Comparative Analysis of Pre-trained Convolutional Neural Networks and Optimizers for Artistic Style Classification

Baya Lina MENAI
LESIA Laboratory
Mohamed khider University
Biskra, Algeria
bl.menai@univ-biskra.dz

Mohamed Chaouki BABAHENINI
LESIA Laboratory
Mohamed khider University
Biskra, Algeria
mc.babahenini@univ-biskra.dz

***Abstract*—The classification of fine art paintings based on their artistic style is a complex task typically reserved for art experts due to its complexity and reliance on deep artistic knowledge. However, as the digital landscape is flooded with a vast array of digital paintings, there is a growing need to introduce automated methods to assist the art community. In this study, we present a comprehensive framework for evaluating the performance of six pre-trained convolutional neural networks (CNNs): Xception, ResNet50, InceptionV3, InceptionResNetV2, DenseNet121, and EfficientNet B3. Notably, we explore the application of the Xception architecture, a novel approach for this specific task. Furthermore, we examine the impact of three distinct optimizers—SGD, RMSprop, and Adam—coupled with two learning rates (1e-2 and 1e-4) on model performance.**

***Keywords*—Computer vision, Image processing, Convolutional neural network, Style Classification, Optimizers, Transfer learning, and Style Classification.**

I. Introduction

In recent years, researchers have been interested in introducing automatic approaches to the field of fine art painting by using the evolution in computer vision techniques and the great performance of machine learning in the domain of image processing as the number of digital fine art paintings accessible on the web continues to grow exponentially. Many studies automatically investigated the artistic style identification of an artwork during the previous years, which could be grouped into traditional and deep learning approaches. The earliest studies are the traditional approaches, which concentrated on extracting handcrafted features from the images of the artworks and then used a classifier such as a Support Vector Machine (SVM) and K-nearest neighbor (K-NN). Whereas the deep learning approaches are state-of-the-art for painting classification. With the excellent outcomes of convolutional neural networks (CNN) on the largest natural images dataset ImageNet, which contains millions of natural images for image classification, researchers proposed to finetune various CNN architectures for artistic style recognition with transfer learning as the available fine art painting datasets have a small number of labeled images of paintings. Transfer learning is the reuse of an already trained model on a large dataset for a specific task to determine a similar purpose with a small dataset.

In this paper, we propose a framework to compare the performances of six various pre-trained convolutional neural networks (Xception, ResNet50, InceptionV3, IncepResNetV2, DenseNet121, and EfficientNet B3) for identifying the artistic style of a painting by using transfer learning.

The tuning of a pre-trained CNN architecture for a specific task is based on hyper-parameters such as the number of units in each activation layer, the activation function, the number of iterations, the optimizer, and the learning rate. These hyper-parameters are defined before the training, and depending on their configuration, we can get different classification results. Setting up the most appropriate weights for the model can lead to the best classification results. Therefore, the hyper-parameters related to the weights (optimizer and learning rate) must be carefully chosen through experiments, as an inappropriate optimizer might get the network stuck at a local minimum without achieving any improvement toward the global minimum. The optimizer is the function that modifies the attributes of a deep neural network (weights and biases) to minimize the loss function and improve the model's accuracy during training. Moreover, the learning rate determines how fast or slow we approach the optimal weights while respecting the loss function.

In this paper, we also focus on studying the effect of various optimizers (SGD, RMSprop, and Adam) with different learning rates (1e-2 and 1e-4) on the pre-trained models to find the most accurate hyper-parameters for each model.

The paper is organized into four main sections. Section 2 describes our proposed methodology in detail. Section 3 describes the used dataset to evaluate the models, Section 4 reports the experimental results and discussion, and finally, we conclude the paper in Section 5.

II. Methodology

In this work, we aim to concentrate on two points. The first is to propose a framework for style classification of a fine art painting, which is illustrated in Figure 1.

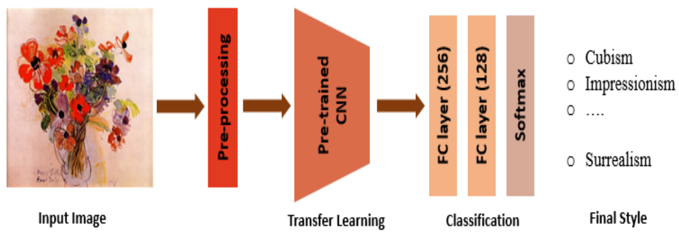

Fig. 1: The proposed framework for style recognition

Our framework consists of two essential parts: the first is the data pre-processing, and the second is feature extraction using transfer learning and classification.

A. Data pre-processing :

Before the training, as the images in our dataset have variant sizes, we resized all the train and test images to a specific size of 480x480 and normalized them. We also applied some data augmentation techniques to the training data by randomly flipping the images horizontally, shifting the width and the height, rotating, and slightly zooming. Furthermore, we used the pre-processing input of each model to avoid overfitting. Figure 2 presents samples of data augmentation techniques applied to a single image.

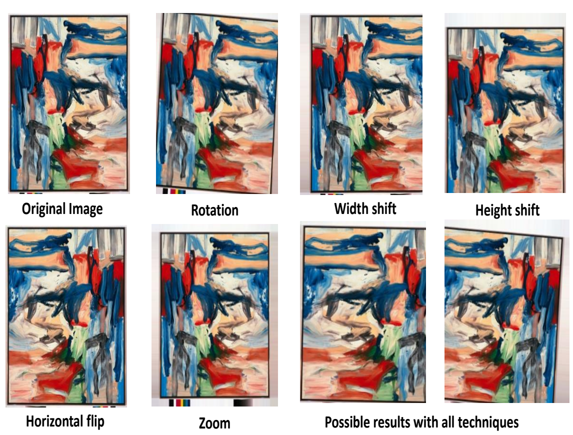

Fig. 2: Samples of data augmentation techniques applied on a single image

B. Feature extraction and Classification:

With the use of transfer learning, we initialized the CNN architectures with weights of the pre-trained ImageNet models rather than recreating the entire training process from scratch. We replaced the last fully connected layers of each architecture that contains 1,000 classes as an output with two dense layers, which have Swish as an activation function with the values of 256 and 128, respectively, followed by a softmax layer with the number of classes in each dataset. These layers are randomly initialized. In addition, to prevent overfitting, we inserted batch normalization and dropout layers after each layer. The output of each model is a probability vector representing the various art-style classes to which the image of the artwork may correspond.

During the training, we applied the finetuning process by unfreezing the last four layers of each model and re-training them besides training the last fully connected layers. The maximum accuracy was considered as the final result after 40 iterations (epochs) of training with a batch size of 64. We ran our experiments using Tensorflow 2.3.0 in Windows 11 with Geforce GTX 1660 Super Intel i9 10900k. All the pre-trained models are from Keras [1]. Figure 3 illustrates the detailed training process.

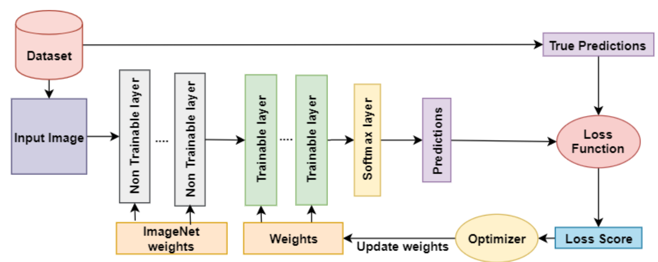

Fig. 3: A schematic illustration of the training process

The second point is to study and compare the effect of different optimizers Stochastic Gradient Descent (SGD) [2], Root Mean Square Propagation (RMSprop) [3], and Adaptive Moment Estimation (Adam) [4] with various learning rates 1e-2 and 1e-4 on the performances of six pre-trained CNN architectures (Xception [5], ResNet50 [6], InceptionV3 [7], InceptionResNetV2 [8], DenseNet121 [9], and EfficientNet B3 [10]) on ImageNet dataset, which has 1.2 million natural images and 1000 classes [11]. Table 1 presents the most important characteristics of each CNN architecture in terms of the input size, depth, the size of the model, and the number of parameters. InceptionResNetV2 is the largest and deepest model we tested in our study.

III. Experimental results

A. Datasets

In our experiments, as we aim to identify the artistic style of a painting, we used the Painting-91 dataset, which includes a total of 4,266 painting images created by 91 artists. They are classified according to the artist and the style. 2,338 paintings have been categorized according to one of 13 artistic styles. These paintings were created by a total of 50 different artists. 1250 of them were utilized for training, while 1088 of them were used for testing. This dataset, created by Khan et al. [12], is one of the most often utilized datasets for classifying artists and styles. Figure 4 shows a few examples from the dataset; each picture has its corresponding style and artist.

TABLE I: The characteristics of CNN architectures

Model	Input Image Size	Depth	Size (MB)	Parameters (Millions)
Xception	299 x 299 x 3	81	88	22.9
ResNet-50	224 x 224 x 3	50	96	25.6
InceptionV3	229 x 229 x 3	48	89	23.9
InceptionResNetV2	229 x 229 x 3	164	213.41	56
DenseNet121	224 x 224 x 3	121	33	7,6
EfficientNetB3	300 x 300 x 3	210	48	12.3

Fig. 4: Samples of Painting-91 dataset

IV. Results and disscussion

The overall accuracy performance of all our experiments on the Painting-91 dataset for artistic style recognition is presented in Table 2. The accuracy is the percentage of successfully identified examples relative to the total number of examples. It is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Where true positive and true negative classification predictions are denoted by TP and TN, respectively, while false positive and false negative classification predictions are denoted by FP and FN, respectively [13]. The percentages in bold represent our best accuracy for identifying the artistic style of a fine art painting for each optimizer with a specific learning rate.

From Table 2, which presents the results of our experiments on the Painting-91 dataset, we can notice that the pre-trained IncepResNetV2 surpassed all the other tested pre-trained CNN architectures with the SGD optimizer with both learning rates (1e-2 and 1e-4). It is also noticeable that the model with the SGD optimizer and a small learning rate of 1e-4 performed poorly and only achieved an accuracy of 24.26%. The accuracy improved by 48.89% to achieve 73.25% with a bigger learning rate equal to 1e-2. In the case of the RMSprop optimizer with a learning rate equal to 1e-2, the IncepResNetV2 achieved the third-best accuracy of 72.06% after ResNet50 and DenseNet121, which achieved 72.15% and 73.99% respectively. Interestingly, in the case of RMSprop and learning rare of 1e-4, the accuracy of IncepResNetV2 increased by 2.94% and achieved the first best accuracy of 75.00% while the accuracy of DenseNet121 decreased by 6.53% and achieved only 67.46%. Similarly, the Adam optimizer with a learning rate equal to 1e-2, the IncepResNetV2 achieved the third best accuracy of 72.89% after ResNet50 and DenseNet121, which achieved 73.07% and 73.99% respectively. Moreover, in the case of Adam and learning rare of 1e-4, the accuracy of IncepResNetV2 increased by 0.18% and achieved the first best accuracy of 75.18% while the accuracy of DenseNet121 decreased by 8.5% and achieved only 65.21%. From the previous results, we can conclude that the best optimizer for each pre-trained model differs from one to another. Additionally, it is crucial to choose an adequate learning rate as the model may fail to achieve good results if an inadequate learning rate is used.

Figure 5 shows the results of all our experiments on the six pre-trained models for style classification with different optimizers: SGD, RMSprop, and Adam, respectively. Each pre-trained model has two bars: the blue bar represents the model's accuracy when trained with a learning rate of 1e-2, and the orange bar indicates the model's accuracy when trained with a learning rate 1e-4. From the figures, we can notice that the pre-trained Xception model performed better than the pre-trained InceptionV3 model. Moreover, the pre-trained ResNet50 model achieved higher accuracy than the pre-trained models Xception, InceptionV3, and EfficientNet B3.

V. Conclusion

In this study, we proposed a framework to compare the performance of six pre-trained CNN architectures (Xception, ResNet50, InceptionV3, InceptionResNetV2, DenseNet121, and EfficientNet B3) for style classification with transfer learning. Furthermore, we studied the effect of different optimizers (SGD, RMSprop, and Adam) with different learning rates of 1e-2 and 1e-4 on each model. In our studies on the classification dataset Painting-91, all the pre-trained models performed poorly with the SGD optimizer and a small learning rate (1e-4). They significantly better performed with a higher learning rate (1e-2). This indicates the impact of choosing the correct learning rate, as the model may fail to achieve good results with inadequate hyperparameters. The results of all the pre-trained CNN models with the RMSprop optimizer and the Adam optimizer show similar results when evaluated with the same learning rate. Both are better than the ones with the SGD optimizer. Moreover, we found that the best-performing optimizer and learning rate for a small model are not always the best hyper-parameters for a more profound and larger model. The pre-trained InceptionResNetV2 was the

TABLE II: The results of style classification on Painting-91

Model \ Optimizer	SGD		RMSprop		Adam	
	1e-2	**1e-4**	**1e-2**	**1e-4**	**1e-2**	**1e-4**
Xception	69.67	18.75	67.10	71.51	69.85	71.32
Resnet50	72.15	22.43	72.15	73.16	73.07	72.24
InceptionV3	69.58	19.12	71.69	68.20	70.96	68.66
Incep-ResNetV2	**73.25**	**24.36**	72.06	**75.00**	72.89	**75.18**
DenseNet121	69.29	15.44	**73.99**	67.46	**73.71**	65.44
EfficientNetB3	68.84	13.24	71.42	70.40	71.78	69.21

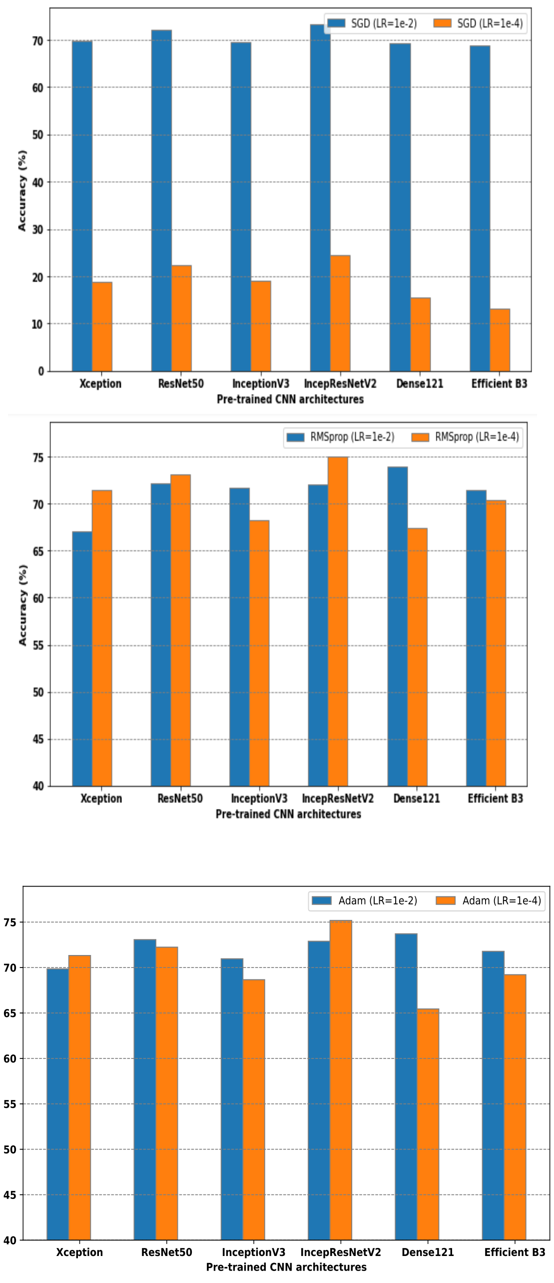

Fig. 5: The results of style classification with SGD, RMSprop, and Adam optimizer on Panting-91 dataset respectively

most accurate model for the artistic style classification on both datasets when it was trained with an Adam optimizer and a learning rate equal to 1e-4.

This article has provided a good foundation for further research, which can be used in future studies to increase artistic style recognition accuracy and decrease the ambiguity between specific styles using more complex and diverse classification techniques.

References

[1] F. Chollet *et al.* (2015) Keras. [Online]. Available: https://github.com/fchollet/keras

[2] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[3] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Cited on*, vol. 14, no. 8, p. 2, 2012.

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

[5] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[7] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[8] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[9] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[10] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[12] F. S. Khan, S. Beigpour, J. Van de Weijer, and M. Felsberg, "Painting-91: a large scale database for computational painting categorization," *Machine vision and applications*, vol. 25, no. 6, pp. 1385–1397, 2014.

[13] Y. Jiao and P. Du, "Performance measures in evaluating machine learning based bioinformatics predictors for classifications," *Quantitative Biology*, vol. 4, no. 4, pp. 320–330, 2016.