Identifying Artists based on their Artwork

Aditi Krishnakumar, Himanshi Gupta
COGS 181

1 Abstract

Authenticating artwork through visual analysis by experts is a time-consuming process that can lead to errors. To address this issue, this project investigates the potential of Convolutional Neural Networks (CNNs) for artist identification. The dataset used includes 81 paintings per artist from 36 artists, with images resized to 256x256 pixels. The study trains several models, including a baseline CNN and ResNet18 and ResNet50 with transfer learning and fine-tuning techniques. The results showcase the effectiveness of CNNs for identifying artists, with the best model achieving an accuracy of 48%. The project sheds light on the impact of various hyperparameters on model performance, such as learning rate, number of epochs, and fine-tuning layers. This study contributes significantly to the field of art identification using Deep Learning, revealing the potential of CNNs to learn artistic style representations. The findings could have far-reaching implications for art authentication, preservation, and curation and pave the way for the development of automated artist identification systems.

2 Introduction

Art authentication and identification have been long-standing challenges for the art world. The ability to attribute works to specific artists is crucial for establishing their authenticity, provenance, and value. Traditionally, art identification has been performed by experts through visual analysis of the painting style, brushstrokes, and other features. However, this approach can be time-consuming, subjective, and error-prone. The use of Deep Learning and Computer Vision techniques offers a promising solution to this problem.

Convolutional Neural Networks (CNNs) have emerged as a powerful tool for image recognition and classification tasks. They have been successfully applied in various domains, including object recognition, facial recognition, and medical image analysis. In recent years, CNNs have also been used for art identification, with promising results. One of the main advantages of CNNs is their ability to learn a representation of image features automatically, which can capture subtle variations in artistic style that are difficult for humans to perceive.

3 Related work

We base our work on Jennie Chen and Andrew Deng's paper [1] which compared the performances of a CNN and an SVM model on the WikiArt Dataset. They were able to achieve an accuracy of 74.7% with their CNN and 68.1% with their SVM, using a dataset of 7462 paintings from 15 artists. Our project focuses on Deep Learning methods alone and also includes a larger number of artists (36).

Work on Artist Identification has also been done by Balakrishnan et al [2] on the Rijksmuseum dataset which involved comparing the performances of pre-trained ResNet and VGG models. This study focused on the work of 10 artists with the most number of paintings.

4 Dataset and Features

WikiArt [3] is a large dataset containing around 250,000 high-quality digital images of artwork from various periods, styles, and cultures, ranging from classical to contemporary art. The dataset contains information about the genre of the artwork, the time period, the title, and the artist's name. We primarily utilized the 'Painters by Numbers' [4] dataset compiled by Kaggle using the WikiArt data, however, due to computational limitations, we used a much smaller subset of the dataset containing a total of 19,501 records. The range of artworks per artist varied from 1 to 119 so in order to balance the classes, we took a subset with 81 artworks per artist with a total of 36 artists. Each color image was resized to 256x256 pixels and normalized according to the ImageNet specifications. Normalization of images before feeding them into a CNN helps in ensuring that the input data has the same scale and prevents any bias that may occur due to the distribution of pixel values. It also helps in generalization of the model by making the model invariant changes in things such as contrast and lighting that may affect pixel values as well as reducing the impact of outliers in the data.

We then proceed to create a custom Dataset and DataLoader according to Pytorch specifications. Our training set was split into train:val with a 80:20 ratio. Our test set had a total of 1080 records; we ensured that it includes the same 36 artists we have trained our model on and a balanced set of classes with 30 artworks per artist.

Fig1: Image before normalization

Fig2: Image after normalization

5 Methods

The goal of this project was to train Convolutional Neural Networks (CNNs) to identify the artist of a painting. The dataset used consisted of 81 paintings per artist from 36 artists. Since the dataset was too large, a subset of the dataset was used, called "df_80", which had data from the original dataset with artists having at least 80 paintings (maximum being 119). The dataset was further resampled so that there were 36 artists, each with 81 paintings. The images were resized to 256x256 pixels and normalized before being fed into the model.

The baseline CNN model used in this project was based on a previous paper by Chen et al. [1], and it was observed that the model overfit the data, likely due to the smaller dataset size. Therefore, various ResNet models were trained with Transfer Learning and fine-tuning techniques to improve the accuracy of the model.

5.1 CNN

For our baseline CNN, we used 6 2D convolutional layers using 3x3 filters and 5 max-pooling layers which reduced the dimensionality of the feature maps by a factor of 2. Each convolutional layer was followed by a batch normalization layer and used a ReLU activation function. These were then flattened and passed through a dropout layer, two linear layers and another dropout layer. Batch normalization helps the convergence of the model and Dropout helps in preventing overfitting by randomly 'dropping' off a subset of neurons making the network more robust. As mentioned previously, our architecture was based on the work done by Chen and Deng [1]. The image included of the architecture is also obtained from their paper.

We cross-entropy loss for our loss function and the ADAM optimizer with a learning rate of 0.001. We trained our model for 100 epochs and used a batch size of 32.

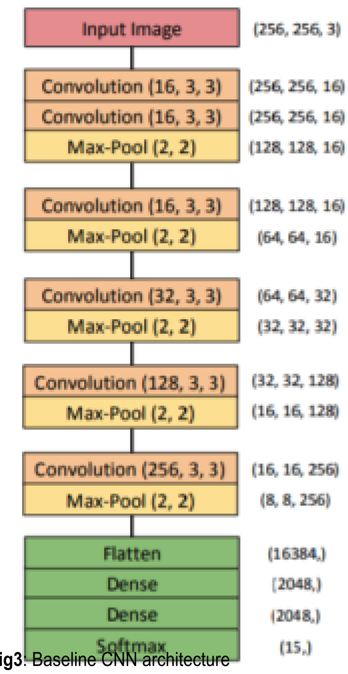

Fig3: Baseline CNN architecture

5.2 ResNet

Given our smaller sized dataset and computational limitations, we believed we would be able to obtain a much better performance using Transfer Learning with a model pre-trained on the significantly larger ImageNet. We decided to focus on the ResNet model and tune various hyperparameters in order to improve the performance.

Initially, we looked at the ResNet-18 modifying the last layer to include a softmax operation and fine tuning this last layer on our dataset. We used an ADAM optimizer with a learning rate of 0.0001 and cross-entropy loss. We trained this for 10 epochs.

We then trained the same model for 20 epochs with a higher learning rate of 0.001. This significantly improved our performance. We then also looked at the impact of fine-tuning a larger number of layers on our dataset and found this improved our accuracy as well.

Wanting to look at the performance of a deeper network, we fine-tuned a ResNet-50 model as well. We looked at changing the number of epochs from 10 to 20 and fine-tuning a larger number of layers on our model.

6 Experimental Results

The performance metric we primarily focused on was accuracy. We additionally included a classification report which looked at precision, recall, and f-1 score to provide more insight into the models' performances. A confusion matrix allowed us to look specifically at which artists were misclassified. A training and validation loss graph also helped in understanding whether the model was overfitting or not. All models were trained on a CPU and evaluated on a test set.

Baseline CNN: This model achieved an accuracy of 9%. Although we are using the same dataset and model as the work by Chen et al. They were able to achieve a much higher accuracy of around 74%. The only difference would be the higher number of classes and the smaller dataset size we use which we suspect lead to overfitting.

Looking at the loss graph, we can see the training loss steadily decreasing while the validation loss increases. This confirms our hypothesis that the model has overfit. In the future, we can try overcoming this by using a larger dataset or using fewer layers in our baseline CNN.

Additionally, multiple classes had a precision, recall, and f1-score of 0.0 which means that the model did not predict any of the samples belonging to these classes correctly. Looking at the confusion matrix, we can see that there are a few classes that have not been predicted at all. The model is predicting only a few of the classes very frequently.

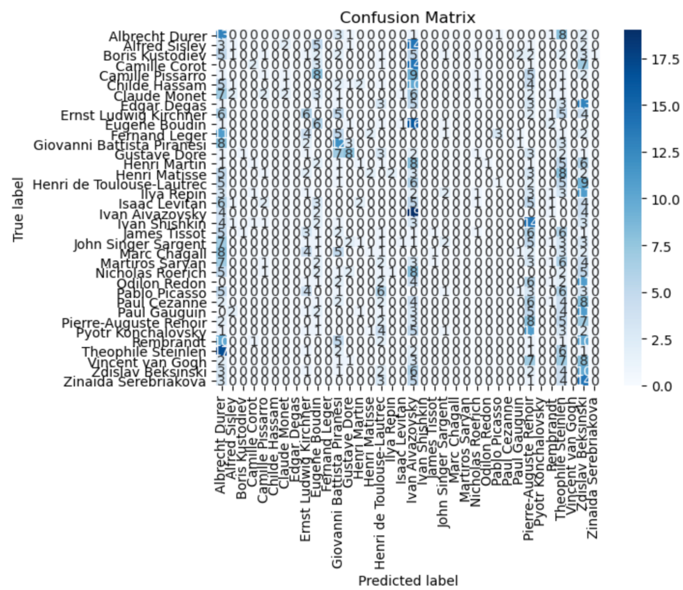

Fig4: Baseline CNN Confusion Matrix

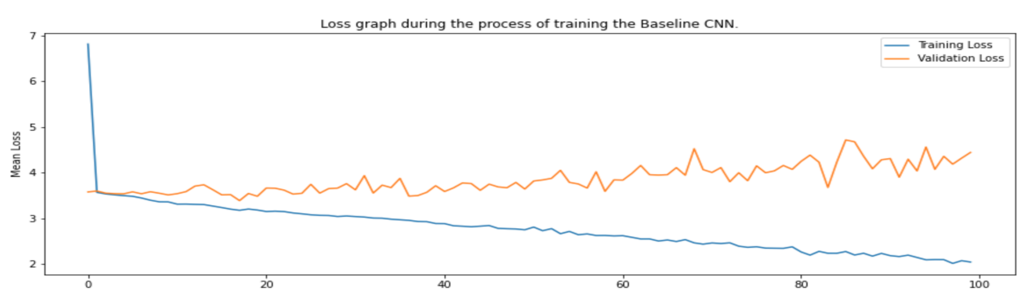

Fig5: Baseline CNN Validation Curve

ResNet18 with 0.0001 learning rate and 10 epochs with transfer learning: This model achieved an accuracy of 26%. The significant improvement with a pre-trained model tells us that the size of our dataset in comparison to the number of classes has impacted our results greatly. The validation curve is decreasing steadily so we decide to train this model for a larger number of epochs in order to achieve a higher accuracy.

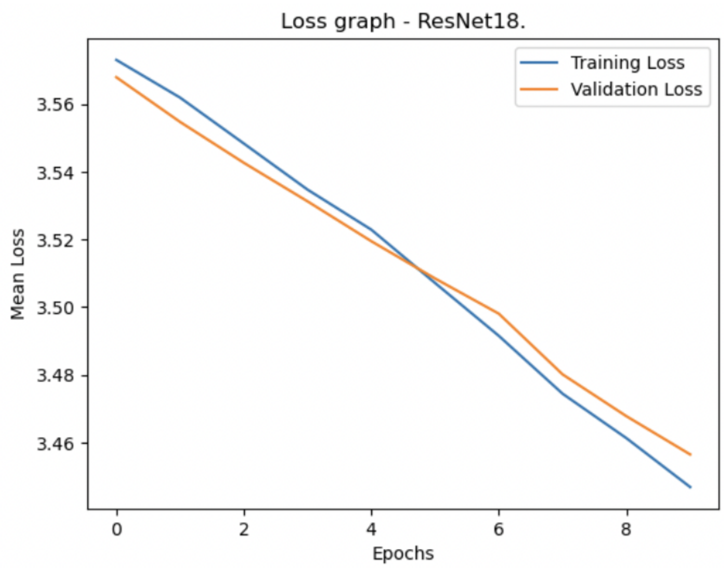

Fig6: ResNet18 Validation Curve

ResNet18 with 0.001 learning rate and 20 epochs with transfer learning: This model achieved an accuracy of 40%. Increasing the learning rate and the number of epochs almost doubled our accuracy. The validation curve is starting to plateau which suggests that training for a larger number of epochs (>20) would potentially not improve the model further.

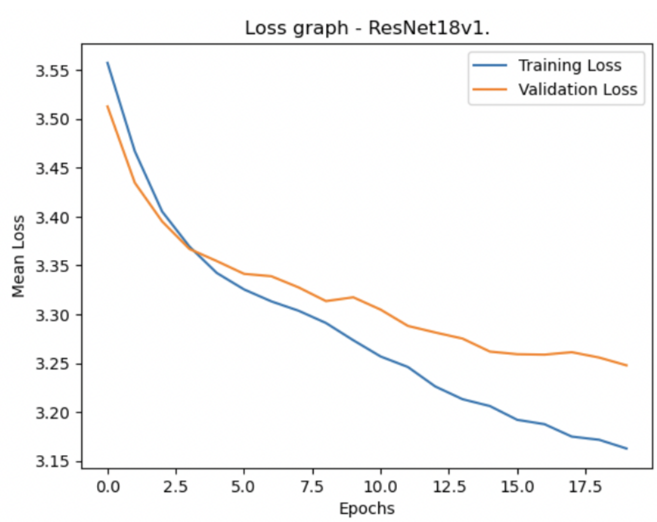

Fig7: ResNet18 Validation Curve

ResNet18 with 0.001 learning rate and 20 epochs and fine-tuning last two layers with transfer learning: This model achieved an accuracy of 48%. After looking at the validation curve of the previous model, we decided training for 20 epochs is sufficient and instead focused on fine-tuning our pre-trained model further. We fine-tuned the last fully connected layer and the layer just before this (layer 4) on our dataset. This gave us the highest accuracy on our dataset.

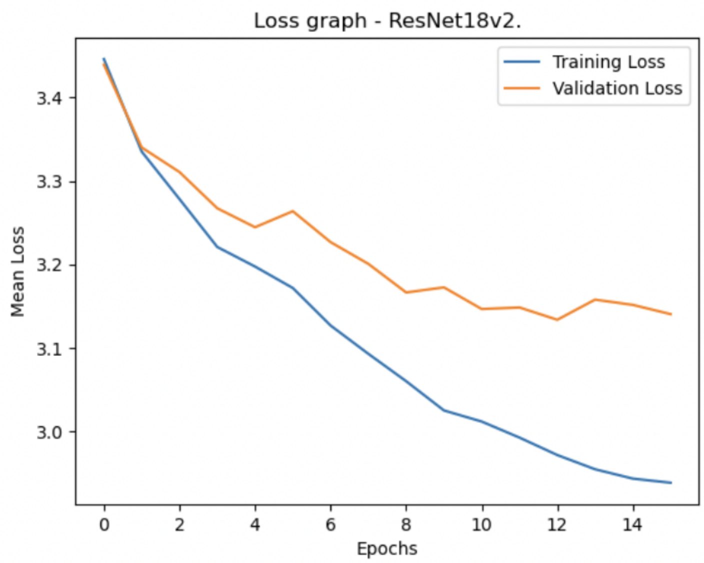

Fig8: ResNet18 Validation Curve

	precision	recall	f1-score	support
0	0.55	0.57	0.56	30
1	0.00	0.00	0.00	30
2	0.00	0.00	0.00	29
3	0.00	0.00	0.00	27
4	0.00	0.00	0.00	30
5	0.55	0.53	0.54	30
6	0.32	0.64	0.43	28
7	0.59	0.55	0.57	29
8	0.44	0.87	0.58	30
9	0.66	0.86	0.75	29
10	0.00	0.00	0.00	30
11	0.75	1.00	0.86	30
12	0.83	0.80	0.81	30
13	0.73	0.80	0.76	30
14	0.34	0.62	0.44	29
15	0.26	0.63	0.37	30
16	0.59	0.33	0.43	30
17	0.29	0.50	0.37	30
18	0.96	0.79	0.87	29
19	0.61	0.82	0.70	28
20	0.62	0.34	0.44	29
21	0.50	0.77	0.61	30
22	0.00	0.00	0.00	30
23	0.18	0.55	0.27	29
24	0.67	0.80	0.73	30
25	0.63	0.41	0.50	29
26	0.00	0.00	0.00	29
27	0.54	0.70	0.61	30
28	0.00	0.00	0.00	30
29	0.57	0.57	0.57	28
30	0.43	0.41	0.42	29
31	0.51	0.75	0.61	28
32	0.52	0.52	0.52	29
33	0.31	0.50	0.38	30
34	0.68	0.72	0.70	29
35	0.00	0.00	0.00	29
accuracy			0.48	1056
macro avg	0.41	0.48	0.43	1056
weighted avg	0.41	0.48	0.43	1056

Fig9: ResNet18 Classification report

ResNet50 with 0.001 learning rate and 10 epochs with transfer learning: This model achieved an accuracy of 39%. Wanting to compare another architecture, we utilized a deeper ResNet with 50 layers and performed a similar analysis on this. While the ResNet50 achieved a higher accuracy than the ResNet18 trained with the same number of epochs, it did not surpass the performance of our further fine-tuned ResNet18.

ResNet50 with 0.001 learning rate and 20 epochs and fine-tuning last two layers with transfer learning: This model achieved an accuracy of 32%. Surprisingly, the deeper ResNet50 did not improve the accuracy in comparison to the ResNet18 with the same hyperparameters and fine-tuning. However, in this case there were no classes that had 0.0 for precision, recall, and f1-score while with our ResNet18, there continued to be some classes that faced this issue.

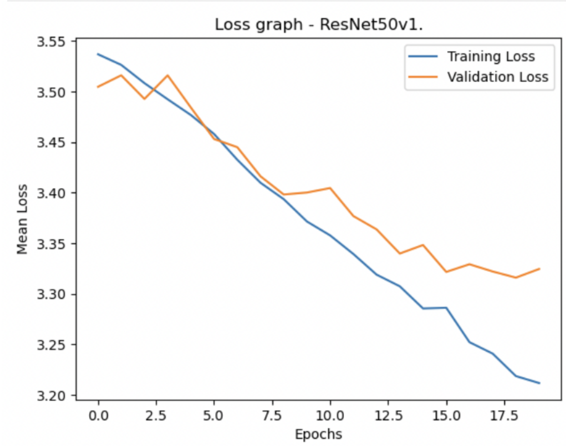

Fig10: ResNet50 Validation Curve

Looking at the confusion matrix of our best performing mode with 48% accuracy, we wanted to perform a qualitative analysis on the misclassified images. 14 out of the 30 artworks tested of Fernand Leger's were incorrectly classified as belonging to Martiros Saryan. Looking at some of their artworks individually, it is evident that the larger brushstrokes of Martiros Saryan are quite different to that of Fernand Leger and therefore a key mark of differentiation between the two. Perhaps we can strengthen our model by including an element of brushstroke analysis or cropping images to focus on smaller, more precise areas of the artworks.

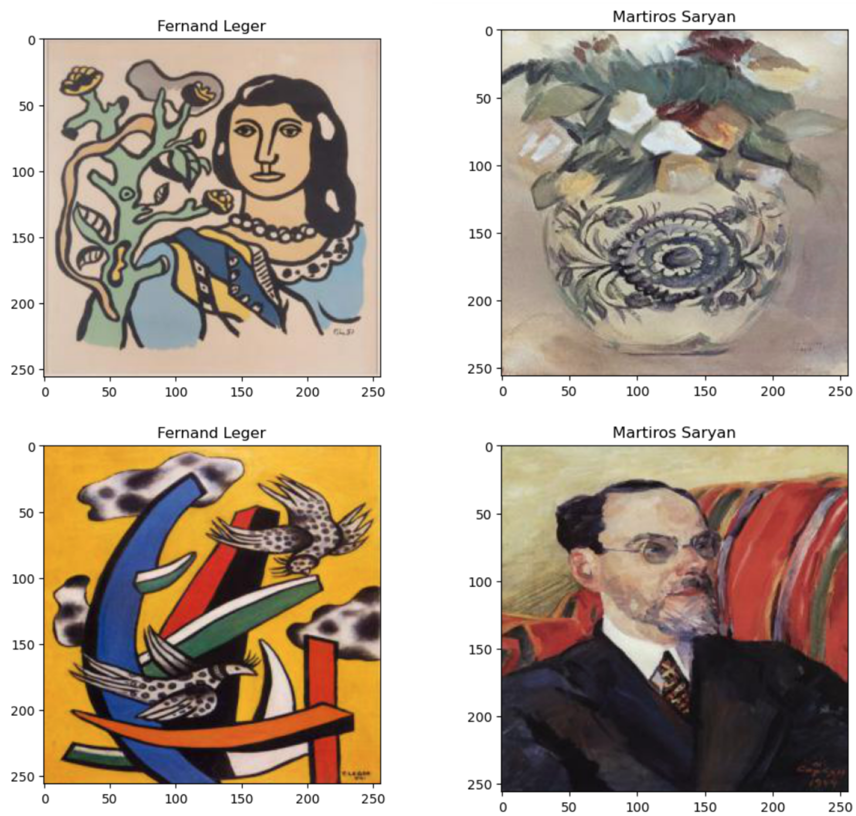

Fig11: Qualitative Analysis

7 Conclusion and Future Work

In this project, we aimed to identify the artist of a painting using convolutional neural networks (CNNs) and explored several architectures and training strategies to improve the accuracy of our model. Our results show that ResNet18 with a learning rate of 0.001 and training for 20 epochs with fine-tuning of the last two layers achieved the best accuracy of 48%.

Our findings demonstrate that CNNs are a powerful tool for artist identification, and that transfer learning and fine-tuning can significantly improve the accuracy of the model. The best-performing model, ResNet18 with fine-tuning of the last two layers, was able to learn a representation of the painting style that allowed it to accurately identify the artist of a painting.

One potential limitation of our study is the small size of our dataset, which may have affected the generalizability of our findings. While our dataset included a diverse set of artists and paintings, we were only able to include a limited number of paintings per artist. In future work, it may be beneficial to use a larger dataset to further explore the accuracy and generalizability of CNNs for artist identification.

Another limitation of our study is the lack of analysis of the features learned by the CNN models. Understanding the features learned by the model could provide insights into the characteristics of an artist's style and the specific aspects of a painting that contribute to its classification. Future work could explore the interpretability of CNN models for artist identification using strategies such as Saliency Maps.

This study focused on CNNs. Future work could compare different architectures such as Transformer models. Transformers have the advantage of capturing longer range dependencies and could improve our accuracy greatly.

In conclusion, our study demonstrates the potential of CNNs for artist identification and highlights the importance of transfer learning and fine-tuning for improving the accuracy of the model. Our findings have implications for art history and authentication, as well as for the development of computer vision algorithms that can identify and classify visual styles in other domains. With the increasing availability of digital art collections, the use of machine learning techniques for art analysis is likely to become more prevalent, and we hope that our study contributes to this growing field of research.

8 References

[1] Chen, Jennie, and Andrew Deng. "Comparison of Machine Learning Techniques for Artist Identification." Stanford University, 2018, http://cs229.stanford.edu/proj2018/report/41.pdf.

[2] Balakrishan, Tara, Sarah Rosston, and Emily Tang. "Using CNN to Classify and Understand Artists from the Rijksmuseum." Stanford University, 2017, http://cs231n.stanford.edu/reports/2017/pdfs/410.pdf.

[3] "WikiArt.org." WikiArt, https://www.wikiart.org/.

[4] "Painter by Numbers." Kaggle, https://www.kaggle.com/competitions/painter-by-numbers.