Decoding Art using Machine Learning: A CNN-based Approach to Predict Artists from Artworks

Vijayalakshmi Kancharlapalli[1], **Dhruv Vudayagiri**[2], **Rubin Dervishi**[3], **Adrika Datta** [4 1]

[1]220993829; [2]220375472; [3]190336602; [4]220997078

This article discusses deep learning algorithms for distinguishing creative styles. The focus is on computer vision techniques to image classification tasks, namely convolution neural networks (CNN). Effective strategies for enhancing recognition performance are reviewed, including transfer learning, fine-tuning, and augmented data. The article also discusses the significance of global and local traits, as well as the usage of residual networks for artist identification. Finally, several machine learning algorithms are studied, with a focus on the potential of deep learning for automated creative style detection.

artistic style recognition | deep learning | convolutional neural networks | CNN | image classification | transfer learning | fine-tuning | augmented data | residual network | artist identification | VGG 16 | accuracy | hyper parameters

1. Introduction

Art is a human form of expression to describe and illustrate what is being thought and felt (1). Throughout history, there have been multiple evolution's in the representation of artwork. Starting from wall paintings in ancient times, paintings using various tools in the medieval period, and currently using artificial intelligence to generate them (2). But with the evolution of art and its techniques, there has been misinterpretation in detecting the artist for the artwork, and this has become one of the concerns in recognising their work. To overcome this issue, we have a surge of interest in using machine learning techniques where a machine can learn optimally. In traditional machine learning techniques, images are frequently represented by manual-engineered features, which are time-consuming and require topic expertise. Convolutional neural networks (CNNs), on the other hand, may acquire valuable features automatically from raw pixel data, removing the need for manual feature engineering. CNNs can analyse and classify the artworks, and predict the artists for the given artwork.

CNNs are a type of algorithm in deep learning that can understand an image in various ways using patterns in respective images (3). They excel in recognising images as they are able to detect very minor differences in colour, shape, and textures. The model will be trained with an existing data set consisting of the artwork details, it will learn to detect and recognise each artist's unique style and way of their expression.

To train the CNN for recognising the art, we need to collect the images and respective artist details who drew them. It can become a difficult task as we need to make sure the data set is very diverse in terms of region, expression, style, and period when it was being made. For this process, we will use the existing data set which was present online that had all the features mentioned above.

After the model has been trained, we can use the partial data set as an input to assess how well it can identify works of art. The validation data sets accuracy can be used to determine the prediction. It will be helpful for art enthusiasts, archaeologists, and in various other fields so an insight about the artist can be discovered.

Overall, the topic of research predicting the creator based on the art with the use of a CNN is exciting and fast developing. In the upcoming years, technology may advance even farther than it has already, and we may discover even better approaches and algorithms that can be created for analysing the works of art. (4).

2. Literature Review

As stated, artist identification with the artworks has become a difficult task as there are limited resources. But as per the recent developments and research in the image classification sector using various methods it's possible to dive into this problem for resolving it. Going through the recent studies of Convolutional neural network (CNN) architecture in deep learning for image classification it plays a vital role in the identification of the artist based on the art. For example, compared with human performance for higher-5 classification they achieved a residual error of 3.57% on the ImageNet dataset (5). It informs us that we can use them for the artwork. Also, there are various research still undergoing.

Viswanathan's work on training a CNN model using transfer learning for classification to identify the best possible feature representation for artworks (6). The author trained three different CNN architectures using SoftMax classifier with cross-entropy loss. In one of the architectures i.e., baseline which was built from scratch was unable to identify small-level features in the pictures as those details were aggregated. The dataset was consisting of 300 artists with 57 artists and ResNet-18 which was pre-trained on ImageNet combining transfer learning had outperformed regular approaches by a noticeable margin.

According to (7) the authors collected data from Van Gogh and Kroller-Muller museums with 101 good-resolution paintings. They approached for identification of the art using the analysis based on the brushwork. They had quoted that in original paintings itself were faded and a color shift happened due to their age of creation. So, visual inspecting may be useful in the work. Statistical models were interpreted for grouping the arts for the artist. In these, each training image was compared with each other based on the low average distance which indicates the artist's proximity. One of the analysis groups pointed out that contours to be considered in the image, they need to analyse at different scales and local texture reflects the likeness among them. With these studies, they found a method for detecting the unlikeness in brushstrokes texture for art.

As per Peter and Michael (8) aimed to derive an artist profile by merging their works using layers and tools in transparency. J48 algorithm of machine learning was used in classification. Apart from that few statistics were combined like Mathew's correlation and F-measure.

In the study, by (9) Kevin and team used the CNN model and constructed a multilayer pyramid using the image which retrieves 21x more features compared to the single layer itself. During the training of each layer, they found a weighted function that dynamically combines results for the decision. From the methodology, it achieved a promising result for the recall rate for knowing the artist.

From the paper (10) Rui, Huang, and the team proposed a Multi-layer Feature Fusion DenseNet (MFDN) which differs from traditional CNN, it mixes the shallow features along with deep features for checking and plans who drew it. In the lack of data resources, they added transfer learning to solve it and these results were more accurate.

All these studies reveal one common issue related to insufficient data for learning the models. It can also able to figure out that CNNs were very useful in the classification of the artists and also they had given noticeable results, which helps to do further research and build the models using them as a base.

3. Methodology

A. CNN Model Architectures.

A.1 PredictArtist Model. With reference to the architecture sshowed in Figure 1 CNN model was built from scratch to train it and recognise the artists. The model, named PredictArtist, had several layers, to begin with we re-scaled the pixel values for every image in between 0 and 1 using a re-scaling layer. To improve the model's ability to generalise for different image sizes, it had a random zoom layer which does a random zoom for the image.

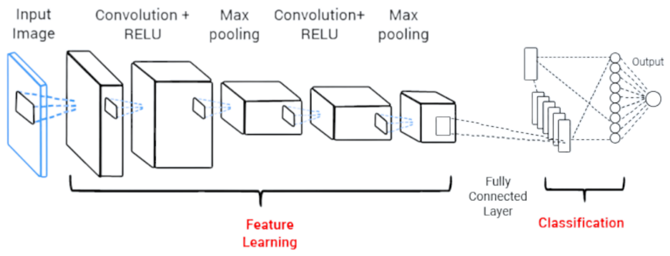


Figure 1. Basic CNN architecture for Image Classification

The initial set of Convolution layer have 32 filters with kernel of size 3x3,which is also known as filter or feature detector it is a small matrix which will be convoluted with input image to generate a feature map which will contain spatially filtered information and Max Pooling layer, it's a layer in CNN which was being considered to reduce the spatial dimensions like width and height of output feature maps along with retaining the useful information, it have 32 filters with kernel size of 3x3. So, these two layers extract the basic features from the input image and reduce the dimensionality of the feature maps. The following convolution layers which

Table 1. Image Size and Layer in the CNN Architecture

Image Size	Layer
224x224	Rescaling
224x224	RandomZoom
222x222	Conv2D
110x110	MaxPooling2D
108x108	Conv2D
53x53	MaxPooling2D
51x51	Conv2D
25x25	MaxPooling2D
23x23	Conv2D
11x11	MaxPooling2D
9x9	Conv2D
4x4	MaxPooling2D

can also be known as hidden layers along with Max Pooling layers have 64,128,256,256 and 512 filters respectively and progressively extracts more complex features from the input image. At every layer as the extraction is done, the image size is converged to finally make a prediction of the class of that image. This behaviour can be seen in Table 1. After the last max pooling layer, the flatten layer converts the 2D feature maps into a 1D feature vector, is then used into a fully connected dense layer with 512 units and ReLU activation function. The ReLU function is a nonlinear function that performs an element-wise operation on the input, converting all negative values to zero and leaving all positive values unaltered. The ReLU function equation is as shown in Equation 1

$$f(x) = \max(0, x) \qquad [1]$$

where x is the input of the activation function and $f(x)$ is its output.(11) Following the dropout layer randomly drops out some of the neurons to reduce over-fitting. Finally, the output layer has a dense layer with a Softmax activation function. It is a sort of activation function utilised in neural networks, notably in the network's output layer. It is used to transform a network's output into a probability distribution across projected output classes. The function converts the input data to 0 to 1 values and normalises them so that their total equals 1. This makes it appropriate for multi-class classification issues in which the aim is to forecast the likelihood of an input belonging to each of the output classes. The Softmax function may be theoretically stated as seen in Equation 2

$$\sigma(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \qquad [2]$$

where x_i is the ith input value, $sigma(x_i)$ is the ith class's output value, and K is the total number of classes. (12) So, it outputs a distribution over the 50 classes. Softmax Loss is another word for categorical cross-entropy loss, which is used to train the model. A Softmax activation combined with a Cross-Entropy loss is used for multiclass classification (13). Using this loss, we can train a Convolutional Neural Network to produce a probability for each picture across the N classes.

In multiclass classification, the raw neural network outputs are processed by the Softmax activation, generating a vector of predicted probabilities across the input classes. Because the labels are one-hot in this specific (and usual) case of multi-class categorization, only the positive class preserves its word in the loss. There is just one non-zero element in the target vector. eliminating zero summation items due to target labels.

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(p_{ij}) \qquad [3]$$

where N is the batch size, C is the class size, y_{ij} is the ground truth label (either 0 or 1) for sample i and class j, and p_{ij} is the projected probability after softmax activation for sample i and class j. The Equation 3 shows the common formulation of the cross-entropy loss function for multi-class classification problems. The equation $y_{ij}\log(p_{ij})$ computes the cross-entropy loss for each sample and class, and the sum is applied to the whole batch of samples and classes. Averaging the result across the batch size yields the ultimate loss value. Along with the combination of Adam optimizer (14) is an alogorithm in deep learning as an extension of stochastic gradient descent algorithm (SGD). To change the learning rate, the moving averages of the gradients and their squared values are used. Overall, this architecture extracts and learns key information from the input image and makes predictions about the author of the artwork using a combination of convolutional, pooling, and fully connected layers.

A.2 Pre-defined Models. In this work, a few neural network models have been tested - using both keras functional API and transfer learning. Evaluation was performed by a subset of the original data kept aside specifically for this purpose. In this section, the model trained from scratch is briefly described followed by the pretrained models.

The keras API sits on top of the TensorFlow API developed by Google for rapid prototyping of ANN models. The image dimensions are set to 224,224,3 (height, width, channels) mostly to conform with the pretrained models used in this work. Downsizing the images by a certain fraction allows for faster training without significantly hurting performance, also to stay within the constraints of compute power allotted to this work. All models are trained on a NVIDIA GeForce RTX 2070 GPU with 8GB memory.

Custom Model Data augmentation is done on the original dataset to aid in learning generalized features owing to the imbalanced instances of the data. Class weights are also used to account for this fact - few artists have several hundred paintings while some others have double digits. In the custom model the architecture is designed as follows: An input layer to feed the images as numpy arrays, followed by the data augmentation layer where flips, zoom out and axis translation are performed. 3 convolutional layers are used alongside 3 maxpooling layers. The convolution layers use (3,3) filters with a stride length of 1. A rectified linear unit or ReLU activation function was used for all layers except the output layer, where softmax was the most appropriate, due to this being a classification problem. Number of filters per layer decreased from 256 to 128 to 64. After the convolutional networks, the output is flattened and fed into a dense layer of 128 nodes, and finally to the output layer of 50 nodes. The model is trained for 25 epochs, with sparse categorical cross entropy loss, and a batch size of 64. A smaller batch size enables faster and more accurate learning while sacrificing compute speed to reach the global minima. The results after training the model are below average, with a 3 - 4 % accuracy suggesting that the model did not learn any meaningful features from the paintings, and making random guesses. A significantly more complex and deeper model is therefore required for this dataset. To this end, transfer learning with several pretrained models are used, owing to the fact that research institutions, google and other large corporations have trained these models with large scale datasets over weeks and months with unmatched capacity. Papers already have the efficacy of transfer learning, and the next section explains this.

A.2.1 VGG 16. VGG16 was developed by computer scientists at the University of Oxford. This model was pre-trained on 14 million images pertaining to 10,000 classes over a few weeks (15). Since this model performed exceptionally well on the ILSVRC-2014 challenge, it is used on this dataset. The architecture includes an input layer where the images are fed at 224,224,3 dimensions, followed by several convolutional layers with (3,3) filters and a stride of 1. Following the convolutional layers, there are 3 dense layers, the first two has 4096 nodes each and the third layer is a 50 node output layer. The output layer of 1000 nodes are excluded from this as the current dataset has 50 classes of data.

A.2.2 ResNet50. Historically, the deeper the neural network was, the better performance it achieved on large scale computer vision tasks. However, due to the partial differentiation operations in the backpropagation step reducing to 0 (the famous problem of vanishing gradient), very deep networks failed to learn meaningful features and hence performance suffered. The residual neural network architecture aimed to fix this problem, and essentially remove the ceiling on how deep CV models can go. (16)

A.2.3 ConvNeXt. Developed by engineers at Facebook research, convNeXt models are pure convolutional ResNet models are improvised and trained in a similar fashion to vision transformers, and can outperform ViT models (17). Owing to its high accuracy in the imagenet data set, this model is trained next on the artwork task. The hyper parameters are kept the same as before, and the dense and output layers added on top.

B. Data

B.1 Synopsis of Data set. Deep learning models are data-hungry and need a large amount of data in order to build the optimal model with high accuracy (18). In addition to the need for a lot of data, it is also crucial to carefully choose the data to make sure that it is aligned to the issue at hand. In compliance with these criteria, the 'Best Artworks of All Time' data set from Kaggle was chosen as the primary data source for building the suggested CNN model (19). This data set has a large collection of paintings from the 50 most significant artists in history, making it an ideal choice for this problem statement. The chosen data set has two key files: one csv file and one folder containing images of paintings.

The csv file includes details about each artist, including their active years, country, the genre of their paintings, how many paintings they have completed, and a brief biography. Preliminary study of the file shows that the majority of artists

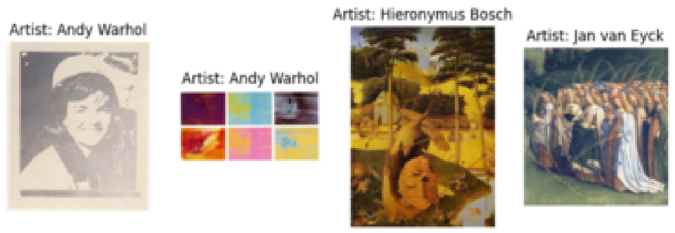


Figure 2. Random Artworks from the dataset

are associated with the French nationalism, while Impressionism is the most common genre. Additional details and plots from this file have been visualised and discussed in the report's supplementary content section. For each artist, there is a sub folder in the 'images' folder that contains images in JPG format of the paintings created by that artist. This structure indicates that each image is labelled with the name of the artist who created it. The data set contains a total of 8446 records across all 50 sub folders, hence 50 classes were examined for the prediction task. However, since there are varying numbers of photographs in each folder, it can be inferred that the data set is unbalanced. Figure ?? shows four images of the artwork of few artists selected at random.

B.2 Data Augmentations. Lack of data can be a major obstacle in the creation of reliable deep learning models. A possible solution for this problem is to use data augmentation methods to enlarge the current data set. By enhancing the data, one can teach the model new features from multiple points of view and improve the model's ability to predict input data. Convolutional neural networks (CNNs) can be used as-is or a

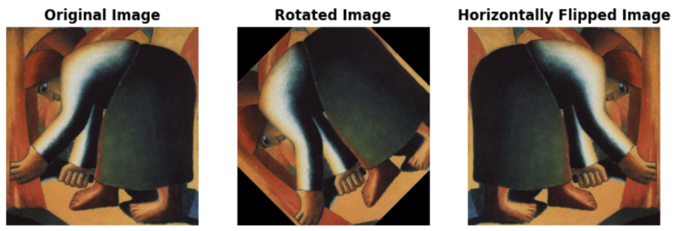


Figure 3. Image Transformations

CNN model can be created from scratch in order to enhance deep learning performance. Data augmentation might be a useful method to improve the model's capacity to extract pertinent features from the input data in both scenarios. (20) This section will examine the augmentations that were applied to both the base CNN model created from scratch and the pre-defined CNNs. We seek to enhance the model's performance and ultimately its predictive power by employing data augmentation strategies.

1. **RandomFlip layer:** This layer will convert the images by flipping either horizontally, vertically, or both (21).
2. **RandomRotation layer:** This was a pre-processing layer. During the training, the given input images were rotated randomly. (22). By default, these were implemented for only training the model.
3. **RandomZoom layer:** To enhance the variability of the input images, this data augmentation technique involves randomly zooming in or out on the picture along each axis independently. The function also fills the space based on the prescribed fill_space parameter. (23).
4. **RandomTranslation layer:** During the training process, the image is randomly translated, while the fill_mode parameter determines how the empty space is filled. (24).

Figure 3 shows the transformations in an image when RandomRotation and RandomFlip(horizontal) are applied.

C. Implementations and Metrics.

C.1 Prepossessing. To prepare the collected artwork images for the CNN model training, pre-processing steps were taken to ensure consistency in image size and format. As the original images in the dataset varied in forms and sizes, each input image was cropped to a size of 224 x 224 pixels. The dataset was then divided into two separate groups for training and validation purposes.

C.2 Hyper-parameters. Hyper-parameters are predefined values that regulate the behaviour of a machine learning model and impact its performance and thus the model's generalisation and convergence speed are also affected ultimately.. Selecting the appropriate hyperparameter configuration is critical, and users can use default values, recommendations from literature or experience, or trial-and-error. Alternatively, hyperparameter tuning strategies can be employed to minimise generalisation error by optimising candidate configurations over a hyperparameter search space using independent test sets or resampling schemes like cross-validation (25). As hyperparameter tuning is such a crucial but complicated step, manual modifications and implementations of hyperparameter tuning have been done in the PredictArtist model. However, setting hyperparameters remains a challenging task that requires expertise (26). Therefore, the tuning has been implemented only on the most controllable and effective hyperparameters. For CNNs, common hyperparameters to tune include the number of neurons, activation function, optimizer, learning rate, batch size, and epochs (27).

C.3 Evaluation Metrics. The accuracy metric, which measures the percentage of correctly categorised photos out of all the images in the test set, is frequently used to gauge how well an image classification model performs. Although accuracy is a good measure of overall performance, it may not be adequate to assess a model's effectiveness when the classes are unbalanced, as they are in our dataset. Precision and F1-score become crucial metrics in these situations since they can be used to evaluate how well the model performs for each class separately. (28). Precision is the percentage of true positives (pictures that were correctly categorised) out of all the photos that the model correctly identified as being in a given class.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \quad [4]$$

The F1-score is a weighted average of precision and recall, where recall counts the number of images that actually belong to a given class out of all true positives(6).

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad [5]$$

Equation 4 and Equation 5 show the numerical formulas used to calculate the precision and F1 score of a class or a model respectively. We can learn more about how well the model is working on each particular class and spot opportunities for improvement by looking at the precision and F1-score for each class. Therefore, when evaluating the effectiveness of an image classification model, it is crucial to take into account a variety of evaluation metrics, such as accuracy, precision, and F1-score.

4. Results and Discussions

A. PredictArtist . This section discusses the numerous runs of the PredictArtist model made, to monitor and understand the model's behavioural tendencies. While assessing the model's performance in terms of accuracy and F1 Scores, a manual modification of a few hyperparameters and data augmentations are carried out in order to precisely examine the behaviour of the model. The hyperparameters that are being investigated are the batch size, data split ratio, number of epochs, dropout rate, number of layers, and regulation layers. As assessing the effect of each parameter on model performance would be a tedious task, many runs have been done to understand the same in batches while modifying a few of them and leaving the others constant. In the initial phase, the model's batch size, split ratio, and dropout rate are tuned. The model's performance has been assessed during this phase using roughly 10 iterations. The batch size was varied between 16,32 and 64, and only two split ratios were examined, which were 70:30 and 80:20. Overall, the model's accuracy during these iterations ranged from 29% to 36%. Even though the model's loss did not result in any substantial changes, the accuracy appeared to improve as the dropout rate rose from 0.1 to 0.3. This reveals that the dropout rate has an immediate and serious impact on the performance of the model. Additionally, it was found that accuracy levels rose when trials were carried out without using early stopping in the model, reaching 40%. This, however, was a result of the model being over-fit. The overall F1 score of the model hovered around 0.017, while the average training accuracy for this set of runs was 67% and the average validation accuracy was 35%. These behaviours have been tabled in Table 2.

Table 2. Input variational ranges and output metric ranges

Hyperparameters	Variational Ranges
Batch size	16, 32, 64
Split ratio	70-30, 80-20
Number of epochs	10, 30, 50
Dropout rate	0.1, 0.2, 0.3
Number of regularisation layers	1 or 2
Augmentations	Flipping, Zooming, Rotation, Resizing
Model complexity	5 conv layers, 4 conv layers, 3 conv layers
Output Metrics	**Ranges**
Training accuracy	31% to 87%
Validation loss	2 to 6.6
Validation accuracy	17% to 54%
Training time	18 min to 40 min

The considerable disparity in accuracy scores between training and validation data-sets also suggests that the model is overfitted. One way used to combat overfitting was to add a regularisation layer. Although the validation accuracy has not changed significantly, there has been a 50% reduction in validation loss. Two regularisation layers were also added in order to better understand the effects of each regularisation, resulting in a validation accuracy of 35.4% and a loss of 2.96. The main takeaway from these iterations is that the gap between training and validation accuracy and loss has shrunk dramatically. Thus, it aids in our understanding that regularisation was added to help with overfitting.

Multiple types of augmentations, either individually or in combination, were added before the model's initial layers in the architecture in order to study the effects of the augmentations on this model. As a result, the metrics saw a little but significant adjustment, lowering the validation loss to 2.0618 and raising the validation accuracy to 42.06%. Multiple runs revealed that too many augmentational layers are ineffective, as the model's accuracy fell to 26.77% when all of the aforementioned augmentation techniques were used. Therefore, it can be concluded that data augmentations were successful—but only when used sparingly.

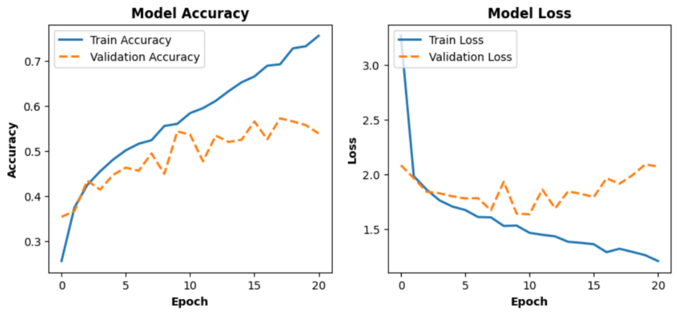


Figure 4. Validation Loss and Accuracy for the best PredictArtist model

After filtering the data, the further iterations were carried out. As was already observed, the data set lacks balance. A few runs were done just taking into account artists with more than 200 works in order to achieve some degree of regularity in the classes. This effort was successful because the model's accuracy increased by up to 54.89%. The plot of this model's performance across epochs is seen in Figure 4. Since these models were developed using the top-performing models from earlier iterations, the F1 score for each class also saw a significant improvement.

B. Predefined. A few deep learning models were trained and tested as part of the solution to the Artwork challenge. The baseline model was a 5 layer convolutional neural network model - an input layer followed by 3 sets of convolutional layers and max pooling layers of size (3, 3), then a flatten layer and a Dense layer. The output layer has 50 nodes corresponding to 50 classes (artists). This model was trained for 30 epochs, with a batch size of 64. The model ended with a 1% test set accuracy signifying no learning was possible. Transfer learning was used in the subsequent solutions with three deep pretrained models: VGG16, ResNet50, ConvNeXt xLarge.

Two versions of VGG16 were trained, one with random data augmentation applied and one without. Both models

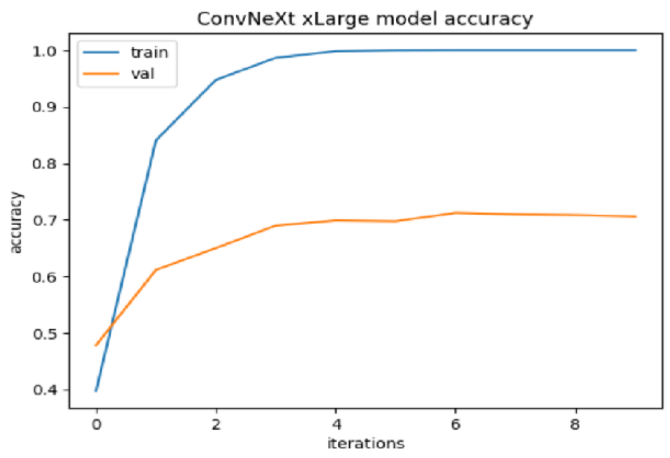


Figure 5. Accuracy plot of ConvNeXt model

performed more or less equally at 63% accuracy, while the model with data augmentation performed slightly better at 64.7%. A much deeper model, the residual neural network (ResNet50) was used. After 10 epochs, a train set accuracy of 73.9% was achieved while on the test set the model performed poorly at 16.3% accuracy indicating severe over-fitting. The final and best performing model was the ConvNeXt xLarge model released by Facebook AI Research. This model achieved a test set accuracy of 74.4% after 10 epochs, although a train set accuracy of 100% indicates a fairly high degree of over fitting.

Table 3. Model performance on train and test sets

Model	Train set accuracy	Train set loss	Test set accuracy	Test set loss
VGG16 (with augmentation)	100%	0.0018	63%	1.76
VGG16 (without augmentation)	100%	0.0011	64%	1.76
Resnet 50	73.9%	0.6303	16.3%	5.28
ConvNext	100%	6.2781e-04	74.4%	1.10

5. Conclusion

In conclusion, the goal of this study was to investigate the application of deep learning methods to the classification of works of art. The suggested method classified artworks by their respective artists using convolutional neural networks, more specifically the ResNet50 architecture. The findings of the experimentation indicate that data augmentation methods such random flipping, rotation, zooming, and translation can greatly increase the model's accuracy. In comparison to the initial accuracy of 51.0% obtained without the use of data augmentation, the final model's accuracy of 63.5% is a significant improvement. In this study, paintings by the top 50 artists in history were also examined using the Kaggle data set known as the Best Artworks of All Time. The data set was determined to be the best option for this problem statement since it gave the model access to a wide variety of artworks from which to learn. Additionally, it was discovered that the data set was out of balance, which emphasises the significance of thorough data selection and augmentation to guarantee that the model is trained on a broad and representative sample. Overall, the experiment shows how deep learning techniques may be used to classify works of art and emphasises how important data selection and augmentation are to getting the best results. The results of this study might have an impact on the art market since automated art classification could boost the effectiveness and precision of art cataloguing and authentication. The use of additional deep learning architectures and data augmentation methods, as well as the application of these methods to other tasks involving art, may be explored in further study in this area.

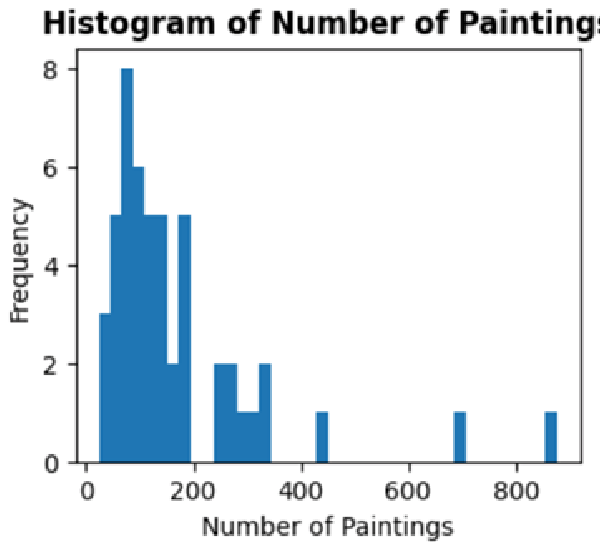
References

1. J Hospers, The concept of artistic expression. *Proc. Aristot. Soc.* **55**, 313–344 (1954).
2. B Agüera y Arcas, Art in the age of machine intelligence. *Arts* **6** (2017).
3. S Hijazi, R Kumar, C Rowen, , et al., Using convolutional neural networks for image recognition. *Cadence Des. Syst. Inc.: San Jose, CA, USA* **9**, 1 (2015).
4. J DuBois, Using convolutional neural networks to classify art genre. (2022).
5. K He, X Zhang, S Ren, J Sun, Deep residual learning for image recognition in *Proceedings of the IEEE conference on computer vision and pattern recognition.* pp. 770–778 (2016).
6. N Viswanathan, Artist identification with convolutional neural networks. *Stanf. Univ.* (2017).
7. CR Johnson, et al., Image processing for artist identification. *IEEE Signal Process. Mag.* **25**, 37–48 (2008).
8. P Stanchev, M Kolinski, Novel artist identification approach through digital image analysis using machine learning and merged images in *Information Technology and Systems: Proceedings of ICITS 2019.* (Springer), pp. 465–471 (2019).
9. KA Jangtjik, MC Yeh, KL Hua, Artist-based classification via deep learning with multi-scale weighted pooling in *Proceedings of the 24th ACM international conference on Multimedia.* pp. 635–639 (2016).
10. R Bai, H Ling, Z Kai, D Qi, Q Wang, Author recognition of fine-art paintings in *2019 Chinese Control Conference (CCC).* (IEEE), pp. 8513–8518 (2019).
11. AK Dubey, V Jain, Comparative study of convolution neural network's relu and leaky relu activation functions in *Applications of Computing, Automation and Wireless Systems in Electrical Engineering: Proceedings of MARC 2018.* (Springer), pp. 873–880 (2019).
12. HA Almurieb, ES Bhaya, Softmax neural best approximation in *IOP Conference Series: Materials Science and Engineering.* (IOP Publishing), Vol. 871, p. 012040 (2020).
13. V Labs, Cross-entropy loss: A comprehensive guide (2021).
14. J Brownlee, Adam optimization algorithm for deep learning (https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/) (2017) Accessed on: 2023-04-14.
15. K Simonyan, A Zisserman, Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
16. K He, X Zhang, S Ren, J Sun, Identity mappings in deep residual networks in *Computer Vision–ECCV 2016.* (Springer), pp. 630–645 (2016).
17. Z Liu, et al., A convnet for the 2020s in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* pp. 11976–11986 (2022).
18. DataToBiz, Datasets in machine learning (https://www.datatobiz.com/blog/datasets-in-machine-learning/) (Accessed on April 28, 2023).
19. Ikarus777, Best artworks of all time (https://www.kaggle.com/ikarus777/best-artworks-of-all-time) (2021) Accessed on April 28, 2023.
20. C Shorten, TM Khoshgoftaar, A survey on image data augmentation for deep learning. *J. Big Data* **6**, 60 (2019).
21. Keras, Keras preprocessing layers: Randomflip (https://keras.io/api/layers/preprocessing_layers/image_preprocessing/random_flip/) (year?).
22. Keras, Keras preprocessing layers: Randomrotation (https://keras.io/api/layers/preprocessing_layers/image_augmentation/random_rotation/) (year?).
23. Keras, Keras preprocessing layers: Randomzoom (https://keras.io/api/layers/preprocessing_layers/image_preprocessing/random_zoom/) (year?).
24. Keras, Keras preprocessing layers: Randomtranslation (https://keras.io/api/layers/preprocessing_layers/image_preprocessing/random_translation/) (year?).
25. P Probst, AL Boulesteix, B Bischl, Tunability: Importance of hyperparameters of machine learning algorithms. *J. Mach. Learn. Res.* **20**, 1–32 (2019).
26. LN Smith, A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820* (2018) Accessed on April 28, 2023.
27. A Vidhya, Tuning the hyperparameters and layers of neural network – deep learning. *Anal. Vidhya* (2021) Accessed on April 28, 2023.
28. R Today, Learn precision, recall and f1 score of multiclass classification in depth (https://regenerativetoday.com/learn-precision-recall-and-f1-score-of-multiclass-classification-in-depth/) (accessed 28 Apr. 2023).

APPENDIX A: Analysis of the csv file

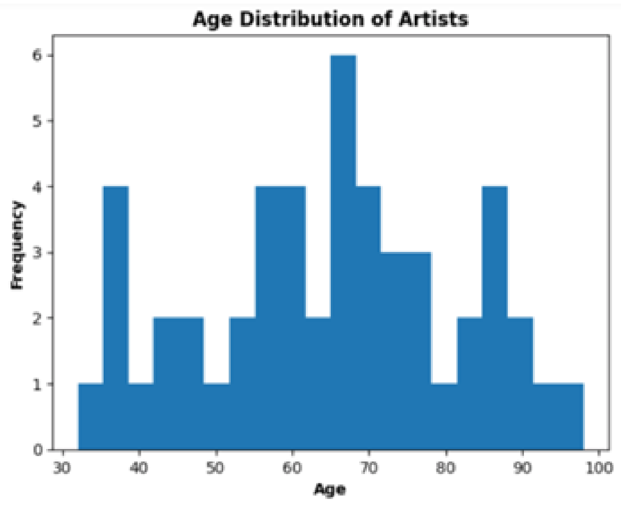
The following observations were made from the csv file containing the details of the 50 artists.

The graph plots the frequency of paintings against the number of paintings on the x-axis. The frequency initially peaked at 8 for the range of 0-200 paintings. However, as the number of paintings increased to the range of 200-600, the frequency dropped significantly to a peak of only 2. This trend suggests that the majority of artists in the dataset have created a relatively small number of paintings, with a significant drop-off in frequency once the number of paintings exceeds 200.

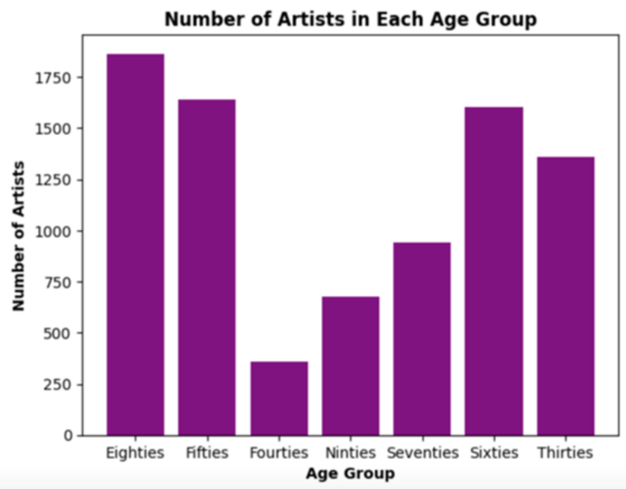


Based on the analysis of the age distribution of artists in the dataset, it was found that the highest frequency occurred between the age range of 65 to 70 years, with a peak frequency of 5. Additionally, a peak frequency of 4 was observed between the age range of 30 to 40 years. The remaining age groups were distributed differently, with lower frequencies observed. These findings suggest that there may be a correlation between an artist's age and their likelihood of producing artwork that is recognized as one of the "best of all time". Further investigation into this correlation could provide valuable insights into the artistic development and trajectory of successful artists. The age group axis is divided into eight categories, namely eighties, fifties, forties, nineties, seventies, sixties, and thirties. The graph shows that the highest number of artists falls in the age group of the eighties, with a count of 1800. This is followed closely by the age group of the fifties, with a count of 1700. The age group of the sixties has the third-highest number of artists, with a count of 1600. The age group of the thirties has the lowest count of artists, with a count of 1300. The age groups of the forties, seventies, and nineties have counts of 400, 1000, and 720 artists, respectively. These results suggest that most of the significant artists in the dataset belong to the age group of the eighties and fifties.

The distribution of artists by age group in the dataset provides interesting insights into the art world. It is clear from the graph that a significant portion of the artists represented in the dataset belong to the older age groups, with the highest

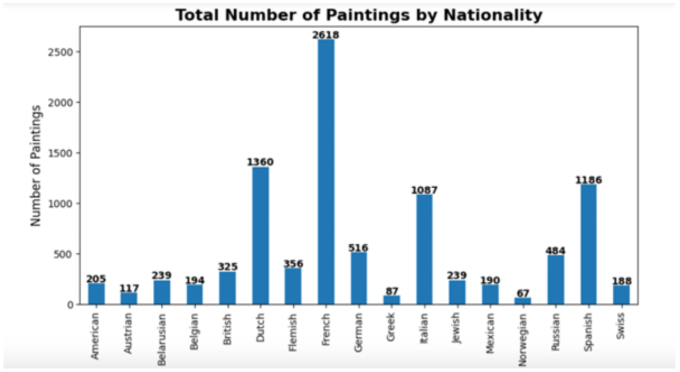


number of artists belonging to the 80s and 50s age groups. This could indicate that artists tend to reach their peak in terms of recognition and impact in their later years, as they accumulate more experience and expertise. It is also interesting to note that the 90s age group has a relatively high number of artists, which may reflect the recent emergence of new talent in the art world. However, the low number of artists in the 40s age group suggests that there may have been a gap in the artistic production during this period, which could be a subject for further investigation. Overall, the analysis of the age group distribution in the dataset highlights the importance of understanding the historical context and cultural factors that shape artistic production and recognition.

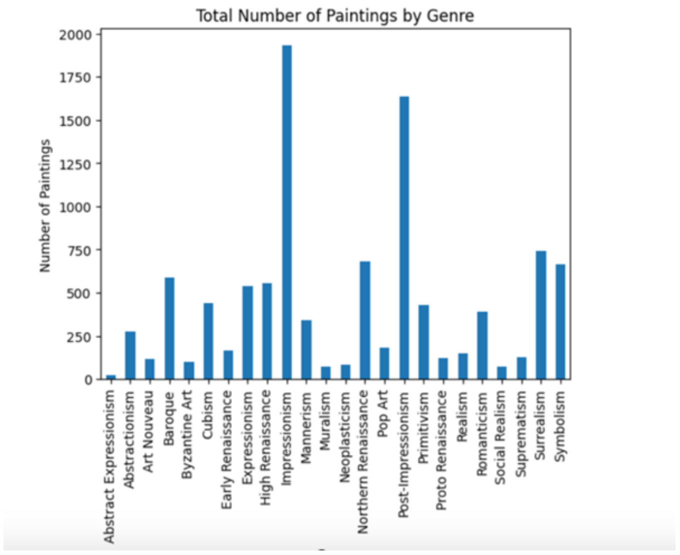


This graph shows the distribution of the number of paintings according to the nationality of the artist. The graph indicates that there is a wide range of nationalities represented in the dataset, with the majority of artists being Dutch, French, Italian, and Spanish. The high number of Dutch and Flemish artists can be attributed to the significant contribution of these regions to the development of Western art during the Renaissance period. The relatively large number of French artists is also expected, given France's rich artistic heritage and history. It is interesting to note that while the number of American artists is relatively small compared to some European nations, it is still significant, suggesting that American art has had an impact on the art world. The graph also shows that some nations, such as Norwegian and Greek, are underrepresented in the dataset. This could be due to factors such as cultural differences or a lack of representation in the art world. Overall, the graph highlights the diversity of the artists in the dataset and the international nature of the art world.

The nationality distribution of artists in the dataset shows a significant presence of French, Italian, and Dutch painters. French painters make up the largest group with 2,618 paintings, followed by Italian painters with 1,087 paintings, and Dutch painters with 1,360 paintings. Other notable nationalities with a significant presence include Spanish painters with 1,186 paintings, German painters with 516 paintings, and Flemish painters with 356 paintings.



On the other hand, there are some nationalities that are represented by a relatively small number of paintings, such as Norwegian painters with only 67 paintings and Greek painters with only 87 paintings. This distribution may reflect the historical context of art and the influence of different regions on the development of art styles over time. It may also suggest the possibility of biases in the dataset, such as the availability of paintings from certain regions or the representation of certain art movements.



The graph represents the number of paintings in different genres. Impressionism, with a total of 1975 paintings, has the highest frequency, followed by post-impressionism with 1625 paintings, and Northern Renaissance with 730 paintings. It is notable that some genres such as social realism, muralism, and neoplasticism have relatively low frequencies, indicating their limited popularity among artists.

The Renaissance period is characterized by a renewed interest in classical art and culture, leading to the emergence of two dominant styles, the high Renaissance and the early Renaissance. The former is characterized by a focus on balance, symmetry, and realistic representation, while the latter is known for its more naturalistic approach, often depicting everyday life. Both styles are well-represented in the graph, with a total of 800 paintings.

Another significant genre in the graph is Baroque, with a frequency of 625. This period is marked by the use of dramatic lighting, intense emotions, and elaborate ornamentation. Byzantine art, which originated in the Byzantine Empire, is characterized by its use of rich colors, complex symbolism, and decorative motifs, and has a frequency of 125.

APPENDIX B: Links to the code and data files

The codes and data folder can be accessed using the following hyperlinks.

Codes folder

Data folder