Peer review

1.You do not mention what your classes were. Were the classes the names of the painters/artists? If so, did you have 50 classes because there are 50 unique names in the Kaggle dataset ? Please discuss and present in the manuscript.

2.Please present precision, recall, F1, AUC and Area under the Precision recall curve (PR AUC) for both models used. Please explain and discuss in the manuscript.

3.What features of the paintings/artworks were extracted and given the most importance for the purposes of classification ? Please present a correlogram/heatmap of features that were used for classification and their importance in both the models studied. In this context, you mention that features such as colorjittering, inverting, shearing, gaussian blur, roation, edge detection…… are unsuited for these tasks. In this case, it is important for the reader to know which features were used by your models and their imporatance. Please explain and discuss in the manuscript.

4.There are AI models that actually rely on edge detection to classify art according to stroke feature extraction. See: https://doi.org/10.1515/jisys-2024-0042 , this algorithm achieved an accuracy of > 92%. Please discuss in the manuscript. Similarly, there are other papers that calculate similarity between patches in each painting as a feature, see: https://doi.org/10.1155%2F2022%2F3119604 (retracted). Yet others rely on probability vectors and can achieve a classification accuracy of > 90% see:https://doi.org/10.3390/math11224564 It is important to report exactly how your two methods extracted features (which features) . Please describe and discuss in the manuscript.

5.You mention that you needed to add a class to your models to accommodate another batch normalization layer. If there were already 50 classes for 50 painters/artists in the dataset, what other class did you specifically add to the dataset to enable you to add the extra batch normalization layer? Please explaina and discuss in the manuscript.

6.Was the entire Kaggle dataset split into training/validation/testing (what was the ratio for each of the two models tested) ? Or were images split per painter? In other words, VanGough images split into training/validation/testing etc. for each painter. Please discuss and explain in the manuscript.

7.If paintings that are not in the Kaggle dataset, i.e. by painters not included in the Kaggle dataset are fed to your two algorithms, as what painter do these get classifed as ? For example, if one were to feed The Raft of the Medusa by Gericault, or The Scream by Munch, or The birth of Venus by Botticelli or Goya's Third of May….. these painters would incorrectly be classifed as one of the painters in the Kaggle dataset. Using this information on how the algorithm incorrectly classifies painters, can the algorithm be improved a posteriori ? If so, how? Please discuss and describe in the manuscript. Please feed at least 84 (1%) images by painters not included in the Kaggle dataset and present what painter the algorithm attributes the image to. This can be a table with actual painter and predicted painter columns. This can be presented as an appendix. While I am aware that this is not a primary objective of your study, I have not seen this type of analysis in the literature. It will add to your manuscript's readership.

8.In making all the images the same size, did you change the length to width ratio of any of the paintings? Describe and discuss to what extent this manipulation may have affected accuracy by influencing features and feature importance (see point 3).

9.Present the computer execution speed of Resnet 152 verus Resnet 50 in your manuscript. What implications, if any, does this have in practical implemenation?

Question 1 about the dataset classes:

Original:

> *Dataset*
> The dataset utilized in this research is the "Best Artworks of All Time" dataset on Kaggle, which consists of 8,446 artworks by 50 famous artists from Europe and Northern America with each class containing the artworks by one author [5].

10.

11.Revised:

> *Dataset*
> The dataset utilized in this research is the whole "Best Artworks of All Time" dataset on Kaggle, which consists of 8,446 artworks by 50 famous artists from Europe and Northern America. Each class is named after a unique artist and contains all artworks by that artist [5].

12.

13.Added the fact that all classes are named after an unique artist and include all artworks of that respective artist. I also used all classes in the dataset as the "whole" in the first sentence refers to the dataset only having 50 classes.

14.

15.

16.Question 2 about performance metrics:

17.

Final Results

The final results of the two models trained in this research are shown in terms of validation and test set accuracy. The combination of these two metrics is an excellent way to evaluate a model's generalization ability on unseen data.

The algorithms and training environments used to train the two ResNet models in this research are identical. The performance of both models is shown in Table 1. Both models reached a validation and test accuracy of above 80%.

Table 1: Final results of the models

	Valid. Acc.	Test Acc.
ResNet50	84%	83%
ResNet152	88%	86%

18.Original:Revised:

Final Results of Trained Model

The final results of the two models trained in this research through various metrics: validation and test set accuracy, precision, recall, F1, AUC, and PR AUC.

Precision measures the percentage of positive predictions that are positive. The recall rate indicates the percentage of positive predictions out of all positive cases. The F1 score is the harmonic mean of precision and recall rate.

AUC, the Area Under the Receiver Operating Characteristic Curve, assesses how well the model can distinguish between positive and negative classes. PR AUC, Area Under the Precision-Recall Curve, evaluates the model's performance on positive predictions in imbalanced datasets.

Combining these metrics is an excellent way to evaluate a model's generalization ability on unseen data.

Table 1: Final results of the models

	ResNet50	ResNet152
Valid. Acc.	85%	88%
Test Acc.	83%	86%
Precision	0.85	0.86
Recall	0.87	0.87
F1	0.85	0.86
AUC	0.99	0.99
PR AUC	0.92	0.93

The algorithms and training environments used to train the two ResNet models in this research are identical. The performance of both models is shown in Table 1. Both models reached a validation and test accuracy of above 80% along with a precision, recall rate, and F1 score of above 0.85.

19.

20.Added the five performance metrics of both models and a description of the metrics. I also switched the orders or the paragraphs to first show the table then the analysis.

21.Question 3 about important features:

22.

Revised:

Analysis of Features Importance Heat Map

Figures 3 and 4 shown above are an image of Roberto Montenegro's Cabeza Azteca and its features importance heat map. The areas of highest significance are centered around the intricate patterns and contours of the figure's head and facial features. These regions are emphasized due to their unique structural elements and contrasting colors, which are critical for distinguishing this artwork from others.

To enhance the model's performance, data transformation methods such as horizontal flipping are used, allowing the model to recognize the artwork's features regardless of orientation. Similarly, slight color jittering helps the model adapt to minor variations in color intensity and lighting. Random affine introduces subtle changes in perspective, helping the model generalize better across different viewing angles.

These methods not only assist the model in identifying the most important features of the artwork but also prevent the loss of critical information—such as the figure's head in Cabeza Azteca — and avoid distortion of spatial structures, issues that can arise from cropping, rotation, and blurring.

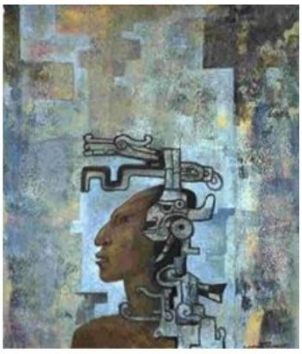

Figure 3: Cabeza Azteca by Roberto Montenegro

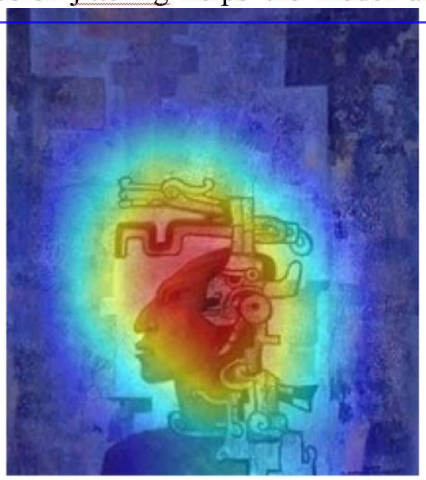

Figure 4: The features importance heat-map of Cabeza Azteca by Roberto Montenegro

44.Added a new section analyzing a features importance heat-map and why I used the three data transformation methods chosen.

45.Question 4 about feature extraction details:

46.

47.Original:

When the image enters the ResNet model, it first passes through an initial convolutional layer and a max-pooling layer, which reduces the dimensions of the image and extracts basic features. The image then passes through a series of residual blocks, where convolution operations and the use of shortcuts occur. The model continuously abstracts features through out these layers, which makes ResNet models highly suitable for image recognition tasks.

48.

49.Deleted the original paragraph and wrote a new paragraph that explains how ResNet models extract features in detail.

50.

51.

52.

53.

54.

55.

56.

57.

58.

59.Question 5 about batch normalization:

60.

61.Original:

ResNet models already have batch normalization layers. To add another batch normalization layer, a class needs to be defined to act as the modified model and inherits a pretrained ResNet50 or ResNet152. A batch normalization layer in then added to the forwards function of the class.

Final Results

62.

Revised:

ResNet models extract features from images through a series of stages. First, the input image is resized to a fixed size of 224x224 pixels. It then passes through an initial convolutional layer which filters low-level features like edges and textures. The core of ResNet consists of multiple residual blocks, each containing convolutional layers that learn feature representations, batch normalization layers that stabilize training, ReLU activation functions for non-linearity, and skip connections that add the input of each block to its output. As images progress through these blocks, ResNet captures increasingly complex and abstract features, moving from simple shapes to high-level abstract patterns through a process called feature hierarchy. Pooling layers are applied after the residual blocks to downsample the feature maps, reducing dimensionality while preserving essential information. Finally, the features are flattened and passed through a fully connected layer to become a feature vector, which outputs the final prediction.

63.Revised:

ResNet models contain batch normalization layers. To add another batch normalization layer, a class needs to be defined to customize a pre-trained ResNet50 or ResNet152 model, which is shown in Figure 6 above. Note that the class here refers to the Python class, not to be confused with the classes in the dataset, which represent the categories fed into the model.

```
5 class ResNet50WithExtraBN(nn.Module):
6     def __init__(self, num_classes=len(dataset.classes)):
7         super(ResNet50WithExtraBN, self).__init__()
8         self.resnet = models.resnet50(pretrained=True)
9         self.resnet.fc = nn.Linear(self.resnet.fc.in_features, num_classes)
10
11        self.extra_bn = nn.BatchNorm1d(num_classes)
12
13    def forward(self, x):
14        x = self.resnet(x)
15        x = self.extra_bn(x)
16        return x
```

Figure 4: Python ResNet50WithExtraBN class used to add a batch normalization layer to a Resnet50 or Resnet152 model

64.Added a figure of the used python class and clarifications of what “class” meant in this batch normalization paragraph.

65.

66.

67.Question 6 about splitting the dataset:

68.

69.Revised:

Each of the 50 classes in the dataset is split into training, validation, and testing sets in a 70/10/20 ratio, minimizing the bias between classes compared to splitting the dataset as a whole.

70.

71.Added a paragraph to explain how I split the dataset.

72.Question 7 about classifying artworks from unfamiliar artists:

73.

74.Revised:

Results of Classifying Pieces From Unfamiliar Artists

After feeding 84 artworks from 24 artists that were not included in the dataset into the model, some intriguing patterns were found (see Appendix I for more details). For example, all three pieces by John Leech were classified as works of Vincent van Gogh. This likely occurred due to stylistic and structural similarities between the two, as both artists were prominent figures in the realism art movement during the 19th century. Similarly, the three artworks by John McLaughlin were all classified as works of Kazimir Malevich, reflecting their shared late 19th-century origins and contributions to abstract art. These observations suggest that the algorithm emphasizes stylistic attributes over individual artist characteristics. Consequently, the pieces of some artists were classified as works of separate artists from the same era who explore similar genres.

To improve classification accuracy, the model should prioritize style and genre over specific characteristics of individual artists or be trained on a dataset where classes only contain one genre of the artist's pieces. Emphasizing individual characteristics could hinder the model's generalization abilities since many artists create pieces across different art movements and genres. By transitioning to a style recognition model, where classes represent various genres and movements, the model could achieve more precise classifications, as the classes would share greater stylistic similarities.

7 Appendix 1

Artwork Name	Actual Artist	Predicted Artist
The Sitting Room	Frances Hodgkins	Vincent van Gogh
A Barn in Provence	Frances Hodgkins	Vincent van Gogh
Der Totentanz Von Anno Neun	Albin Egger-Lienz	Pablo Picasso
Der Bauer	Albin Egger-Lienz	Henri Matisse
All Lanes of Lilac Evening	Max Ernst	Joan Miro
Appius Claudius Punished By The People	John Leech	Vincent van Gogh
Ariadne	George Frederick Watts	Titian
Austrian Family, The	Josef Kriehuber	Gustav Klimt

75.

76.

77.Added a whole new appendix of all classified artworks from artists outside of the dataset and a paragraph to explain the results and possible improvements.

78.Question 8 about resizing images:

79.

80.Original:

Another important data augmentation method used to increase model performance is resizing all images to a single size. The model shown in Figure 1 is trained using transforms.Resize(224), which resizes the shorter side of the image to 224 pixels while keeping the original aspect ratio of the image. However, the model in Figure 2 is trained using transforms.Resize(224, 224), which resizes all images to 224*224 pixels. Keeping all images the same size during training is crucial as it increases the efficiency and stability of the model.

Revised:

One important data augmentation method used to increase model performance is resizing all images to a uniform size. The model shown in Figure 1 is trained using 'transforms.Resize(224)', which resizes the shorter side of the image to 224 pixels while keeping the original aspect ratio of the image. However, the model in Figure 2 is trained using 'transforms.Resize(224, 224)', which resizes all images to 224*224 pixels, changing the original length-to-width ratio of the image and contributing to the large increase in validation accuracy mentioned in the last paragraph. Even though resizing images can affect the structure of some artworks (for example the proportions of the figure's head in Cabeza Azteca shown in Figure 3 below), keeping all images the same size during training is crucial since it increases the efficiency and stability of the model.

81.Clarified what transforms.Resize(224, 224) does, how it affects the model's performance, and how it affects features of images (which is also explained in the Areas of Improvements section) to the original paragraph about resizing images.

82.

83.

84.Question 9 about execution time:

85.

86.Revised:

87.

Execution Speed

When classifying a single artwork, both the ResNet50 and ResNet152 model have an execution time under 0.03 seconds. These models are very efficient and suitable for real-time applications, such as an art recognition app. This speed enables the model to provide instant results, enhancing user experience, especially when used on mobile devices or in situations where quick response times are crucial.

speed of the two models when classifying ions.

Execution Speed

When classifying a single artwork, both the ResNet50 and ResNet152 model have an execution time under 0.03 seconds. These models are very efficient and suitable for real-time applications, such as an art recognition app. This speed enables the model to provide instant results, enhancing user experience, especially when used on mobile devices or in situations where quick response times are crucial.

Each of the 50 classes in the dataset is split into training, validation, and testing sets in a 70/10/20 ratio, minimizing the bias between classes compared to splitting the dataset as a whole.

Final Results

The final results of the two models trained in this research are shown in terms of validation and test set accuracy. The combination of these two metrics is an excellent way to evaluate a model's generalization ability on unseen data.

The algorithms and training environments used to train the two ResNet models in this research are identical. The performance of both models is shown in Table 1. Both models reached a validation and test accuracy of above 80%.

Table 1: Final results of the models

	Valid. Acc.	Test Acc.
ResNet50	84%	83%
ResNet152	88%	86%

Thank you for addressing my comments. Accept.