European Journal of Computer Science and Information Technology, 14(8),36-48, 2026
Print ISSN: 2054-0957 (Print)
Online ISSN: 2054-0965 (Online)
Website: https://www.eajournals.org/
Publication of the European Centre for Research Training and Development -UK

Evaluating VGG16 and Mobilenetv2 for Automatic Recognition of Art Periods Using Transfer Learning

Erarda Vuka, Jurgen Mecaj, Olgerta Idrizi
Mediterranean University of Albania, Tirana, Albania
erarda.vuka@umsh.edu.al, jurgen.mecaj@umsh.edu.al, olgerta.idirizi@umsh.edu.al,

doi: https://doi.org/10.37745/ejcsit.2013/vol14n83648 Published November 14, 2026

Citation: Vuka E., Mecaj J., Idrizi O. (2026) Evaluating VGG16 and Mobilenetv2 for Automatic Recognition of Art Periods Using Transfer Learning, *European Journal of Computer Science and Information Technology*, 14(8),36-48

Abstract: *Artificial intelligence offers new ways to organize and share cultural heritage collections. This paper evaluates its potential for automatically classifying paintings by art period, as a tool for the digitization and accessibility of art-related information. Two convolutional neural network (CNN) models, VGG16 and MobileNetV2, were adapted through transfer learning from ImageNet-pretrained weights and trained on the WikiArt dataset, which contains 81,444 paintings across 27 art periods. Class imbalance was addressed with data augmentation and class weights, and the data were split into stratified training, validation and test sets (70/15/15). VGG16 achieved the best performance, with an AUC of 0.966, an accuracy of 0.763 and an F1-score of 0.701, compared with 0.946, 0.703 and 0.699 for MobileNetV2. MobileNetV2, however, trained faster and used less memory. The results reveal a trade-off between accuracy and efficiency that can guide the choice of model for galleries, museums and educational platforms.*

Keywords: computer vision, convolutional neural networks, VGG16, MobileNetV2, transfer learning, art period classification

INTRODUCTION

This paper focuses on the main concepts of artificial intelligence, Computer Vision and convolutional neural networks. Also, in this paper the VGG16 and MobileNetV2 models will be treated, as well as the way in which they are used for image classification. The aim of this study is to present a valid method for the use of artificial intelligence as an aid tool in the exhibition and better understanding of art. The study focuses on the way in which an AI model can help in identifying the artistic periods of paintings. Through this classification, it is intended that information about works of art becomes more accessible to the public.

One of the fields where artificial intelligence has had the most impact is the field of art. The use of advanced algorithms has opened many different doors in this field, starting from the analysis and classification of works and up to their generation. However, the integration of artificial intelligence into art has prompted numerous discussions regarding its impact on the creative process and on the role of the artist. Beyond these debates, artificial intelligence presents considerable potential as an aid

tool for the preservation, organization and promotion of artistic heritage in digital environments (Cameron and Kenderdine, 2007).

Within the framework of this study, artificial intelligence is treated as an aid tool which can contribute to the digitization and increased accessibility of artistic information. By using image classification models, the automatic identification of the artistic periods of paintings becomes possible, creating new opportunities for the organization of digital collections and for providing educational information to the public. Research on painting classification shows that computational methods can be used to analyze visual characteristics and to predict elements such as the style, genre or author of the work (Saleh and Elgammal, 2016). For this reason, this paper focuses on the analysis and comparison of artificial intelligence models for the classification of artistic periods and on the evaluation of their potential for use in art galleries and cultural digitization systems.

Digital technologies have been seen as tools that can enrich the experience of visitors and offer more flexible forms of interaction and learning in museums (Tallon and Walker, 2008). Therefore, the need arises for the use of technological methods that can help in the analysis and classification of works of art. Artificial intelligence can be used to identify the artistic period of a painting and to help in presenting appropriate information to the public. Studies on AI and art show that artificial intelligence can be used in the analysis of digitized art collections, the classification of works, similarity search and the visual understanding of art (Cetinić and She, 2022).

LITERATURE AND THEORETICAL UNDERPINNING

COMPUTER VISION IN PAINTING ANALYSIS

In recent decades, computer vision has seen important developments. This field is used in many practical applications, such as object recognition, medical image analysis, face recognition and image classification. This development of recent decades has enabled us to create this study, as it has opened an important door for us in image analysis. As we mentioned, the most important element of computer vision is the extraction of visual features. These features can be related to contours, texture, colors, shapes, lighting and the position of objects. In traditional systems, researchers or developers often defined these characteristics manually. With the development of modern methods of machine learning and deep learning, many of these features can be learned automatically by the model itself during the training process (Goodfellow, Bengio and Courville, 2016; Janiesch, Zschech and Heinrich, 2021; LeCun, Bengio and Hinton, 2015).

The development of deep learning has greatly influenced the progress of computer vision. Models with many layers are able to learn different representations of visual data. The first layers can learn simple elements such as lines, angles and basic shapes. Deeper layers can learn more complex structures related to objects, patterns and more general content of the image. This makes deep learning especially important for tasks that require automatic image analysis (Goodfellow et al., 2016; LeCun et al., 2015). Computer vision is used in many practical fields. It is used in object recognition, medical image analysis, security systems, face recognition, autonomous vehicles, robotics and visual search. In each of these cases, the goal is for the computer system to obtain information from an image or video and use it to perform a specific task. This shows that computer vision is not only a theoretical concept, but also a field with wide use in modern technologies (LeCun et al., 2015; Szeliski, 2022). Another important aspect of computer vision is the ability to automate processes that previously required human intervention. For example, a system can analyze a large number of images in a much shorter time than

a person. This is useful in cases where the amount of visual data is very large and where fast organization, search or interpretation of information is required (Szeliski, 2022).

In general, computer vision aims to create systems that can understand visual content in a way as close as possible to the way humans interpret the world through sight. However, this process is not simple, because images can vary in lighting, viewing angle, quality, colors and the presence of different visual elements. For this reason, computer models must be able to learn from many examples and to recognize stable patterns within visual data (Goodfellow et al., 2016; Szeliski, 2022).

In this paper, computer vision is related to the analysis of images of paintings and to their use to help make information about art easier to access. Paintings contain many visual elements, such as color, texture, form, composition and the use of light. Computer models to help in organizing and interpreting digitized art collections can process these elements. Saleh and Elgammal (2016) show that computational methods can be used to analyze works of art and to predict elements such as style, genre and author. This makes computer vision an important foundation for the project of this paper (Saleh and Elgammal, 2016; Szeliski, 2022).

CONVOLUTIONAL NEURAL NETWORKS

Unlike traditional neural networks, CNNs are built in a way that takes advantage of the spatial structure of images. An image is not simply a set of numerical values. It has order, shape and relationships between pixels. For this reason, it is important for the model to analyze not only the pixels individually, but also the way in which these pixels are related to one another (Goodfellow et al., 2016).

CNNs use convolutional filters to analyze small parts of the image. These filters move over the image and look for certain visual features. In the first layers, the model can identify simple elements such as edges, lines and basic shapes. In deeper layers, the model can learn more complex elements such as textures, structures and more complicated shapes (Goodfellow et al., 2016; LeCun et al., 2015).

This way of functioning makes CNNs very suitable for the classification of paintings. Paintings contain many visual elements that can help in distinguishing artistic periods. These elements include colors, textures, composition and the use of light. A CNN model can learn these characteristics during the training process and then use them to predict the class of a new painting. Studies on painting classification show that computational methods can be used to extract visual features and to predict elements such as the style, genre or artist of the work (Saleh and Elgammal, 2016).

VGG16

VGG16 is a CNN architecture proposed by Karen Simonyan and Andrew Zisserman of the Visual Geometry Group at the University of Oxford. This model was introduced in the paper *Very Deep Convolutional Networks for Large-Scale Image Recognition*, published at ICLR in 2015. The main aim of this paper was to analyze the effect of the depth of convolutional networks on the performance of large-scale image classification. The authors showed that increasing the depth of the network, using small 3×3 filters, can improve performance, and that the best configurations reach 16 to 19 weight layers (Simonyan and Zisserman, 2015).

The main idea of VGG16 is the use of small filters of size 3×3, instead of larger filters in the first layers. This approach allows the model to increase the depth of the network and to learn more complex features, while keeping a simple architectural structure. Using several 3×3 layers one after another

helps increase the receptive field and adds non-linearity to the model, without making the architecture too complicated (Simonyan and Zisserman, 2015).

When we mention the depth of a neural network, we are referring to the number of layers through which information passes from the moment the image is fed in as input until the moment the model produces the result. A deep network contains many layers that analyze the data at different levels of abstraction. The first layers usually learn simple features, while deeper layers learn more complex features. This is related to the principle of deep learning, where models with many layers learn representations of data at different levels of abstraction (Goodfellow et al., 2016; LeCun et al., 2015).

VGG16 is often used in transfer learning, because it can be loaded with weights pretrained on large image datasets, such as ImageNet. In this case, the convolutional part of the model can be used to extract visual features, while the final layers can be adapted to the classes of the new dataset. Keras provides VGG16 with pretrained weights and states that Keras Applications models can be used for prediction, feature extraction and fine-tuning (Keras, n.d.).

MOBILENETV2

MobileNetV2 is a convolutional neural network architecture proposed by Sandler, Howard, Zhu, Zhmoginov and Chen in the paper *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. This model was published in 2018 and was developed with the aim of performing various computer vision tasks, such as image classification, object detection, etc. The authors present MobileNetV2 as an architecture built for environments where processing efficiency and lower memory usage are required (Sandler, Howard, Zhu, Zhmoginov and Chen, 2018).

One of the main ideas that this model presents is the use of what are called inverted residual blocks. In this structure, the input and output of the block are in narrow layers, called bottleneck layers, while the middle part of the block is expanded to process more information. So, the model first expands the number of channels and then applies depthwise convolution. Finally, the information is projected into a narrower representation (Sandler et al., 2018).

ARCHITECTURAL DIFFERENCES BETWEEN VGG16 AND MOBILENETV2

Both VGG16 and MobileNetV2 are CNN-based models, however, they both operate differently. VGG16 relies on a deep and regular architecture based on layers composed of 3x3 filters, followed by max pooling layers and dense layers (Simonyan and Zisserman, 2015). MobileNetV2 relies on inverted residual blocks and linear bottlenecks for efficient processing of visual information (Sandler et al., 2018). One important difference between VGG16 and MobileNetV2 is their architectural differences. VGG16 follows a more conventional design, with an increasing number of filters in deeper layers of the image. Its role is to extract visual features through its convolutional layers and perform classification through its dense layers. MobileNetV2 relies on architectural blocks that contain information expansion, processing, and projection back into a thinner representation (Sandler et al., 2018). Another difference is the complexity of the model. VGG16 is a complex model due to its depth. VGG16 is composed of a large amount of parameters as seen in the Keras documentation, 138.4 million, and a large size of 528 MB (Keras, n.d.). In contrast, MobileNetV2 was developed for efficient processing and memory usage (Sandler et al., 2018). The main features of each architecture as seen in the original papers and Keras documentation are summarized in Table 1 (Keras, n.d.; Sandler et al., 2018; Simonyan and Zisserman, 2015).

Table 1.Main characteristics of the VGG16 and MobileNetV2 architectures.

Characteristic	VGG16	MobileNetV2
Year of publication	2015	2018
Main building block	Successive 3×3 convolutions	Inverted residual blocks with linear bottlenecks
Convolution type	Standard	Depthwise separable
Number of parameters (millions)	138.4	3.5
Model size (MB)	528	14
Depth*	16	105
Input size (pixels)	224×224	224×224
Top-1 accuracy on ImageNet (%)	71.3	71.3

* Topological depth as reported in the Keras documentation, which also counts activation and normalization layers.

Both models are suitable for transfer learning, which shows us that both are more than applicable for our study. Through this approach, the models can use the knowledge gained from large image datasets and then be adapted to the specific task of classifying paintings according to artistic periods (Pan and Yang, 2010; Yosinski et al., 2014; Zhao et al., 2021).

METHODOLOGY

THE WIKIART DATASET

The dataset that we have selected for image classification is the dataset called WikiArt (WikiArt, n.d.), which is available on Kaggle. This dataset is known for the variety of paintings it contains from different periods of art. This dataset is also large, with 81,444 paintings divided into 27 different and unique categories.

After identifying all the categories present in the dataset, it is important to identify the population of these categories. The population of the categories can cause problems in the training process if measures are not taken to balance the dataset. To present the population of the classes, in order to identify whether the dataset is balanced or not, we will present it in the form of a table, “Population of the categories of the WikiArt dataset”:

Table 2.Classes of the dataset and their population.

Index	Art period	Number of paintings
1	Impressionism	13060
2	Realism	10733
3	Romanticism	7019
4	Expressionism	6736
5	Post Impressionism	6450
6	Symbolism	4528
7	Art Nouveau Modern	4334
8	Baroque	4240
9	Abstract Expressionism	2782
10	Northern Renaissance	2552
11	Naive Art Primitivism	2405
12	Cubism	2235
13	Rococo	2089
14	Color Field Painting	1615
15	Pop Art	1483
16	Early Renaissance	1391
17	High Renaissance	1343
18	Minimalism	1337
19	Mannerism Late Renaissance	1279
20	Ukiyo e	1167
21	Fauvism	934
22	Pointillism	513
23	Contemporary Realism	481
24	New Realism	314
25	Synthetic Cubism	216
26	Analytical Cubism	110
27	Action Painting	98

From the table we can see that the WikiArt dataset, despite being a diverse dataset, is not balanced, and this can cause problems in the accuracy of our models. Therefore, we will focus on trying to resolve this problem, so that our models have a satisfactory accuracy.

HANDLING CLASS IMBALANCE

One method that we have available is resampling. This method consists of adding materials (oversampling) to the categories that have fewer elements, or removing materials (undersampling) from the categories that have more elements. In this study we will avoid the undersampling method, since we would have to delete many elements. Although this method could bring faster execution and lower memory usage, it would cause a very high loss of data, which would affect the accuracy of the model.

Oversampling is a valid option for this study. To perform oversampling, we will need to add more materials to the categories that have fewer elements. If we choose to duplicate the images, then this can cause overfitting in the model.

IMAGE PREPROCESSING AND AUGMENTATION

Before the data are fed into the model, it is important that they have the same size. This is necessary because CNN-based models require all inputs to be of the same size. All images were resized so that they had a size of 224×224 pixels. Another important step is the normalization of pixel values. In their original form, pixel values range from 0 to 255. These values were transformed into the range 0 to 1 using the rescale function. This normalization helps the model to be trained in a more consistent and efficient way, since smaller numbers are simpler to process.

Image augmentation was applied only to the training set and not to the validation and test sets. These two sets were only resized and normalized. This is done because it is important that the information in these two sets is as close as possible to real data. If this process were also applied to these two sets, the evaluation of the model would become less objective. Another important part of data balancing is class weights. To identify the class weights, it is important to carry out the appropriate calculation, in order to identify which class needs a greater weight and more attention from the model.

TRAINING, VALIDATION AND TEST SPLIT

After preparing the dataset, the next step is splitting the data into separate groups. This step is very important for building an accurate model. The model must not only be trained with the existing images. It must also be tested with images that it has not seen before. In this study, the WikiArt dataset was divided into three main parts. These parts are the training set, the validation set and the test set. Each part has a specific role in the process of training and evaluating the model.

In this paper, a split of 70% for the training set, 15% for the validation set and 15% for the test set was used. First, the dataset was divided into two parts. 70% of the data were used for training the model. The remaining 30% was kept for validation and testing. Then this 30% part was divided into two equal parts. In this way, the validation set with 15% and the test set with 15% were created.

A very important element in this split is stratification. Stratification is used to preserve the distribution of the categories in a similar way across all three sets. This is important because the WikiArt dataset is not fully balanced. Some categories have many images. Some other categories have very few images.

The complete methodological process followed in this study, from the preparation of the dataset to the evaluation of the models, is presented in Figure 1.

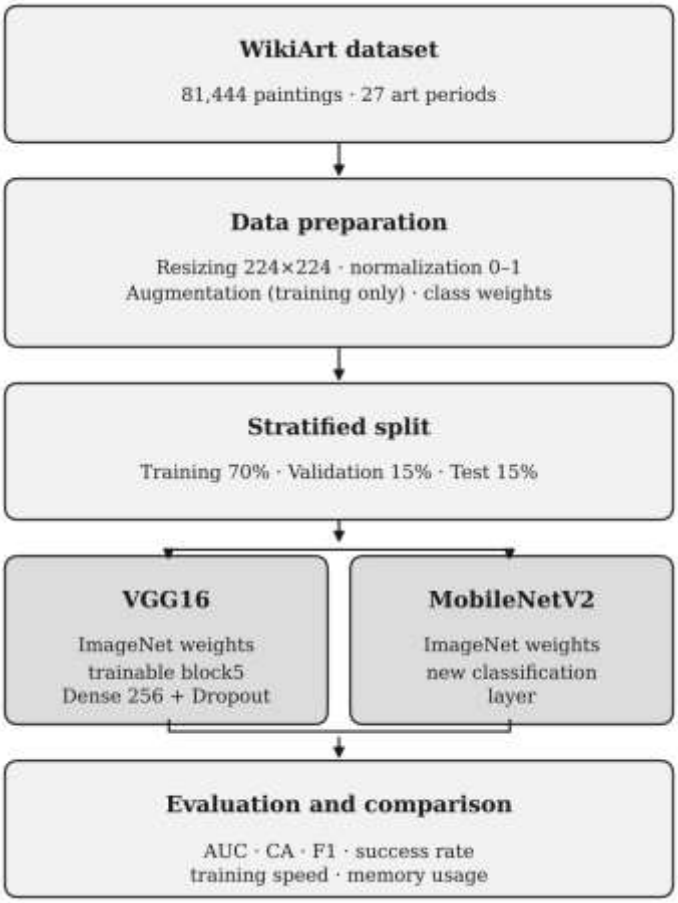


Figure 1. Methodological process of the study.

ADAPTING THE MODELS TO PAINTING CLASSIFICATION

In the original model, VGG16 was built for image classification on the ImageNet dataset, where the model gives predictions for 1000 different classes. The aim of this study is not the classification of general objects, but the classification of paintings according to artistic periods. For this reason, we will not use VGG16 in its original form, but will adapt it to the needs of the study.

The main adaptation concerns the final part of the model. The convolutional layers of VGG16 can be used to extract visual features from images, because they have learned to recognize elements such as shapes, edges, colors, etc. The final layer, which was built for the 1000 ImageNet classes, will be replaced with a new layer that matches the number of classes in the dataset used in this paper.

In this way, we will use VGG16 as a base for transfer learning. The model will not be trained entirely from scratch, but uses weights trained on a large image dataset. This helps us because it reduces the training time of the model (Pan and Yang, 2010; Yosinski, Clune, Bengio and Lipson, 2014; Zhao, Zhou and Qiu, 2021). The first layers of VGG16 will remain as they are, since they learn general features. The final layers will be trained on our dataset, so that the model learns to distinguish between artistic periods.

This step is important, since paintings do not have the same nature as ordinary images. They contain different visual elements such as brushwork, light contrast, colors and composition, which are important points for their classification.

As mentioned earlier, the main aim of this study is the classification of paintings according to artistic periods. For this reason, MobileNetV2 will not simply be used in its original form, but will be adapted to the classes of the dataset used in this paper.

In this case, the first layers of the architecture will be used to extract general visual features, while the final part of the model will be adapted to the specific task of classifying paintings. This means that the model will keep the ability to identify visual elements such as shapes, colors, textures and structures, but the final classification will be directed toward the artistic periods defined in the dataset.

VGG16 IMPLEMENTATION

First, the VGG16 model was used, which was imported from the Keras library. This model was used to carry out the transfer learning process. VGG16 was initialized with weights previously trained on the ImageNet dataset. This enables the model to use the knowledge gained from a large image dataset and to adapt it to our classification problem.

This study does not aim to build a model from scratch. The main aim is to adapt an existing model for the classification of images of paintings according to defined categories. For this reason, at first the base layers of the VGG16 model were set as non-trainable. This is done to preserve the ability of the model to distinguish general visual features such as shapes, contours, colors and textures.

Finally, the layers belonging to block5 were assigned to trainable. This permits the last part of the model to adapt to the characteristics of the dataset used in this study. After the VGG16 base, new layers were added: Flatten layer, Dense layer with 256 neurons and ReLU activation function, Dropout layer for avoiding overfitting, and Softmax output layer for multi-classification.

The model was initialized with the Adam optimizer, with a learning rate of 0.00001. This optimizer was chosen because in the initial tests it gave the best performance compared with the other optimizers tested. To select the most suitable optimizer, an initial test was carried out on a part of the dataset. During this test, the ReLU activation function was kept the same, while several different optimizers were compared. From the results, it was observed that Adam performed better than the other optimizers. Although this test was not carried out on the whole dataset, it can be used as a guiding basis for selecting the most suitable configuration of the model.

Table 3.Results of the optimizer testing.

Optimizer	Neurons	AUC	CA	F1	Success rate (%)
Adam	500	0.905	0.609	0.600	61
SGD	500	0.897	0.600	0.589	59
L	500	0.891	0.583	0.579	58
RMSprop	500	0.895	0.582	0.569	57

MOBILENETV2 IMPLEMENTATION

For building the MobileNetV2 model, steps similar to those used for the VGG16 model were followed. As in the previous case, the transfer learning method was used, where the model is taken ready-made from the Keras library (Keras, n.d.; TensorFlow, n.d.) and the weights previously trained on the

ImageNet dataset are used. This approach helps to take advantage of the knowledge that the model has gained from training on a larger set of images and makes the training process more efficient. First, the model was initialized without the final classification layer. This is done because the original layer of the model was built for the classes of the ImageNet dataset and does not fit directly with the classes of our dataset.

MobileNetV2 is a lighter architecture with fewer parameters compared with VGG16. For this reason, its use is important to observe how a more efficient and lighter model compares with a larger model such as VGG16. This comparison helps to understand whether a simpler architecture can achieve good results in the task of image classification.

RESULTS AND FINDINGS

EVALUATION METRICS

This section presents the results achieved by the two models used in this study, namely VGG16 and MobileNetV2. The main aim of this comparison was to evaluate which model achieves better performance in the classification of images of artworks. For this reason, the comparison was not based only on accuracy, but also on other metrics such as AUC, CA, F1-score and Model Success Rate.

During training, the metrics training accuracy, validation accuracy, training loss and validation loss were analyzed. These metrics help to understand the way in which the model learns during training and whether it manages to generalize to the validation data. In our case, both models showed improvement during the first phases of training, but afterwards signs of overfitting were observed.

RESULTS OF THE VGG16 MODEL

From the results obtained, the VGG16 model achieved the best performance compared with MobileNetV2. This model achieved AUC = 0.966, CA = 0.763, F1 = 0.701 and Model Success Rate = 75%. These results show that VGG16 had a better ability to distinguish the classes and to correctly classify the images of the dataset. Also, the model reached its peak performance at epoch 10, where the validation accuracy was about 75%.

After epoch 10, there was not a big change in the performance on validation anymore and even a slight drop. The training accuracy however, still increased. There is a difference in performance between training and validation, meaning that the model adapted more to the training data. This is a clear indication of overfitting, the model fits well to the training data, but can not improve in the same way on the validation data. The loss graph also indicates this. The training loss kept decreasing as the epochs progressed. The validation loss however had its lowest point and started increasing slowly. This means that the model was still improving on the training data and was not able to generalize the results onto the validation data.

RESULTS OF THE MOBILENETV2 MODEL

The MobileNetV2 model achieved lower results compared with VGG16. This model achieved AUC = 0.946, CA = 0.703, F1 = 0.699 and Model Success Rate = 70%. Although the F1-score value is very close to that of VGG16, the model had a lower CA and success rate. This shows that MobileNetV2 managed to give stable results, but did not exceed the performance of VGG16 in this classification task.

MobileNetV2 reached its peak performance earlier, at epoch 6, with an accuracy of about 69–70%. After this point, the validation accuracy did not improve further and began to show small fluctuations. Overfitting was also observed in this model, since the training accuracy continued to increase while the validation accuracy remained lower and did not follow the same rate of improvement.

PERFORMANCE VERSUS EFFICIENCY

In terms of efficiency, MobileNetV2 was faster during training and used less memory. This is related to its lighter structure, which was built to be more efficient on devices with more limited resources. On the other hand, VGG16 was slower and consumed more memory, but achieved higher results in the main performance metrics.

Table 4. Performance comparison of VGG16 and MobileNetV2.

Metric	VGG16	MobileNetV2
AUC	0.966	0.946
Classification accuracy (CA)	0.763	0.703
F1-score	0.701	0.699
Success rate (%)	75	70
Peak epoch	10	6
Training speed	Slower	Faster
Memory usage	Higher	Lower
Overfitting	Yes	Yes

DISCUSSION

The results show that there is a clear difference between the performance and the efficiency of the two models. VGG16 gave better results in AUC, CA and Model Success Rate, but required more time and more memory. MobileNetV2 was lighter and faster, but achieved lower performance compared with VGG16.

For this reason, in the context of this paper, the VGG16 model was considered more suitable for the continuation of the project. Although it requires more resources during training, it gave better results in the classification of images of artworks. However, the fact that both models showed overfitting indicates that the results can be improved through techniques such as *early stopping*, *better balancing* of the dataset, more controlled data augmentation and tuning of the model parameters.

The results also show that the difference between the two models is not the same across all metrics. While VGG16 was clearly better in AUC, classification accuracy and success rate, the F1-scores of the two models were almost identical (0.701 and 0.699). This suggests that, when precision and recall are considered together, the gap between the two architectures is smaller than the accuracy values alone indicate.

A likely reason for the difficulty of this task is the visual similarity between closely related art periods, such as Cubism, Analytical Cubism and Synthetic Cubism, or the different phases of the Renaissance. These periods share many visual characteristics and can be difficult to separate even for experts. The findings are consistent with earlier studies, which show that fine-tuning networks pretrained on ImageNet is an effective approach for art classification tasks (Cetinić and She, 2022; Zhao et al., 2021).

IMPLICATION TO RESEARCH AND PRACTICE

For research, this study offers a direct comparison between a deep high capacity model and a light-weight model on the same art classification task with the same data preparation and evaluation setup. This study shows that the decision of which model to use cannot be based only on the achieved accuracy, but also on the computational cost of training and using the model. This study additionally confirms that class imbalance and visual similarity between art periods remain major hurdles on automatic art classification. For practice, these results can help galleries, museums, educational platforms etc. to make an informed decision when applying artificial intelligence in their digital archives. VGG16 is the preferred model when high accuracy is the main concern, for instance when digital archiving. MobileNetV2 is the preferred model when speed and memory use are relevant, for instance when applying the technology in a mobile context or as an interactive guide for visitors.

CONCLUSION

A direct comparison shows that VGG16 classified artistic periods more accurately than MobileNetV2. VGG16's validation accuracy kept improving until about epoch 10 and then levelled off at around 75%. MobileNetV2 peaked earlier, near epoch 6, and reached about 70% on the same data split. The AUC scores follow the same pattern: 0.966 for VGG16 and 0.946 for MobileNetV2. Under the conditions we tested, this makes VGG16 the better fit for the task. That advantage has a cost. VGG16 took longer to train and needed more memory. MobileNetV2 was faster and lighter to run, but it was less accurate. Choosing between the two therefore comes down to a trade-off that depends on the system's main constraint: whether the priority is the highest possible accuracy or the lowest computational cost.

FUTURE RESEARCH

Both models showed signs of overfitting. They learned the training data well, but did not manage to generalize to the same extent to the validation data. This problem is related to several characteristics of the dataset: the uneven distribution of images between classes, the visual similarity between closely related artistic styles, and the complexity of the paintings themselves. For this reason, further work could include more controlled data augmentation, regularization techniques, early stopping, better balancing of the dataset and testing of other architectures.

REFERENCES

Cameron, F. and Kenderdine, S. (Eds.) (2007) *Theorizing digital cultural heritage: a critical discourse*, MIT Press, Cambridge, MA.

Cetinić, E. and She, J. (2022) Understanding and creating art with AI: review and outlook, *ACM Transactions on Multimedia Computing, Communications, and Applications*, 18, 1–22. https://doi.org/10.1145/3475799

Goodfellow, I., Bengio, Y. and Courville, A. (2016) *Deep learning*, MIT Press, Cambridge, MA.

Hugging Face (n.d.) *huggan/wikiart* [Dataset]. https://huggingface.co/datasets/huggan/wikiart

Janiesch, C., Zschech, P. and Heinrich, K. (2021) Machine learning and deep learning, *Electronic Markets*, 31, 685–695. https://doi.org/10.1007/s12525-021-00475-2

Keras (n.d.) *Keras Applications*. https://keras.io/api/applications/

LeCun, Y., Bengio, Y. and Hinton, G. (2015) Deep learning, *Nature*, 521 (7553), 436–444. https://doi.org/10.1038/nature14539

Pan, S. J. and Yang, Q. (2010) A survey on transfer learning, *IEEE Transactions on Knowledge and Data Engineering*, 22 (10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

Saleh, B. and Elgammal, A. (2016) Large-scale classification of fine-art paintings: learning the right metric on the right feature, *International Journal for Digital Art History*, 2. https://doi.org/10.11588/dah.2016.2.23376

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L. C. (2018) MobileNetV2: inverted residuals and linear bottlenecks, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, USA, pp. 4510–4520.

Simonyan, K. and Zisserman, A. (2015) Very deep convolutional networks for large-scale image recognition, In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, USA.

Szeliski, R. (2022) *Computer vision: algorithms and applications*, 2nd ed., Springer, Cham.

Tallon, L. and Walker, K. (Eds.) (2008) *Digital technologies and the museum experience: handheld guides and other media*, AltaMira Press, Lanham, MD.

TensorFlow (n.d.) *tf.keras.applications.MobileNetV2*. https://www.tensorflow.org/api_docs/python/tf/keras/applications/MobileNetV2

WikiArt (n.d.) *WikiArt.org: visual art encyclopedia*. https://www.wikiart.org/

Yosinski, J., Clune, J., Bengio, Y. and Lipson, H. (2014) How transferable are features in deep neural networks?, *Advances in Neural Information Processing Systems*, 27.

Zhao, W., Zhou, X. and Qiu, X. (2021) Compare the performance of the models in art classification, *PLOS ONE*, 16 (3), e0248414. https://doi.org/10.1371/journal.pone.0248414