Image Classification using Convolution Neural Networks

Rishabh Mer
University College Dublin
rishabh.mer@ucdconnnect.ie

Abstract

In this paper, I explore the use of a convolution neural network (CNN) for image classification. There are various pre-trained neural networks like Vgg16, RestNet50, Alex Net, etc, and many image datasets. This study focuses on the training and evaluation of images of paintings called wikiart dataset using a custom CNN model and comparing against various pre-trained models.

1 Introduction

1.1 Problem Statement

The problem statement for this project centers around image classification using deep learning models. While there are many pre-trained models available for use, including Vgg16, and ResNet50, this study focuses on the development and evaluation of a custom convolutional neural network (CNN) model. The dataset used for this study is the wikiart dataset, which contains over 150,000 images of paintings and a CSV file that provides classification by style, genre, movement, and artist. The goal of this project is to classify images based on their style label

1.2 Motivation

One of the main challenges in image classification is the ability to accurately categorize images based on their label or tag. This can be particularly difficult when working with a large and diverse dataset such as wikiart, which contains paintings from many different artists and movements. By developing a custom CNN model and comparing it to state-of-the-art pre-trained models, this study aims to identify the most effective approach to image classification on this dataset. Ultimately, the goal is to improve the accuracy and efficiency of image classification using deep learning methods and provide insights into the performance of different models on this type of data.

Outline In the paper, I have included the results of the experiments, which provide insights into the performance of the different models and the impact of different hyperparameters on accuracy. Then proceeded to a conclusion and future work section, where we interpret the results and consider future upgradation of the project. Finally, I have provided a list of references cited in the text.

Overall, this report paper contributes to the growing body of literature on image classification using deep learning techniques and highlights the importance of selecting appropriate models and hyperparameters for achieving high accuracy.

2 Related Work

This project was inspired by the performance of Convolution Neural Networks in the image classification task. A CNN mainly consists of a number of convolutional layers, a pooling layer, and a fully-connected layer. A full CNN architecture can be formed by stacking these layers in order [1]. In a paper, (Banerji and Sinha; 2016) discussed the use of CNN to extract the features. And use all kinds of layers to identify the best of them to be used in the classification of paintings [2].

Another paper proposes a deep learning-based approach to identify the artist of a given fine-

Preprint. Under review.

art painting. The authors used a Convolutional Neural Network (CNN) architecture to extract features from the input images, and then trained a Support Vector Machine (SVM) classifier on the extracted features to identify the artist. The proposed approach achieved promising results on a dataset of 1,888 paintings by 14 artists, achieving an accuracy of 78.3%. Overall, the paper demonstrates the effectiveness of deep learning-based approaches for artist identification in fine-art paintings and provides insights into the design choices and parameters that can affect the performance of such systems. This work can be related to other studies that have used similar techniques for image classification and identification tasks in the domain of art history and cultural heritage [3].

Another paper proposes an image classification of images of paintings where the author's original dataset consists of more than 150,000 images and 2300 different artists and was trying to identify artists of paintings. The author demonstrated the classification by using a Baseline CNN model, ResNet-18 created from scratch and ResNet-18 using transfer learning. The author took the subset of the dataset, which consists of 17100 images across 57 artists. The most accuracy obtained was from the ResNet-18 using the transfer learning model. To increase the accuracy, the author suggested applying data augmentation and increasing the size of the data [4].

3 Dataset

3.1 Overview

In order to train a CNN to identify movement, we first obtain a large dataset of art compiled by Kaggle which is the WikiArt dataset [5]. The dataset contains roughly around 170,000 images of paintings by 3209 artists spanning from a variety of time periods.

Every image is labeled with the artist's name, movement (what the image represents), genre, and style. My aim is to predict the movement for each image.

In this vast dataset, some artist contains less than 100-200 images and some contain more than 7,500 images. Therefore, our dataset consists of images for artists or movements with more than 1000 and less than 5000, and the data is randomly shuffled by 2023. This is ensured the data is not imbalanced

We split the dataset into Training, Validation, and Testing sets using (70,15,15) split per movement and follow the folder structure of PyTorch's ImageFolder.

3.2 Pre-processing & Data Augmentation

3.2.1 Pre-processing

Because the images of painting come in a variety of shapes, sizes, and formats. For pre-processing of images, some images have to be converted into RGB format because every model only supports RBG format.

3.2.2 Data Augmentation

Data augmentation is a technique used in machine learning to artificially increase the size of a dataset by creating new variations of existing data. This is typically done by applying various transformations to the original data, such as rotating, scaling, cropping, or flipping the images. The reasons for using data augmentation:

- Prevent Overfitting
- To address the issue of an imbalanced dataset
- To increase the diversity of data

In my project, for a custom CNN model, the model takes the input of an image size of 128x128 because a larger image file contains more detailed information which can help the model to better distinguish between similar patterns and features. Also, larger images can reduce the blurring which can occur due to resizing or compressing.

4 Experiments

4.1 Setup

All of our models and experiments are implemented in PyTorch [6]. We used to set up the ResNet-50 architecture, and Vgg16 architecture and obtain weights for pre-trained on ImageNet. We used [7] as a guide for replacing the fully-connected layer of models for performing transfer learning on the pre-trained model network. I used [8] as a guide for creating models and training and validation functions.

4.2 Implementation Details

We trained the custom model using the Adam optimizer [9] and also explored using SGD but better results were obtained from the Adam optimizer. Train and Validation accuracy achieved is around 50% but the loss decreased from 1.9 to 1.1 which is higher in Figure 1.

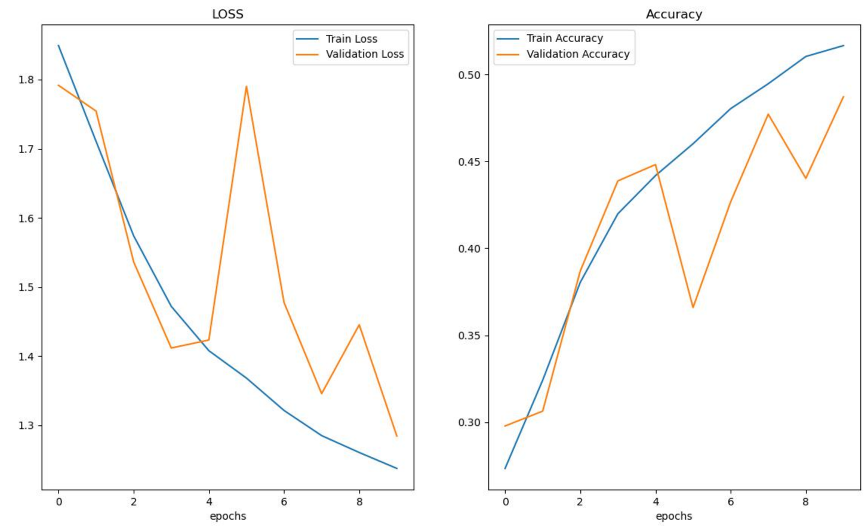

Figure 1: Training and Validation - Loss and Accuracy for custom model

We have not only trained our data on a custom model, but we have also utilized state-of-the-art models such as the pre-trained Vgg16 and ResNet-50 models. In order to adapt the images to these models, we transformed their sizes to a fixed input size of 224x224.

For the pre-trained Vgg16 model, we froze and unfroze the parameters and trained it using the SGD optimizer. As the model was a multi-classification model, we calculated the loss using the CrossEntropyLoss, refer to Figure 2.

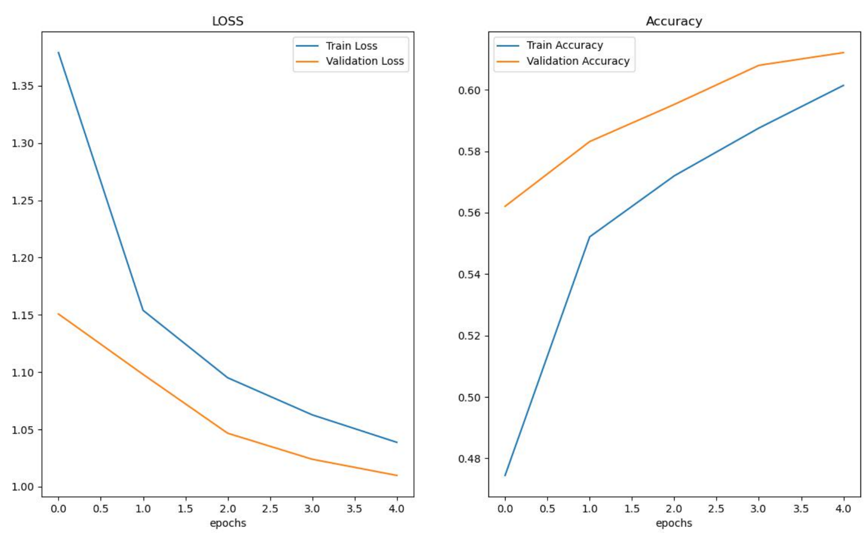

Figure 2: Training and Validation - Loss and Accuracy pre-trained Vgg16 Model (freeze parameters)

From Figure 2, we can see that there is an improvement in accuracy from the previous custom model especially in the validation dataset.

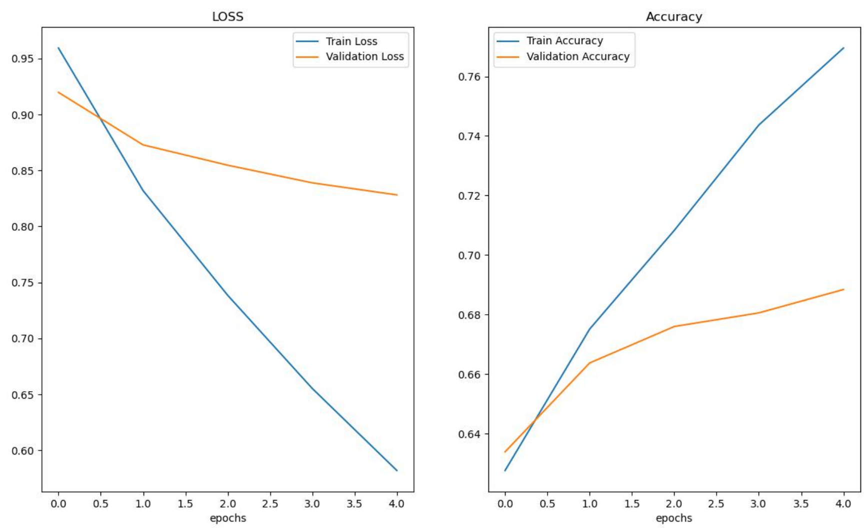

Figure 3: Training and Validation - Loss and Accuracy pre-trained Vgg16 Model (unfreeze parameters)

From Figure 3, we can observe that there is an increase in accuracy in both training and validation and loss is significantly lower than in the previous two models. In Figures 2 and 3, the model was trained on 5 epochs. If trained for higher epochs can achieve better results.

For the pre-trained ResNet-50 model we froze and unfroze the parameters and trained it using the SGD optimizer. As the model was a multi-classification model, we calculated the loss using the CrossEntropyLoss, refer to Figure 4. The performance of the Vgg16 model, Figure 3, is slightly better than ResNet-50 for frozen parameters.

From Figure 5, we can observe that there is ResNet-50 with unfrozen parameter has performed best among other models, with training loss of around 40% and validation loss around 78%. Accuracy has improved significantly to 85% and 73% from the training and validation datasets respectively.

5 Results

Figure 6 shows a confusion matrix, which was calculated using torchmetrics [10] for PyTorch. Each row represents the true artist and each column represents a predicted style.

We can see that for most styles, the diagonal entry is medium blue or dark blue, indicating that most paintings are classified correctly.

One style, second in diagonal, Academicism, is light blue and is less accurately predicted among other styles.

From Figure 7 we can see that overall accuracy is 48%. These findings suggest that our network is able to construct a representation of artistic style, but may encounter difficulties when an artist demonstrates a diverse range of styles or has been influenced by multiple sources.

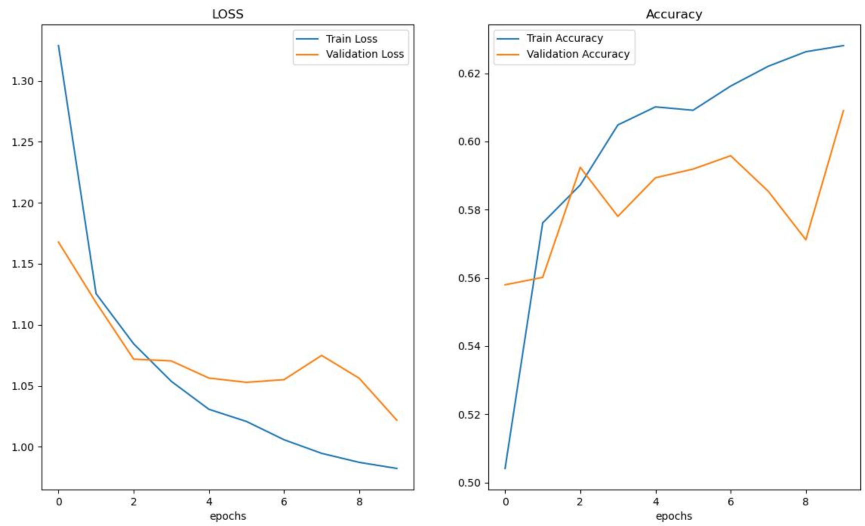

Figure 4: Training and Validation - Loss and Accuracy pre-trained ResNet-50 Model (freeze parameters)

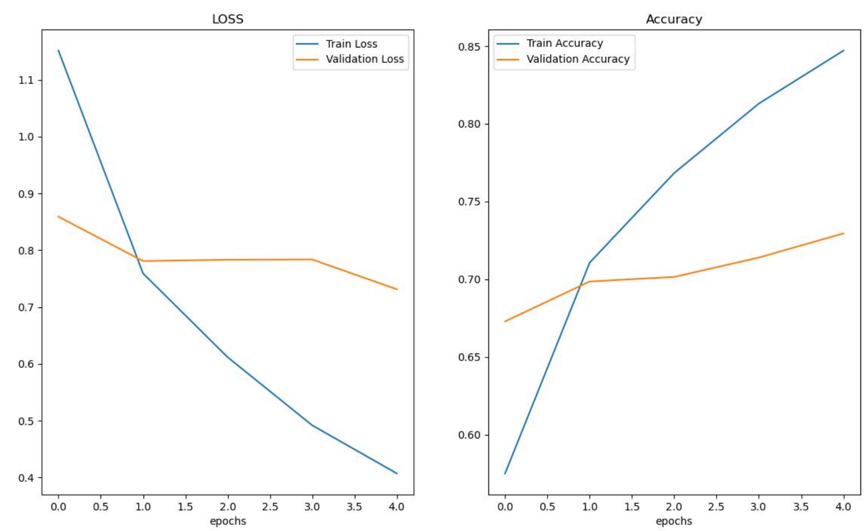

Figure 5: Training and Validation - Loss and Accuracy pre-trained ResNet-50 Model (unfreeze parameters)

6 Conclusion

In our study, we addressed the problem of identifying the artistic style of paintings and attempted to improve the classification accuracy by applying various Convolutional Neural Network (CNN) architectures. Prior work in this field has not explored this problem extensively, making our study unique.

We collected a dataset of approximately 17000 images of paintings from 57 different artists, encompassing a wide range of styles and time periods. We used this dataset to train and evaluate multiple CNN architectures. Our best-performing network was based on the ResNet-18 architecture, which was pre-trained (but unfrozen) on ImageNet with transfer learning. This ResNet-18 model outperformed both the Vgg16 model

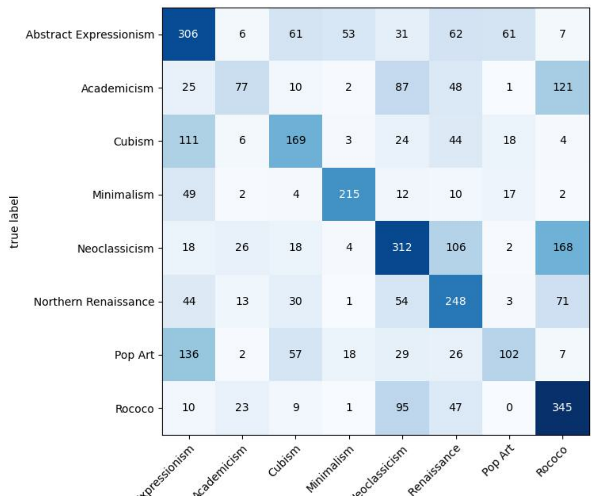

Figure 6: Confusion Matrix of custom model

	precision	recall	f1-score	support
Abstract Expressionism	0.27	0.56	0.36	282
Academicism	0.35	0.44	0.39	301
Cubism	0.73	0.36	0.48	774
Minimalism	0.73	0.67	0.70	341
Neoclassicism	0.54	0.53	0.54	671
Northern Renaissance	0.65	0.42	0.51	711
Pop Art	0.13	0.48	0.20	101
Rococo	0.53	0.57	0.55	492
accuracy			0.48	3673
macro avg	0.49	0.50	0.47	3673
weighted avg	0.57	0.48	0.50	3673

Figure 7: Classification Report of custom model

and a custom CNN model, achieving a significant increase in classification accuracy.

In addition to its superior performance, our ResNet-18 model also created a representation of the style of paintings when asked to predict the style. We conducted various experiments to verify this claim, analyzing the underlying representation and the model's decision-making process. Through these experiments, we confirmed that our network was capable of identifying the unique stylistic features of paintings and making accurate predictions accordingly.

Overall, our study demonstrates the potential of CNN architectures for identifying the artistic style of paintings and highlights the benefits of using pre-trained models and transfer learning techniques for this task.

7 Future Works

For future work, I would like to expand the approach to custom model creation. From all the above figures, the accuracy and loss can be increased and decreased respectively by adding more data and use of data augmentation techniques, and hyperparameter tuning to increase and decrease the accuracy and loss of the models, like normalization, flipping the images vertically or horizontally, cropping the images, and rotating the images.

Some of the images in training, validation, and testing were imbalanced which can be handled by the use of upsampling to increase the weights and downsampling to decrease the weights.

8 References

[1] *"Handwritten Digit Recognition with a Back-Propagation Network"* by Yann LeCun, Bernhard Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne Hubbard, and Lawrence D. Jackel

[2] Banerji, S. and Sinha, A. (2016). *Painting classification using a pre-trained convolutional neural network, International conference on computer vision, graphics, and image processing* , Springer.

[3] *"A Deep Learning Approach to Artist Identification in Fine-Art Paintings"* by Mohsen Hejrati, Radu Babiceanu, and Brian Bailey

[4] *"Artist Identification with Convolutional Neural Networks"* by Nitin Viswanathan, Stanford University

[5] https://www.kaggle.com/datasets/simolopes/wikiart-all-artpieces

[6] https://pytorch.org/vision/stable/index.html

[7] S. Chilamkurthy. http://pytorch.org/tutorials/beginner/transfer - learning tutorial.html

[8] https://www.learnpytorch.io/

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

[10]https://torchmetrics.readthedocs.io/en /stable/classification/confusion_matrix.html