Recognizing Art Genre with Machine Learning

Jonathan Avila, Joel Krim, Nitay Caspi

CIS 519 Fall 2018

1 Introduction

One of the many interesting topics in machine learning is using learning algorithms to analyze and recognize art. Among this field of work, several authors have attempted to recognize the *style* of a painting with machine learning: in other words classifying paintings by their style. In this paper, we will take a similar approach to recognize the *genre* of a painting. The implementation of such a classifier (as explained below) draws on various components of machine learning that we have learned in this class, and also offers the opportunity to expand beyond the lecture material and apply our knowledge to the interesting problem of defining art genre.

A work of art can be described with a set of attributes like *style*, *genre*, *medium*, etc. These attributes are of interest to art historians, curators, and the public because they help build an understanding of a particular artwork, and they illustrate trends in the art world. While art can take many forms, this paper will only consider the medium of "painting". From a computational perspective, paintings are relatively easy for machines to process given their two-dimensional nature. Additionally, a lot of recent work in Machine Learning research has focused on image processing and recognition, resulting in state-of-the-art algorithms (described below) that learn on images.

2 Related Work

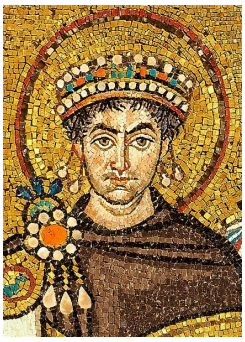
Most existing research in this area has focused on art *style*. Two papers have recently been published that attempt to classify images of artworks based on their style: Karayev et al. (2014) and Lecoutre et al. (2017). Karayev et al. worked with the Flikr and Wikipaintings datasets to predict artistic style using a Stochastic Gradient Descent model. This paper used several types of features, including color histograms, GIST features (low-dimensional representation), and Graph-based visual saliency features (to model human visual fixation patterns) [3]. Lecoutre et el. use a deep residual neural network trained on the ImageNet dataset, then re-train it on the Wikipaintings dataset labeled with visual style [5]. Another paper by Tan et al. (2016) used similar procedures to classify painting style and genre [7]. The results from these papers are summarized in the Figure 2 below.

The findings from this area of research have shown that recognizing and understanding visual style is computationally viable, and other papers have expanded on this finding by generating images with particular styles. Gatys et al, use a generative model to "re-draw" one image in the style of another image. For example, their model takes an image (say Van Gogh's "Starry Night") and transforms any other image with the unique swirls and shades from the original image [1].

In this paper, we will attempt to classify paintings by their *genre*, similar to Tan et al [7]. Paintings are typically categorized into genre based on their subject matter. For example, paintings in the "portrait" genre are classified as such because their subject matter is a person, while "landscape" paintings are defined by their depiction of natural landscapes, such as mountains, valleys or rivers. We expect that in order to learn these classes, a classification

Figure 1: Examples of WikiPaintings Images and Genres

(a) Mona Lisa



(b) Mosaic Portrait

(c) Portrait of a Young Man

(d) Venice Cityscape

(e) Markeplace in Pirna

Images in top row are classified in the portrait genre. Images in the bottom row are classified in the cityscape genre. Note the similarities across genres: for example, a portrait will always have a face and a cityscape will have a foreground and a background with buildings and sky. However, there are also important differences between the classes (such as rotation of faces, orientation of buildings and sky, etc.).

Figure 2: Summary of Related Research

Paper	Karayev et al. (2014)	Tan et al. (2016)	Lecoutre et al. (2017)
Dataset	Flickr & Wikipaintings	WikiPaintings	WikiPaintings
Best Architecture	Linear Classifier	AlexNet variant	Re-Trained ResNet
Classification Task	Style	Genre*	Style
Number of Classes	25	10	25
Best Result	0.441 Average Precision	74.14% Acc.	62.5% Acc.

*Paper may have included other classification tasks, but we report only the most relevant result to this study.

algorithm will need to learn spatial and object-related features. Therefore, we expect the CNN architectures used in previous research to work well for this problem.

3 Problem Definition and Algorithms

3.1 Task Definition

We will compare two state-of-the-art, deep convolutional neural network architectures, as well as a simple convolutional neural network as a baseline, on classifying images from the WikiPaintings dataset based on their genre. In this classification task, we will take images of paintings and classify them into one of nine genre classes, which include: abstract, cityscape, genre painting, illustration, landscape, portrait, religious painting, sketch and study, and still life.

3.2 Algorithms Definition

We will train deep convolutional neural networks to perform the classification tasks. Each network will have a different architecture (described below).

3.2.1 Basline CNN

We first tested a fairly simple Convolutional Neural Network architecture, trained on a portion of the images from WikiPaintings. This network consisted of three convolutional layers with 3x3 kernels, each followed by a 2x2 max-pooling layer. The third convolutional layer is followed by one dense layer.

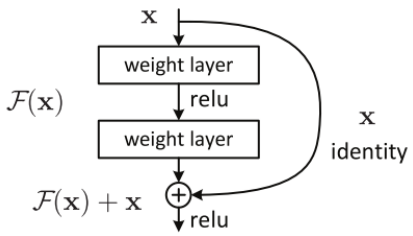
3.2.2 VGGNet

The VGGNet architecture, developed at the Visual Geometry Group at the University of Oxford, is similar to AlexNet [4]. It is composed of five groups of layers, with each group containing two or three convolutional layers followed by a max pooling layer. After the convolutional layers, there are three fully connected layers of decreasing size until the output softmax layer. In total, VGGNet has 138 million parameters [6].

3.2.3 ResNet

The ResNet architecture was developed by a team at Microsoft Research with the hopes of solving the vanishing-gradient problem inherent in deep networks. The vanishing-gradient problem is a result of the fact that gradients have to be back-propagated across many layers, which can result in extremely small values once the propagation reaches the shallowest layers. To remedy this, ResNet is composed of residual blocks, which are blocks of convolution layers after which the input to the block is added to the output of the block, and then fed to the next block (Figure 3) [2].

Figure 3: Residual Block



He et al. [2]

3.2.4 Training Procedure

We ran our preprocessing, training, and testing pipelines on a Kaggle Kernel, which allowed us to harness cloud computing power (including the use of GPUs), without using the limited space on our local machines to store the large dataset. The GPU used is a NVIDIA Tesla K80 GPU, with 2 CPU cores and 14 GB of RAM. We used the Keras library, with a Tensorflow backend, to implement the data preprocessing, training, and testing pipeline, as well as scikit−learn and matplotlib for data visuaulization.

Each image was downsampled to fit the input dimensions of the respective CNN architecture: 224x224x3. For some trials (see Section 4.4) we applied the augmentation steps for preprocessing (see Section 4.3). The data were split into 80% training and 20% validation (because the data is from a Kaggle competition, the test labels are not available and thus we cannot measure the model's performance on the test set).

For the baseline CNN, we fully trained the network on the training data using categorical cross-entropy loss and Stochastic Gradient Descent. For the other architectures, we used weights pre-trained on the ImageNet dataset for all the layers until the output layer, which we replaced with a custom output layer to fit the number of classes in our data. Only these final output weights were trained.

Every model was trained over 50 epochs. In each epoch, the full dataset was trained in batches of 64 images, with validation steps every 4 batches. All results are reported either per-epoch metrics, or outputs from the final epochs.

3.3 Expectations

We expect genre classification to perform similarly to style classification: deeper, more complex networks (i.e. networks with more parameters) should result in better performance. This expectation is derived from our understanding of the definition of genre in this context. A painting's genre is highly dependent on its content, which translates to geometric features such as shapes and higher-dimensional features like objects. Since deeper convolutional networks tend to perform better on object recognition tasks, we expect them to also perform better on genre classification. One potential problem is that some genre classes are very similar, especially cityscape and landscape. We expect there to be higher errors on these classes.

4 Experimental Evaluation

4.1 Methodology

For each of the CNN architectures outlined above, we train according to the procedure in 3.2.4 using the data described in 4.2. For each epoch, we calculate the accuracy, loss, precision, and f1-score for both training and validation sets. Then, we report the accuracy, loss, precision, and f1-score for the final epoch as the final performance measures of the model.

4.2 Data

We used data from the WikiPaintings dataset, which is a collection of about 80,000 images labeled by artist, style, genre, and medium [1]. Although there were 60 unique genre classes among the paintings in the dataset, the classes were heavily unbalanced (some genre classes contain over 10000 examples, while others contain less than 10), so we decided to only use only classes with greater than 2000 examples (see Figure 4). Figure 1 illustrates several instances from this dataset, as well as the examples of two genre classes (portrait and cityscape).

4.3 Data Augmentation

On some trials, we pre-processed the images with augmentation techniques. During these trials, each image was randomly rotated by up to 20 degrees, randomly shifted vertically or horizontally by up to 20%, randomly flipped horizontally, or randomly applied a shear transform. Each of these transformations ocurred with 0.5 probability.

4.4 Results

We first trained the simple CNN architecture on the roughly 50,000 training images from the WikiPaintings dataset. As expected, this architecture performed very poorly.

[1] https://www.wikiart.org/

Figure 4: Class Distributions

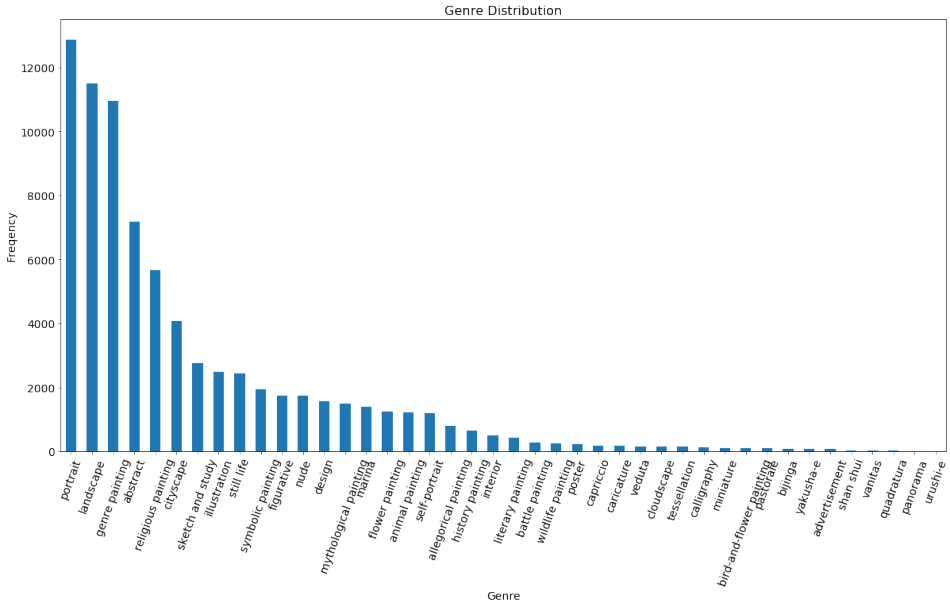


Figure 5: Baseline CNN Performance

(a) Accuracy

(b) Cross-Entropy Loss

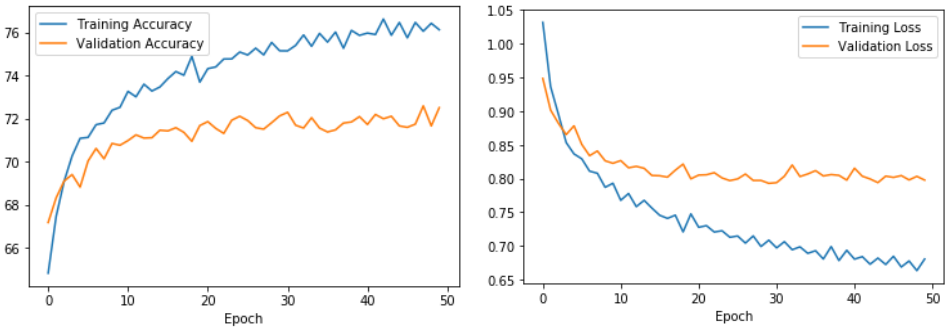
These results could have several meanings. First of all, since the accuracy hovers around 12%, which is roughly 1/9, the CNN is essentially randomly guessing on the classes. One obvious issue with this trial is that the CNN is only trained on 50,000 images, which is a much smaller training set than typical image recognition models. This also means that more complicated architectures are necessary to learn genre, if it is possible at all.

Next, we ran the pre-trained architectures, VGGNet and ResNet. As expected, ResNet performed best, achieving a validation accuracy of 72.6% and a Top-3 accuracy of 94.0%. We then compared the ResNet and VGG results on raw data versus augmented data, and the models trained on augmented data performed slightly better (Figures 7, 9). Compared to results from previous research, our results are similar to those of Tan et al. [7] for genre classification.

4.5 Discussion

Overall our results matched our expectations. We expected ResNet to perform better than VGG and our custom CNN architecture, which is consistent with the results. The most unexpected result was the relative success of the trials with augmented data versus trials with regular data. For ResNet, training on augmented data resulted in higher accuracy but slightly lower precision and f1 (Figure 9). Similarly, the VGG network trained on augmented images performed much better than the non-augmented trial (Figure 7). This might be explained by the higher variation in training instances that are augmented, which

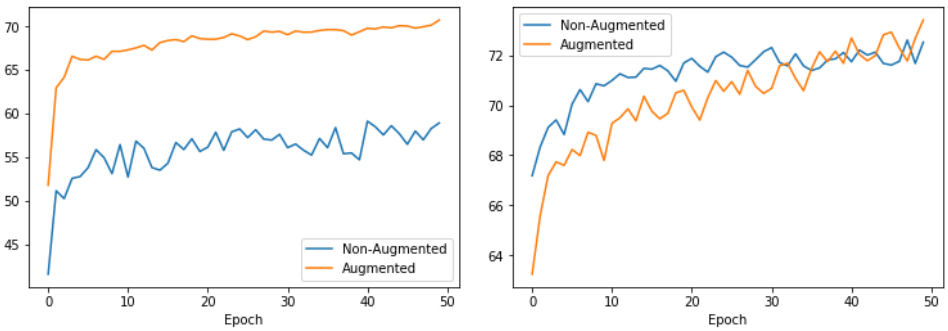
Figure 6: ResNet Performance



(a) Accuracy

(b) Cross-Entropy Loss

Figure 7: Performance on Augmented v. Non-Augmented Data



(a) VGGNet

(b) ResNet

could increase generalization to the witheld validation set. In other words, shifting, rotating, and transforming the images increases the robustness of the data.

Figure 8 contains a Confusion Matrix of the Resnet Non-Augmented results. The confusion matrices for both VGG models are similar, and are thus not reported. In general, the precision per class was robust: genres that are not visually similar to one another, such as abstract paintings and still-lives are rarely misclassified. Pairs of genres with a high degree of visual and artistic overlap, such as portraits & sketch and study works, or genre paintings & illustrations are the only classes that are misclassified. This is to be expected, as these paintings would fall into both categories more often than not. This is reinforced by the results of our top-3 accuracy for all the deep, convolutional models which have a high degree of accuracy (between $94-95\%$).

Rather than comparing CNN architecture performance, we can also interpret this study as comparing several feature extraction procedures. Each pre-trained CNN outputs a set of features for each image, and a simple multi-class linear classifier is trained on these features. In this respect, the best performance is a result of the most robust features, which are generated from ResNet. This interpretation is similar to the study in Karayev et al. [3]. Using this interpretation, future work could further improve accuracy by adding more features, as in Karayev et al.

5 Conclusion

In this paper, we presented a study comparing various architectures of deep convolutional neural networks on the task of classifying paintings by genre. Our results show that large amounts of data are necessary to train CNNs in order to perform this task well, as confirmed by the poor performance on the baseline CNN. Further, the successful use of pre-trained

Figure 8: Confusion Matrix on ResNet Results

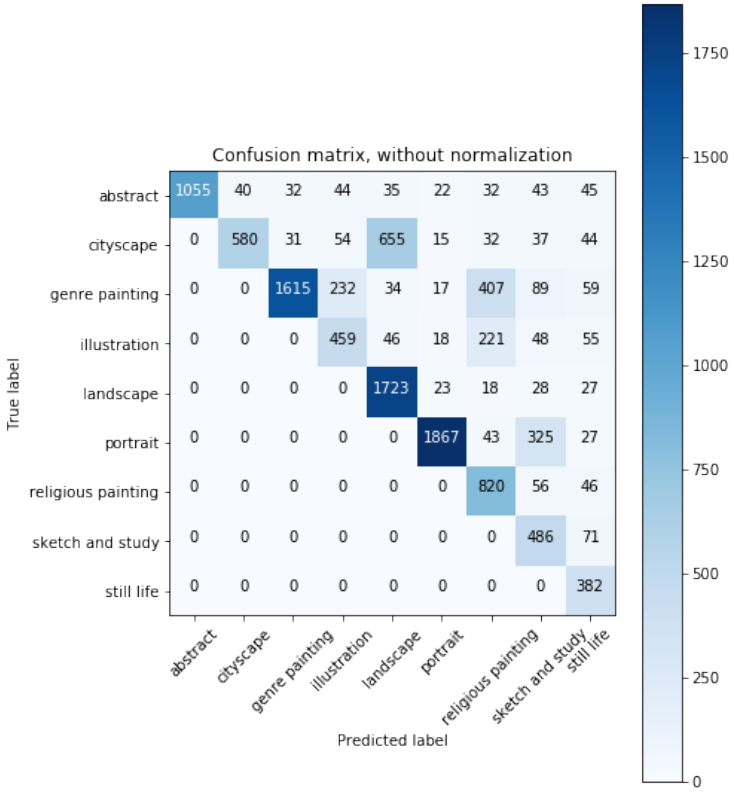


Figure 9: Summary of Results

	Simple CNN	VGGNet	VGGNet Augmented	ResNet	ResNet Augmented
Top-1 Accuracy	12.3%	58.9%	73.1%	72.6%	**73.4%**
Top-3 Accuracy	-	87.6%	94.1%	94.0%	**95.1%**
Precision	-	0.5913	0.7389	**0.7893**	0.7853
F1	-	0.5893	0.7073	**0.7158**	0.7031

weights shows that the features necessary to identify genre are similar to the features used to recognize image concepts like animals or vehicles, since the weights are pre-trained on image recognition task from ImageNet. Finally, our results show that preprocessing images with augmentation yields better results than training on raw images.

References

[1] Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. “A neural algorithm of artistic style.” arXiv preprint arXiv:1508.06576 (2015).

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[3] Karayev, Sergey, et al. “Recognizing Image Style.” Proceedings of the British Machine Vision Conference 2014, 2014, doi:10.5244/c.28.122.

[4] Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E. (2017-05-24). ”ImageNet classification with deep convolutional neural networks”. Communications of the ACM. 60 (6): 84–90.

[5] Lecoutre, Adrian, Benjamin Negrevergne, and Florian Yger. “Recognizing Art Style Automatically in painting with deep learning.” Asian Conference on Machine Learning. 2017.

[6] Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556, 2014.

[7] Tan, Wei Ren; Chee Seng Chan, Hernan E Aguirre, and Kiyoshi Tanaka. Ceci nest pas une pipe: A deep convolutional network for fine-art paintings classification. In Image Processing (ICIP), 2016 IEEE International Conference on, pages 3703–3707. IEEE, 2016.