 78% accuracy. They find that there is little distinction between accuracy on paintings versus prints and that using more images per artist substantially improves accuracy. Based on these findings, we plan to use over 800 images per artist for our training set, which we hope results in increased accuracy.

Karayev, et al (2013)[14] approach a different but related problem in which they classify artwork from Flickr by style (eg. impressionism) and find that a CNN pretrained for ImageNet outperformed feature selection. Their model is eight layers deep, however the paper does not specify what specific layers are used. They achieve 57% accuracy on this Flickr dataset, but the crucial finding that we employ is that models pre-trained on ImageNet learn underlying image features which ultimately improves accuracy. There are other attempts at more sophisticated models, such as Levy et. al's [17] attempt to use genetic algorithms in conjunction with deep learning. Yet, the ability of pure neural networks to predict artist make these advancements unnecessary for our project.

While our project's focus is not on style classification, we were interested in learning about the trade-off between using image features vs neural networks. We looked at a paper by Bar et al. (2014) which uses a combination of CNNs and feature extraction for style classification.[5]. Additionally, Westlake, et al. (2016) [24] discuss transfer learning, finding that only the first few layers of a pretrained CNN are useful for retraining a CNN for person-detection. However, since that is a more specific problem than artist recognition our hope is that transfer learning will be superior to creating our own network. We read over papers describing models that were used for the ImageNet challenge. While the VGG-16 [21] model isn't as deep as many of the most recent networks built for the ImageNet challenge, it seems

like a good starting point for implementing transfer learning due to the ease of retraining the last three fully connected layers. We also looked into ResNet [10] which we believe will perform better than a VGG model on the Rijksmuseum Challenge, given that the network is deeper and presumably learns more image features.

Our contribution to existing work for this challenge space will then be to train a CNN for this classification task using transfer learning from an ResNet model, using convolutional neural networks instead of feature extraction and to build an understanding of the model through visualization and reclassification of images using style transfer.

# 3. Technical Approach

For our baseline model, we extract SURF features from the images and build a Linear SVM. To improve on this baseline, we build two CNN models using transfer learning. Transfer learning is effective because our dataset is quite small. We use pretrained ResNet-18 and VGG-16 models on ImageNet, and fine-tune the last layer of ResNet-18, and the last three layers of VGG-16. We then understand our best model, fine-tuned ResNet, through many visualizations: occlusion heat maps, saliency maps, and the first convolutional layer filter.

## 3.1. Linear SVM Baseline

We download the 1000 images in our dataset and extract visual features using the standard bag of words model in computer vision. We randomly sample 100 images to create a dictionary of 1000 words. To create the dictionary, we extract Speeded Up Robust Features (SURF) images descriptors from the 100 images using OpenCV. SURF is a performant scale- and rotation-invariant interest point detector and descriptor used to find similarities between images [6]. We use K-means clustering to form 1000 clusters, or visual words for the dictionary and then create feature vectors for each image by extracting SURF descriptors from the image. For each descriptor, we select the closest cluster in the dictionary. In brief, the visual feature vector contains the frequency of each visual word in the listing image, and is normalized. We then split our data into training and test (80/20), and train our linear SVM model.

## 3.2. Convolutional Neural Network (CNN) Approach

### 3.2.1 Data Preprocessing

Each painting or sketch was initially downloaded and stored as a picture with dimensions that mimicked the size of its canvas. We preprocess each image depending on the model, VGG or ResNet. We use a modified version of the standard preprocessing for VGG, taking a random 224x224 crop of the input image and subtracting the mean without horizontally flipping the image [21]. We remove the horizontal flip because a painting might not necessarily still reflect the same artist if flipped, unlike animals that could be seen from all perspectives. For our fine-tuned ResNet model, we apply a similar preprocessing, taking a random crop and then normalizing by the mean and variance of each color channel on ImageNet dataset.

### 3.2.2 Transfer Learning

Since our dataset is significantly smaller than ImageNet and our SVM baseline performance is quite poor, we turn to transfer learning to see if we can achieve a higher classification accuracy by fine-tuning a model that is already trained on ImageNet and performant[13][19]. We apply transfer learning to both VGG and ResNet, to see which performs better.

For our VGG model, we fine-tune the VGG-16 network [21]. The key feature of VGG is to use small filters (usually 3x3) and very deep networks. The architecture contains 16 layers, of 3x3 CONV layers followed by 2x2 POOLING layers, where the last three layers are fully-connected (FC) layers [21]. We download weights trained on ImageNet from the VGG-16 checkpoint, and fine-tune the last three fully-connected layers on our new data. We compare the accuracies of fine-tuning just the last fully-connected layers, with the last three layers.

For our ResNet model, we fine-tune the ResNet-18 network [10]. The residual network uses batch normalization and skip connections [10]. Compared to VGG, it has far fewer parameters because there are no fully-connected layers at the end, so the model is easier to optimize. Similar to the approach for VGG, we download weights trained on ImageNet from the ResNet-18 checkpoint, and fine-tune just the last layer on our new data. We compare the performance of ResNet and VGG, and use the higher performing ResNet model for understanding and visualizing the network.

## 3.3. Understanding and Visualizing Our Networks

To better understand the styles of each artist, we wanted visualize how our network works using the methods described below.[1][2]

### 3.3.1 Occlusion Heat Map

We use occlusion on our input images to build heat maps of the importance of different parts of the image. We also compute and visualize the saliency of the painting with saliency maps, as well as the first convolutional layer of our network.


We apply the sliding window technique with a black box that occludes parts of the test image. We run the occluded image through our fine-tuned ResNet model to compute the probability of the artist class, and visualize the importance of each pixel in predicting the correct class with heat maps[22].

#### 3.3.2 Saliency Maps

To compute saliency, we first compute the gradient of the correct class score with respect to the input image, and then take the maximum value over the RGB channels of the computed gradient. The equation used to computed the saliency map $M$, is shown in Equation 1 where $w$ is the derivative with respect to the correct class, $c$ is the color channelm $i, j$ are the location within the images, and $h(i, j, c)$ is the index of $w$ corresponding to $i, j, c$ in the original image.[25]

$$M_{ij} = \max_c |w_{h(i,j,c)}| \tag{1}$$

We then visualize saliency with a heat map.

#### 3.3.3 Style Transfer Experiment

How unique is an artist's style, and is it more important to classification than image content? To answer this question, we conduct an experiment using style transfer. Style transfer combines two image's content and style into image using layers of the VGG-19 model [9]. We apply the implementation of the algorithm provided by Johnson [12], and transfer styles from one artist's image to another artist's image content. Style transfer works by minimizing a style loss with respect to the styling image and a content loss with respect to a content image[9]. We then run this new input image through our fine-tuned ResNet model, to see what artist gets predicted. Through this approach, we hope to better understand whether the style or the content of the artist is more important to our network. For instance, if Rembrandt's style is transferred to another artist's image, will our model still predict the new image to be Rembrandt?

## 4. Experiments

### 4.1. Model Selection

Our baseline classifier was a linear SVM using a visual bag of words model. By tuning the number of visual words in the Bag of Words model, and saw that when we use 50 visual words, we get the highest training accuracy of 0.4475 (where chance is 0.1, since there are 10 total classes) and our test accuracy was 0.39 on our test set.
After establishing a baseline accuracy, we substantially outperformed our baseline with CNNs and transfer learning. We retrained up to the last three fully connected layers of VGG-16 and the last layer of ResNet-18, and achieved the highest validation accuracy with ResNet-18.

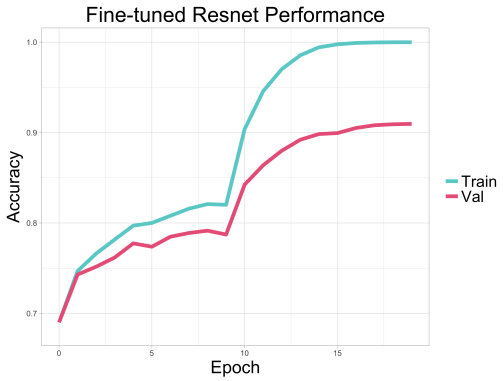



Figure 4. Resnet Accuracy by Epoch

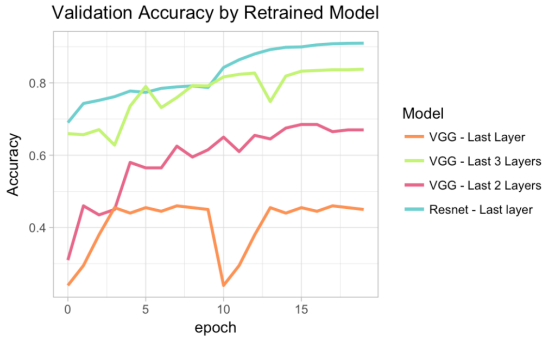



Figure 5. Validation Accuracy of Retrained Models by Epoch

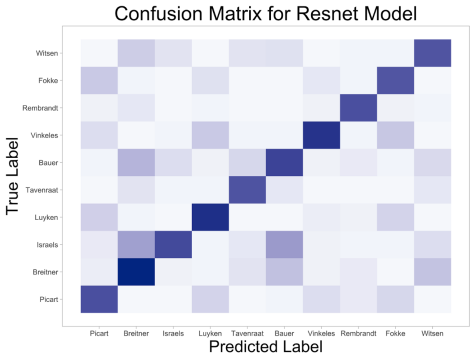



Figure 6. Confusion Matrix for Test Set

The ResNet learning process is shown in Figure 4, where the first 10 epochs are only retraining the weights in the last layer and the next 10 epochs optimize the weights for all

| Artist | Images | Precision | Recall | F1 |
|---|---|---|---|---|
| Jan Luyken | 4452 | 0.957 | 0.945 | 0.475 |
| Reinier Vinkeles | 3816 | 0.935 | 0.961 | 0.474 |
| Rembrandt | 1797 | 0.951 | 0.933 | 0.471 |
| George Breitner | 5403 | 0.939 | 0.912 | 0.463 |
| Johannes Tavenraat | 1592 | 0.919 | 0.880 | 0.450 |
| Marius Bauer | 2981 | 0.869 | 0.863 | 0.433 |
| Bernard Picart | 1937 | 0.880 | 0.850 | 0.432 |
| Isaac Israels | 2498 | 0.795 | 0.945 | 0.432 |
| Willem Witsen | 1519 | 0.852 | 0.833 | 0.421 |
| Simon Fokke | 1472 | 0.865 | 0.814 | 0.419 |

Table 1. Accuracy By Artist

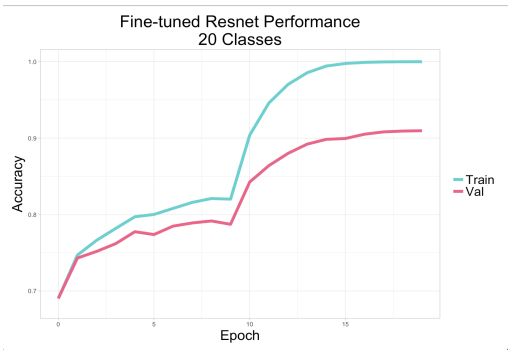



Figure 7. Resnet Accuracy with 20 Classes

layers of the network.

As shown in Figure 5, the fine-tuned ResNet model outperform all of the fine-tuned VGG models and the best VGG model was the one with the most retrained layers, which makes sense given that the images and classes in the Rijksmuseum dataset are very different from the ImageNet classes and images that the pretrained weights were optimized for.

Our final ResNet-18 model has a train accuracy of 0.9999, a validation accuracy of 0.9096, and a test accuracy of 0.9075. The precision, recall, and F1 scores by artist are shown in order of F1 score in Table 1. These accuracy numbers are interesting in that high F1 score seems to be more correlated with a distinctive style than a a higher number of images, for example, Rembrandt has far fewer images in the dataset than Breitner, but has a more distinctive style and the model has a higher precision, recall, and F1 score for Rembrandt images, which. Breitner and Israels similar styles and are confused with each other, as shown in the confusion matrix in Figure 6, while Rembrandt is not commonly confused with any other artist.

We also trained the a ResNet model on a larger dataset including artwork by the 20 most represented artists in the dataset, which had a train accuracy of 1.0, a validation accuracy of 0.888, and a test accuracy of 0.877. The training and validation accuracy by training epoch is shown in Figure 7 where the first 10 epochs train only the last layer of Resnet and the second 10 train the all layers.

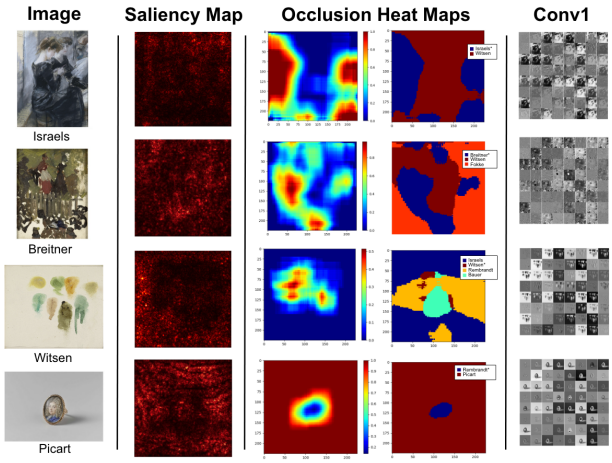



Figure 8. Visualizing Our Network on Select Input Images. In the occlusion heat maps, *for the first three images (Israels, Breitner, and Witsen), the dark blue areas indicate that occluding those area causes the model to classify incorrectly, where as the red areas indicate that occluding those areas does not affect model accuracy. For Picart, the scale is the opposite; red areas affect model accuracy, whereas blue areas do not.*

## 4.2. Model Understanding

### 4.2.1 Occlusion Heat Map and Saliency Maps

We find that for different artists, occluding very different parts of the image causes misclassification. In Willem Witsen's painting, the third in Figure 8, the negative space is more important for correctly classifying the images, as the dark blue areas indicate that covering up that specific area causes the model to classify incorrectly. However, for the other artists, the content is more important. This makes sense with Witsen's painting style, which is more monochromatic, and many of the training images in the dataset for Witsen contained blank or nearly blank canvases. Picart's image is misclassified as Witsen when the main content is fully occluded, which also makes sense, because without the picture in the image, the artwork is a blank canvas.

We also see similar results with saliency maps which show the pixels that have the most impact on the gradient of the final class scores. The Witsen painting also has the highest gradients in the negative spaces, while for the other images, the content pixels have more impact on the gradient.

### 4.2.2 Style Transfer

To understand whether our network learns more about the content or style of an artist, we apply style transfer across

Figure 9. Isaac Israel's paintings often have light colors

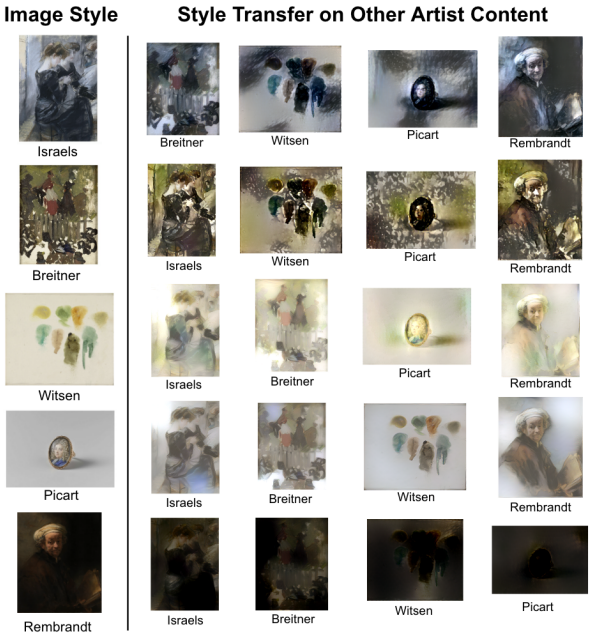



Figure 10. Results from Style Transfer on Select Input Images

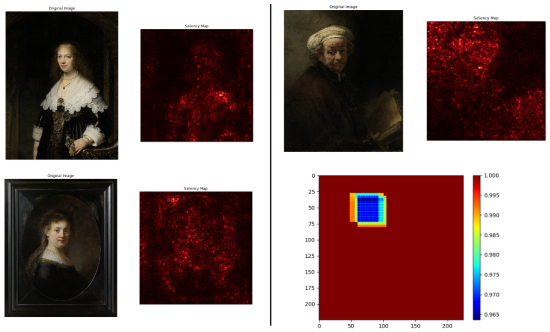
images from pairs of artists to create an image with content from one artist and the style of another. Sample results of style transfer are show in 10. The results vary significantly by artist pair, but often predict a third artist who uses similar colors to the style artist but content more similar to the second artist, for example when Willem Witsen's style is applied to other artists, the result is a much lighter image that still has recognizable content. The output image is most often classified as Isaac Israels, an impressionist whose paintings, like in figure 9 are often light but are focused on people or objects.

## 4.3. Deep Dive into Rembrandt

We select the most famous artist in the Rijksmuseum collection, Rembrandt, to conduct an even deeper analysis into understanding his style. In addition, our model seems to learn Rembrandt very well, as we can see on Table 1 on page 5, we achieve a precision of 0.951 and recall of 0.933, with an F1 score of 0.471 for Rembrandt. Rembrandt was famous for his self-portraits and portraits; his use of light and atmosphere draws attention to the subjects [7]. In our analysis below of the occlusion heat maps and saliency maps, as well as style transfer, we see how portraits, and human faces in particular, are emphasized in our model.

### 4.3.1 Occlusion Heat Map and Saliency Maps

In the saliency maps of three selected images from Rembrandt's work below, we see that there are clusters of red pixels around the face area, and even the shoulder areas (top left), which makes sense as portraits focus on the upper half of the body. This is revealed further in the occlusion heat map (right hand side of the figure), where covering the face causes our model to classify incorrectly. Thus, we see that in our network, higher importance is given to the subjects faces for Rembrandt.



### 4.3.2 Style Transfer Results

We hypothesize that given the prominence of portraits in Rembrandt's style, that perhaps even after transferring another artist's style to Rembrandt's images, that our model would still classify the image as Rembrandt. As described in the technical approach, we apply style transfer on select input images, and run the new image through our model to see which class gets predicted by our fine-tuned model. For Rembrandt specifically, we use Rembrandt as the content image, and apply the other nine artist styles to Rembrandt's image.

| Style | Content | Prediction |
|---|---|---|
| Rembrandt | Fokke | Rembrandt |
| Rembrandt | Vinkeles | Rembrandt |
| Rembrandt | Luyken | Rembrandt |
| Rembrandt | Bauer | Rembrandt |
| Rembrandt | Breitner | Breitner |
| Rembrandt | Tavenraat | Breitner |
| Rembrandt | Israels | Breitner |
| Rembrandt | Witsen | Breitner |
| Rembrandt | Picart | Bauer |
| Israels | Rembrandt | Rembrandt |
| Picart | Rembrandt | Rembrandt |
| Breitner | Rembrandt | Rembrandt |
| Fokke | Rembrandt | Rembrandt |
| Vinkeles | Rembrandt | Rembrandt |
| Bauer | Rembrandt | Rembrandt |
| Luyken | Rembrandt | Picart |
| Tavenraat | Rembrandt | Israels |
| Witsen | Rembrandt | Israels |

Table 2. Rembrandt Style Transfer

We find that our hypothesis is correct. 6 of 9 other artist styles applied to Rembrandt's painting get classified still as Rembrandt as show in Table 2. Thus, we see that the Rembrandt's content of portraits and human faces is heavily influential in our model, such that applying another artist's style does not affect our model.

Interestingly, we also find that when we apply Rembrandt's style to other artist's images, that our model predicts George Breitner as the artist as often as it predicts Rembrandt. Looking through a sample of Breitner's paintings, we note that Breitner uses similar colors to Rembrandt, but does not paint as many portraits as Rembrandt does.

## 5. Conclusion and Future Work

We have shown that it is possible to predict artists given a painting or work of art with over a 90% test accuracy. Given that we have achieved such a high accuracy this classifier could be packaged as is and used in a number of different contexts, such as a tool for students to look up pieces of art. Yet, there is definitely room for improvement if we hope for it to have broader applications. While we have have shown that we can classify artists who have fairly unique styles, it is unclear how extensible this solution is to recognize true artists such as in the case of forged paintings. There is definitely more stylistic analysis of brush strokes, attention, paint colors, and other features that we can do which may increase the robustness of our classifier. Additionally, we hope to extend the artistic analysis that we conducted for Rembrandt images to other artists to try and identify crucial features, and how they change for each artist, that the network learns. Although there is definitely room for improvement though additional analysis, our final results are promising, which indicate that this classifier is effective in predicting artist.

## References


[1] Cs 231n assignment 2 code, 2017.

[2] Cs 231n assignment 3 code, 2017.

[3] S. Agarwal, H. Karnick, N. Pant, and U. Patel. Genre and style based painting classification. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 588–594. IEEE, 2015.

[4] R. S. Arora and A. Elgammal. Towards automated classification of fine-art painting style: A comparative study. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3541–3544. IEEE, 2012.

[5] Y. Bar, N. Levy, and L. Wolf. Classification of artistic styles using binarized features derived from a deep neural network. In *Workshop at the European Conference on Computer Vision*, pages 71–84. Springer, 2014.

[6] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[7] Q. Buvelot, C. White, and N. Mauritshuis (Hague. *Rembrandt by Himself*. National Gallery Publications, 1999.

[8] E. Cetinic and S. Grgic. Automated painter recognition based on image feature extraction. In *ELMAR, 2013 55th International Symposium*, pages 19–22. IEEE, 2013.

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

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

[11] C. R. Johnson, E. Hendriks, I. J. Berezhnoy, E. Brevdo, S. M. Hughes, I. Daubechies, J. Li, E. Postma, and J. Z. Wang. Image processing for artist identification. *IEEE Signal Processing Magazine*, 25(4), 2008.

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

[13] J. Johnson. pytorch-finetune. `https://gist.github.com/jcjohnson/6e41e8512c17eae5da50aebef3378a4c`, 2017.

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

[15] F. S. Khan, S. Beigpour, J. Weijer, and M. Felsberg. Painting-91: A large scale database for computational painting categorization. *Mach. Vision Appl.*, 25(6):1385–1397, Aug. 2014.

[16] F. S. Khan, J. van de Weijer, and M. Vanrell. Who painted this painting. In *The CREATE 2010 Conference*, 2010.

[17] E. Levy, O. E. David, and N. S. Netanyahu. Genetic algorithms and deep learning for automatic painter classification, 2014.

[18] T. Mensink and J. van Gemert. The rijksmuseum challenge: Museum-centered visual recognition. In *ACM International Conference on Multimedia Retrieval (ICMR)*, 2014.

[19] O. Moindrot. tensorflow_finetune. https://gist.github.com/omoindrot/dedc857cdc0e680dfb1be99762990c9c, 2017.

[20] B. Saleh and A. Elgammal. Large-scale classification of fine-art paintings: Learning the right metric on the right feature. *arXiv preprint arXiv:1505.00855*, 2015.

[21] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[22] N. van Noord, E. Hendriks, and E. Postma. Toward discovery of the artist's style: Learning to recognize artists by their artworks. *IEEE Signal Processing Magazine*, 32(4):46–54, 2015.

[23] N. van Noord, E. Hendriks, and E. O. Postma. Toward discovery of the artist's style: Learning to recognize artists by their artworks. *IEEE Signal Process. Mag.*, 32(4):46–54, 2015.

[24] N. Westlake, H. Cai, and P. Hall. Detecting people in artwork with cnns. In *European Conference on Computer Vision*, pages 825–841. Springer, 2016.

[25] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.