Art Style Classification: Enhancing Accuracy with Shallow Neural Network Adapter

Lu Yingxi*
2023011435
Class 31
lu-yx23@mails.tsinghua.edu.cn

Zhu Fangyu†
2023011410
Class 32
zhufy23@mails.tsinghua.edu.cn

Abstract

We tackle the art style classification problem by combining deep neural networks (DNNs) with a shallow neural network (SNN) adapter. Our approach involves a two-stage architecture where a DNN extracts features from image patches and an SNN adapter makes the final classification. We propose a hierarchical architecture that integrates the strengths of both the DNN and SNN adapter, and explore different patch selection strategies to optimize model performance. Experiments on three pretrained DNN models (DenseNet-121, VGG-19, and ResNet-50) using the Painter by Numbers dataset show that our method improves classification accuracy and stability. Notably, using object-detecting patch selection enhances performance while reducing computational cost. Our work advances art style classification by effectively combining DNNs with SNN adapters and provides valuable insights for future research.

1. Introduction

In this project, we aim to address the challenge of classifying paintings according to their artistic styles.

This task lies at the intersection of computer vision, machine learning, and art history, making it a multifaceted problem with significant implications for both technological and cultural domains[3]. In recent years, the automated classification of art painting styles using deep convolutional neural networks (CNNs) has become essential for analyzing and categorizing vast digitized art collections. These models can learn hierarchical visual features directly from raw image data, providing a powerful tool for art analysis[13][9]. However, accurately recognizing and distinguishing stylistic characteristics across diverse art movements remains a challenge due to high intra-class variability and inter-class similarity[1]. Developing robust classification methods is critical for scalable digital archiving, enhancing curatorial workflows, and enabling large-scale quantitative analysis of artistic trends to support art historical research.

Our project builds upon the approach proposed in[9], which involves a two-stage architecture combining a deep neural network (DNN) and a shallow neural network (SNN) adapter. The DNN acts as a feature extractor, while the SNN adapter is responsible for decision-making. Each input image is divided into five distinct patches: top-left, top-right, bottom-left, bottom-right, and center. The DNN independently classifies each of these five patches, and the SNN adapter aggregates the results to produce a final classification. This architecture allows for a more detailed examination of different regions within an artwork, capturing fine-grained information and preserving important artistic details[9]. Importantly, the SNN operates independently of the DNN, meaning that the introduction of the SNN does not alter the architecture or weights of the DNN. This independence allows the SNN to function as a flexible adapter, enhancing the classification process without imposing any architectural constraints on the DNN. Thus, the integrity and performance of the original DNN are maintained while adding an additional layer of refinement to the classification results.

By comparing the prediction results of the DNN direct output and the SNN adapter output, we observed that while the SNN achieves higher overall accuracy, the DNN direct prediction exhibits superior accuracy in certain specific classes. To integrate the strengths of both networks, we proposed a hierarchical architecture. We trained an SNN adapter using four image patches: left-top, left-bottom, right-top, and right-bottom. The final prediction is then calculated as a weighted sum of the DNN's direct prediction and the SNN adapter's output. This combined approach not only yields higher

total accuracy than the original SNN adapter architecture but also results in more stable accuracy across different classes.

Furthermore, we also explored how the selection of patches affects the performance of the model. To improve the model's performance, we tried multiple ways to process the images and extract patches. A straightforward idea is to add the original image as an additional patch–actually it is surprising that the the original work [4] did not mention this. Therefore we have 6 patches helping the model to make a decision. In addition, to make the patch selection more syntactically meaningful, we tried to add a patch focusing on the main object in the image. However, sometimes it is hard to say whether there exists a main object in an artwork, and the main object appears near the center of the image, so we merged the center patch and the main object patch into one. As the four corner patches usually contains less information than the original and center patches, we tried to do some experiments with only two patches to reduce the computational cost, and found that it still works well, sometime even better since some inaccurate predictions are removed.

- We implement DNN + SNN sdapter architecture proposed in [9], and proof its effectiveness by ablation study.
- We propose a hierarchical architecture that combines the strengths of both the DNN and SNN adapter, resulting in improved accuracy and stability across different classes.
- We conduct an extensive analysis of the impact of different patch selections on the model's performance, providing insights into the optimal configuration for art style classification.
- We provide a conprehensive evaluation of our proposed method on 3 various already-trained DNN models, namely DenseNet-121, VGG-19, and ResNet-50, demonstrating its effectiveness and generalizability across various models.

2. Related Work

2.1. Traditional and CNN-based Approaches

Early art classification systems relied on hand-crafted descriptors. Researchers extracted low-level features (e.g., color histograms, Gabor filters, LBP) and trained classical classifiers: Paul and Malathy [15] evaluated dense SIFT, HOG, LBP, GLAC, color naming, and GIST, concluding LBP performed best; Nunez-Garcia et al. [14] emphasized combining color and texture for brushstroke and composition modeling.

With deep learning, convolutional neural networks (CNNs) became dominant: Saleh and Elgammal [17] used pretrained CNNs with metric learning for style recognition; Hentschel et al. [6] and Cetinic et al. [3] fine-tuned ImageNet models on WikiArt, demonstrating substantial gains. While modern CNNs (ResNet, EfficientNet) typically end in a softmax layer, our method combines CNN feature extraction with a Spiking Neural Network (SNN) adapter for efficient classification on neuromorphic hardware.

2.2. Patch-Based and Multi-Stage Models

To capture local style cues, patch-wise and multi-stage models split images into subregions. Hua et al. [7] proposed a CNN-MRF pipeline that smooths per-patch predictions via a Markov Random Field; Jangtjik et al. [10] developed a CNN-LSTM that aggregates multi-scale patch outputs. [9] introduced the DNN+SNN adapter, dividing each image into five patches, classifying them by a deep network, and fusing probability vectors through a shallow adapter. Although effective, these models incur extra complexity in inference or training stages. Our baseline reproduction of DNN+SNN maintains simplicity and demonstrates strong accuracy before further extensions.

2.3. Proposed Hierarchical Fusion

Building on DNN+SNN, we introduce a hierarchical fusion scheme: we compute a global prediction from the CNN and a local prediction from the SNN on image patches, then combine them with entropy-weighted coefficients. This two-layer adapter enhances robustness on classes where either global or local view alone may be insufficient, while preserving the single-stage inference flow and minimal parameter overhead of the original model.

2.4. Semantic and Attention-Based Approaches

Some works leverage semantic object detection or attention mechanisms. Transformer-based models (ViT, MLP-Mixer) have been applied to style classification by Iliadis et al. [8], and Swin Transformers excel in art authentication tasks by Schaerf et al. [18]. However, these approaches often require substantial computational resources. Our method, by contrast, combines CNN feature extraction with a lightweight SNN adapter, boosting accuracy of existing prediction models without adding significant complexity. Other methods use detectors: Aslan and Steels [2] highlighted domain-shift issues for generic detectors like YOLOv3 and Mask R-CNN on paintings. After reproducing the DNN+SNN baseline [9], we fine-tuned YOLOv5 [11] to crop the main subject's bounding box, focusing on semantically rich regions and further improving accuracy while reducing computational cost.

3. Method

Our method consist of three main parts: basic SNN adapter proposed in [9], hierarchical SNN architecture, and patch selection for SNN.

3.1. Basic SNN Adapter

The basic classification pipeline combining a base DNN with an SNN adapter is illustrated in Figure 1. The process consists of two main stages.

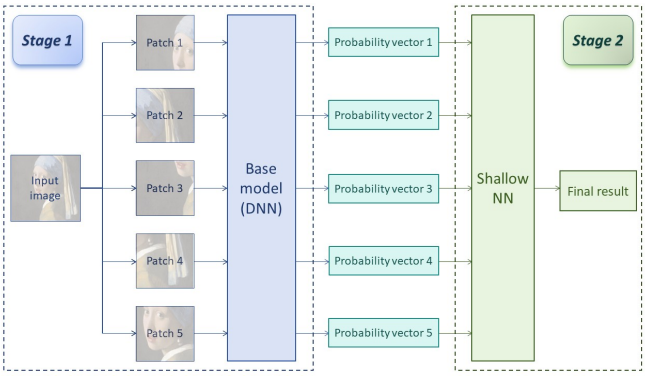

Figure 1. The architecture of the basic SNN adapter

In the first stage, the input image is divided into five distinct patches (Figure 2). The DNN processes each patch independently, generating a preliminary classification for each one. This allows the model to focus on local features within different regions of the image.

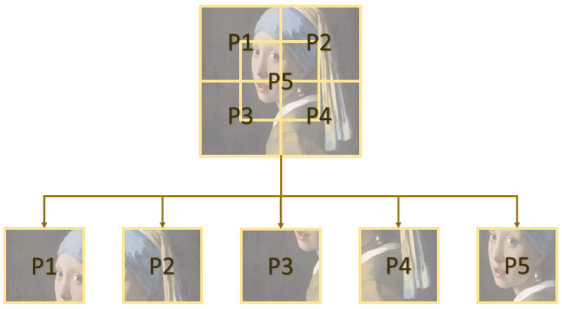

Figure 2. Patch dividing of each image

In the second stage, the SNN adapter acts as a decision-maker. It takes the probability distributions from the DNN's patch classifications as input and produces the final classification result. The SNN architecture, shown in Figure 3, includes a single hidden layer. The SNN is trained using data comprising five probability vectors (from the DNN's patch classifications) as inputs and ground truth labels as outputs. Once trained, it generates the final style prediction during inference.

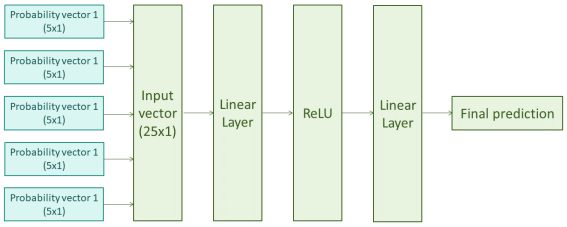

Figure 3. The architecture of the SNN adapter

The proposed two-stage architecture combining a DNN and an SNN offers several key advantages:

- It enables a more detailed examination of different regions within an artwork. By capturing fine-grained information and preserving important artistic details, this architecture enhances classification accuracy[9].
- Using probability vectors instead of images as inputs to the SNN reduces computational costs and avoids potential errors during image processing.
- The SNN functions as an independent decision-making adapter, making the model more flexible and generalizable. Since the DNN and SNN are trained independently, we can adjust their weights and architectures according to our needs. Each SNN adapter can fit different kinds of DNN models.

3.2. Hierarchy SNN Architecture

To fully leverage the capabilities of both the DNN and the SNN adapter, we designed a two-layer hierarchical classification architecture. This approach is motivated by the complementary strengths of global and local feature extraction in the context of art style classification.

3.2.1. Architecture Design

As illustrated in Figure 4, the hierarchical architecture comprises two distinct layers.

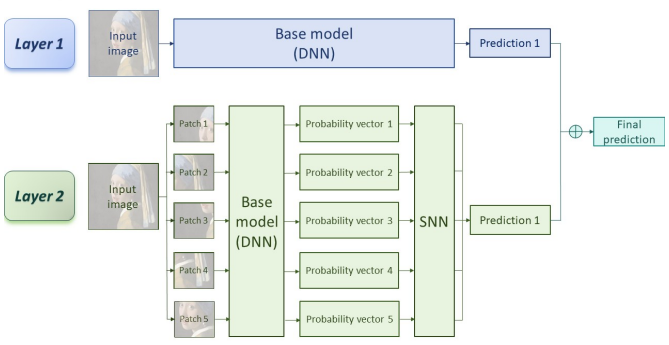

Figure 4. The architecture of the hierarchical SNN adapter

In the first layer, the input image is directly fed into the base model, which is a pretrained DNN. This layer serves as a global feature extractor, providing a comprehensive overview of the image's content and style. The output of this layer is a probability vector, denoted as p_{layer1} , which represents the initial prediction based on the entire image.

The second layer builds upon our previous work on the SNN adapter. Here, the input image is divided into four patches: left-top, left-bottom, right-top, and right-bottom. The DNN processes each patch individually, generating probability vectors p_1, p_2, p_3, p_4, p_5. These vectors are then fed into the SNN adapter, which produces a second probability vector, p_{layer2}. This layer focuses on local features, capturing fine-grained details that might be overlooked in the global analysis.

3.2.2. Weighted Sum for Final Prediction

The final prediction is calculated as a weighted sum of the predictions from the two layers:

$$p_{\text{final}} = w_1 \cdot p_{\text{layer1}} + w_2 \cdot p_{\text{layer2}}, \tag{1}$$

where the weights w_1 and w_2 are proportional to the "reliability" of the respective layers. Inspired by [7], the "reliability" can be evaluated based on the entropy of the probability vectors. Specificly, the entropy of the layer 1 prediction is calculated as:

$$H^1 = -\sum_{i=1}^{C} p_{\text{layer1}}(i) \cdot \log(p_{\text{layer1}}(i)), \tag{2}$$

and the corresponding weight w_1 is given by:

$$w_1 \propto 1 + \lambda / \exp(H^1). \tag{3}$$

Similarly, for the second layer, the entropy:

$$H^2 = -\sum_{i=1}^{C} p_{\text{layer2}}(i) \cdot \log(p_{\text{layer2}}(i)), \tag{4}$$

and the corresponding weight w_2 is given by:

$$w_2 \propto 1 + \lambda / \exp(H^2), \tag{5}$$

where λ is a hyperparameter.

This entropy-based weighting mechanism allows the model to dynamically adjust the contribution of each layer based on the confidence of their predictions. When a layer's prediction has low entropy (indicating high confidence), it is assigned a higher weight, thereby emphasizing more reliable predictions in the final output.

By combining global and local feature extraction in this hierarchical manner, our architecture not only improves the overall accuracy of art style classification but also enhances the model's robustness and generalizability.

3.3. Patch Selection for SNN

Although dividing an image into five patches has been shown to be effective, we still wish to explore better patch-selection methods. During our tests, we observed that the baseline model's classification scores for the patches in the four corners are quite inaccurate. This is because the original patch-selection method only considers spatial information and ignores semantic content. The main subject of an image (for example, a person) usually appears near the center, whereas corner patches often contain only partial views of the subject or merely background. We suspect that this unbalanced composition also affects the baseline model's style judgment. To address this issue, we propose a new patch-selection method that detects the image's main subject and uses its bounding box as the patch for classification. Specifically, we employ a pretrained YOLOv5 model [11] to detect the main subject and use the resulting bounding box as the patch for classification.

Moreover, we surprisingly found that the original patch-selection method by [9] does not mention using the original image as a patch for classification. So in our own experiments, we tried to use only 2 patches that contain most of the image information: the original image and object bounding box. We found that this method can achieve similar or even better performance to the original 5-patch method, while significantly reducing the computational cost, since the SNN only needs to process 2 inputs.

4. Experiment

In this section, we present an comprehensive evaluation of our proposed method.

4.1. Experiment Setup

4.1.1. Dataset

We utilize the Painter by Numbers dataset [12] as the primary source of paintings for our study. This dataset is a vast collection of approximately 103,250 unique paintings, each labeled with the painter's name, style (or movement), genre, and additional information[4]. For each experiment, we select a subset of paintings from the dataset to generate the training data. The scale of of the training dataset is set to be about 5000 at the beginning of our experiment, but when we use the 2-patch method, we found that we are able to handle more data as the computing cost is reduced, so the training dataset is expanded to about 20000 images. The training set samples are selected uniformly according to the baseline model's classifications. Additionally, we use 500 paintings from the test set for

evaluation purposes.

4.1.2. Implementation Details

The experiment pipeline is almost invariant across different baseline DNN models.

- Data Processing: We scan every row of the .CSV file provided the Painter by Numbers dataset to extract the paintings' file names, their true style labels, and whether they belong to the training or test set. If the file path, the label and the set type are all valid, we start to process it as a data sample. For each painting, we divide it into five patches (or fetch the main object patch pre-detected by YOLOv5) and resize each patch to the correct input size for the DNN model. The model returns a probability vector for each patch, which is then stored in a list. We wrap the file name, the true label, and the list of probability vectors into a JSON file named after the painting. The JSON file is then saved in the training or test set folder according to the set type. We generate data samples uniformly according to the baseline model's classifications by set a threshold for the number of samples for each class.
- SNN Training: We load all JSON files from the dataset folder, convert the inputs and labels(one-hot) into PyTorch tensors, and feed them via a DataLoader. The shallow network flattens each patch set, applies several hidden layers with dropout, and outputs a style score vector. We train the model end-to-end using a regression loss and an adaptive optimizer with learning-rate scheduling, periodically reporting training loss, and finally save the learned weights for downstream evaluation.The optimal hyperparameters for each experiment can be silightly different, but 2 hidden layers with dropout technique applied usually works well.
- Evaluation: We feed the 500 images we select as test set to the baseline model or the DNN+SNN architecture, and accumulate per-sample and per-style counts to compute overall and per-class accuracy, and finally output the precision, recall, and F1-score breakdowns for each style. The specific evaluation metrics are described in detail in the next section.

4.1.3. Evaluation

In order to get a comprehensive understanding of the model's performance, we evaluate the model using four metrics: accuracy, precision, recall and F1 score. [1] We denote:

TP Number of correctly identified positive samples.

TN Number of correctly identified negative samples.

FP Number of negative samples incorrectly labeled as positive (Type I error).

FN Number of positive samples incorrectly labeled as negative (Type II error).

Then we can calculate the four metrics as follows:

- Accuracy: The ratio of correctly predicted instances to the total instances in the dataset.

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

- Precision: The ratio of true positive predictions to the total predicted positives. It indicates the quality of the positive predictions.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- Recall: The ratio of true positive predictions to the total actual positives. It measures the model's ability to identify all relevant instances.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- F1 Score: The harmonic mean of precision and recall. It provides a balance between precision and recall, especially useful when dealing with imbalanced datasets.

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Moreover, we not only evaluate the overall model performance but also analyze each class's performance. This provides deeper insights into the model's strengths and weaknesses across categories, helping identify specific areas needing improvement and ensuring consistent performance across all classes.

4.1.4. Baseline DNN Architecture

We evaluate out SNN adapter on 3 various already-trained DNN models, DenseNet-121, VGG-19, and ResNet-50. For detailed information about the classification of each model, please refer to Appendix 5.

- DenseNet-121: We adopt ArtNet model[4] trained with DenseNet-121 architecture as our first baseline A key feature of ArtNet is its preprocessing strategy, where each input image is first padded to a uniform size and then divided into five patches. Both the full

[1] The precision, recall and F1 score mentioned here are all weighted-averaged, meaning they are calculated for each class and then averaged across all classes. The detailed values for each class can be found in the Appendix.

image and its patches are included in the training set, enabling the model to learn from global composition as well as local texture details, making it a perfect choice for our baseline model.

- VGG-19: We utilize the model trained by [16] as our baseline for VGG-19. This model is trained on the same dataset as our method, with a test accuracy lower than ArtNet.
- ResNet-50: We also employ the pretrained ResNet-50 model from [5]. Our results show that even if the base model's performance is subpar, the SNN adapter can still significantly enhance the overall performance.

The following sections present the detailed experimental results and analysis.

4.2. Basic SNN Architecture

In our ablation study, we selected two baseline models for comparison with our DNN + patch-based SNN adapter architecture. First, we chose a baseline DNN model without any patch processing or adaptive integration, which represents a standard deep learning approach to art style classification. This allows us to directly observe the performance improvement gained from our patch-based methodology.

Second, we introduced a patch-based DNN model that generates predictions by averaging the five patch predictions. This model serves as an intermediate step to evaluate the inherent value of the patch-based approach itself. By comparing the performance of the baseline DNN model, the patch-based DNN model, and our DNN + patch-based SNN adapter architecture, we aim to comprehensively assess the incremental benefits of each component in our proposed framework.

The results of our ablation study are presented in Table 6. As expected, the patch-based DNN model demonstrates improved performance over the baseline DNN model. This highlights the importance of considering local features within different image patches for art style classification. The patch-based approach allows the model to capture fine-grained details that might be overlooked when processing the entire image at once. Furthermore, our DNN + patch-based SNN adapter architecture outperforms both the baseline DNN model and the patch-based DNN model. This comparison between the patch-based DNN model and our SNN-integrated architecture underscores the power of the SNN in effectively aggregating and interpreting the patch-level information. The SNN adapter acts as a sophisticated decision-maker that can weigh and combine the predictions from different patches more intelligently than simple averaging.

Notably, these improvements were realized using a training dataset of fewer than 6,000 paintings, with the SNN training process taking less than 5 minutes. This highlights the efficiency of the SNN adapter in significantly enhancing the classifier's performance with limited training data and time, thereby underscoring its effectiveness in improving DNN models for art style classification.

	Accuracy	Precision	Recall	F1 Score
Baseline	51.8%	0.5571	0.5180	0.4876
Averaged	61.0%	0.6332	0.6100	0.5885
SNN	62.2%	0.6600	0.6220	0.6276

(a) DenseNet-121

	Accuracy	Precision	Recall	F1 Score
Baseline	25.8%	0.3276	0.2580	0.2410
Averaged	27.4%	0.3585	0.2740	0.2477
SNN	29.0%	0.3513	0.2900	0.2690

(b) VGG-19

	Accuracy	Precision	Recall	F1 Score
Baseline	20.0%	0.1369	0.2000	0.0959
Averaged	23.2%	0.2393	0.2320	0.2148
SNN	29.0%	0.2675	0.2900	0.2436

(c) ResNet-50

Table 1. Evaluation results of the basic SNN adapter

4.3. Hierarchy SNN Architecture

In our experimental analysis, we compare our proposed hierarchy SNN architecture against two benchmarks: the baseline DNN model and the basic SNN adapter. This comparison is designed to evaluate the added value of our hierarchical approach over both a standard DNN and a previous SNN implementation.

The results, as shown in Table 2, reveal a clear pattern of performance enhancement. For both DenseNet-121 and VGG-19, the hierarchy SNN architecture surpasses the basic SNN adapter in both accuracy and F1 score. This suggests that the hierarchical integration of global and local features is more effective in these contexts.

In the case of ResNet-50, although the hierarchical SNN architecture shows a slight dip in prediction accuracy compared to the basic SNN adapter, it achieves a higher F1 score. This indicates that while the hierarchical approach may not always lead to outright improvements in raw accuracy, it provides a more balanced performance, particularly in terms of precision and recall. The F1 score's improvement implies that the hierarchical SNN architecture is better at maintaining a balance between these two metrics, which is

	Accuracy	Precision	Recall	F1 Score
Baseline	51.8%	0.5571	0.5180	0.4876
Basic	61.0%	0.6332	0.6100	0.5885
Hierarchy	62.2%	0.6600	0.6220	0.6276

(a) DenseNet-121

	Accuracy	Precision	Recall	F1 Score
Baseline	25.8%	0.3276	0.2580	0.2410
Basic	29.0%	0.3513	0.2900	0.2690
Hierarchy	29.6%	0.3751	0.2960	0.2762

(b) VGG-19

	Accuracy	Precision	Recall	F1 Score
Baseline	20.0%	0.1369	0.2000	0.0959
Basic	29.0%	0.2675	0.2900	0.2436
Hierarchy	25.8%	0.2586	0.2580	0.2520

(c) ResNet-50

Table 2. Evaluation results of the hierarchy SNN architecture

crucial for a comprehensive evaluation of model performance.

We delved deeper into the performance of our proposed hierarchy SNN architecture by conducting a per-class analysis. Table 3 presents the detailed results for the DenseNet-121 model as a representative example.

The analysis reveals a stark contrast in performance across different art styles for the baseline DenseNet-121 model. It achieves high accuracy of 93

In comparison, the basic SNN adapter shows significant improvements in Expressionism and Impressionism, increasing accuracy to 49

The hierarchy SNN architecture, as expected, demonstrates a balanced and enhanced performance. It effectively combines the strengths of both the baseline DenseNet-121 model and the basic SNN adapter. By integrating global and local feature information, it not only maintains the high accuracy of Realism and Abstract styles but also significantly improves the classification of Expressionism and Impressionism. This comprehensive performance enhancement across different art styles underscores the effectiveness of the hierarchy SNN architecture in capturing the hierarchical structure of paintings and provides a more robust solution for art style classification tasks.

4.4. Object-Detecting Patch Selection

To further enhance the model's performance, we explore the impact of patch selection on the classification results. Instead of the basic 5-patches data processing, we experiment with a 2-patches method, where we only

Class	Accuracy	Precision	Recall	F1 Score
Cubism	49%	0.8033	0.4900	0.6087
Expressionism	11%	0.2500	0.1100	0.1528
Impressionism	34%	0.7556	0.3400	0.4690
Realism	93%	0.4227	0.9300	0.5813
Abstract	72%	0.5538	0.7200	0.6261

(a) Baseline DenseNet-121 model

Class	Accuracy	Precision	Recall	F1 Score
Cubism	54%	0.8852	0.5400	0.6708
Expressionism	49%	0.3740	0.4900	0.4242
Impressionism	59%	0.5728	0.5900	0.5813
Realism	84%	0.6942	0.8400	0.7602
Abstract	65%	0.7738	0.6500	0.7065

(b) Basic SNN adapter

Class	Accuracy	Precision	Recall	F1 Score
Cubism	53%	0.8833	0.5300	0.6625
Expressionism	45%	0.3814	0.4500	0.4128
Impressionism	55%	0.5978	0.5500	0.5729
Realism	90%	0.6818	0.9000	0.7759
Abstract	71%	0.7245	0.7100	0.7172

(c) Hierarchy SNN architecture ($\lambda = 10$)

Table 3. Results of each class for DenseNet-121

use the original image and the main-object patch. The main-object patch is obtained by detecting the main subject in the image using a pretrained YOLOv5 model [11] and resize its bounding box.

We compare the performance of the 2-patches method with the basic 5-patches method and baseline models. The results are shown in Table 4.

We can see that object-patch method achieves better performance than the baseline model and even the basic 5-patches SNN adapter with only 2 patches used. This phenomenon may suggest that the semantic information of an image plays a more important role in style classification tasks.

In addition, we also tried to combine the object-patch selection with original 5-patches strategy by repalacing the center patch with the object patch. However, the performance has no significant improvement compared to the 2-patches experiment. This can likely be attributed to the difference in data volume: in the same amount of time, we can generate 20,000 training samples with 2-patches but only thouthands of training samples with 6-patches. This also illustrates an advantage of this patch-selection method: it achieves good prediction results with less information and computational effort.

	Accuracy	Precision	Recall	F1 Score
Baseline	51.8%	0.5571	0.5180	0.4876
Basic	61.0%	0.6332	0.6100	0.5885
2-Patches	64.8%	0.6851	0.6480	0.6560

(a) DenseNet-121

	Accuracy	Precision	Recall	F1 Score
Baseline	25.8%	0.3276	0.2580	0.2410
Basic	29.0%	0.3513	0.2900	0.2690
2-Patches	29.6%	0.3550	0.2960	0.2934

(b) VGG-19

	Accuracy	Precision	Recall	F1 Score
Baseline	20.0%	0.1369	0.2000	0.0959
Basic	29.0%	0.2675	0.2900	0.2436
2-Patches	31.7%	0.3169	0.2960	0.2747

(c) ResNet-50

Table 4. Evaluation results of the object-detecting patch selection

5. Conclusion

Our project demonstrates that combining DNNs with an SNN adapter is effective for art style classification. The hierarchical architecture we propose improves classification accuracy and stability by leveraging both global and local features. We also find that object-detecting patch selection enhances performance while requiring less computation. Future work could refine patch selection strategies, integrate attention mechanisms to help the model focus on relevant features, and test the method on larger datasets to further validate its robustness. These extensions could broaden the application of our approach in art history and cultural heritage preservation.

References

[1] Moritz Alkofer. Using convolutional neural networks to compare paintings by style. Intersect: The Stanford Journal of Science, Technology, and Society, 15 (1), 2021. 1

[2] Sinem Aslan and Luc Steels. Aligning figurative paintings with their sources for semantic interpretation. International Journal of Interactive Multimedia and Artificial Intelligence, In Press:1, 2023. 2

[3] Eva Cetinic, Tomislav Lipic, and Sonja Grgic. Fine-tuning convolutional neural networks for fine art classification. Expert Systems with Applications, 114:107–118, 2018. 1, 2

[4] Rendi Chevi. Artnet - painting style classification and feature visualization. https://github.com/rendchevi/artnet-app. Accessed: 2025-04-24. 2, 4, 5

[5] Kayra Coşkun. Art style classification with transfer learning (vgg16 and resnet50v2). https://github.com/kayracoskun/Art_Style_Classification/tree/main, 2023. 6

[6] Christian Hentschel, Timur Wiradarma, and Harald Sack. Fine tuning cnns with scarce training data — adapting imagenet to art epoch classification. pages 3693–3697, 2016. 2

[7] Kai-Lung Hua, Trang-Thi Ho, Kevin-Alfianto Jangtjik, Yu-Jen Chen, and Mei-Chen Yeh. Artist-based painting classification using markov random fields with convolution neural network. Multimedia Tools and Applications, 79:12635–12658, 2020. 2, 4

[8] Lazaros Alexios Iliadis, Spyridon Nikolaidis, Panagiotis Sarigiannidis, Shaohua Wan, and Sotirios K. Goudos. Artwork style recognition using vision transformers and mlp mixer. Technologies, 10(1), 2022. 2

[9] Saqib Imran, Rizwan Ali Naqvi, Muhammad Sajid, Tauqeer Safdar Malik, Saif Ullah, Syed Atif Moqurrab, and Dong Keon Yon. Artistic style recognition: combining deep and shallow neural networks for painting classification. Mathematics, 11(22):4564, 2023. 1, 2, 3, 4

[10] Kevin Alfianto Jangtjik, Trang Thi Ho, Mei Chen Yeh, and Kai Lung Hua. A cnn-lstm framework for authorship classification of paintings. In 2017 IEEE International Conference on Image Processing, ICIP 2017 - Proceedings, pages 2866–2870. IEEE Computer Society, 2017. Publisher Copyright: © 2017 IEEE.; 24th IEEE International Conference on Image Processing, ICIP 2017 ; Conference date: 17-09-2017 Through 20-09-2017. 2

[11] Glenn Jocher and the Ultralytics Team. Ultralytics yolov5. https://github.com/ultralytics/yolov5. Zenodo DOI: 10.5281/zenodo.3908559. Accessed: 2025-05-24. 2, 4, 7

[12] Kaggle. Painter by numbers. https://www.kaggle.com/c/painter-by-numbers/data. Accessed: 2025-04-24. 4

[13] Weiwei Li. Enhanced automated art curation using supervised modified cnn for art style classification. Scientific Reports, 15(1):7319, 2025. 1

[14] Ivan Nunez-Garcia, Rocio Lizarraga-Morales, Uriel Hernandez-Belmonte, Victor jimenez Arredondo, and Alberto Lopez-Alanis. Classification of paintings by artistic style using color and texture features. Computación y Sistemas, 26, 2022. 2

[15] Alexis Paul and C. Malathy. An innovative approach for automatic genre-based fine art painting classification. 706:19–27, 2018. 2

[16] Qchaldemer. painting-classification: Classifier of paintings depending on their style (impressionism, barocco, …). https://github.com/qchaldemer/painting-classification/tree/master, 2023. 6

[17] Babak Saleh and Ahmed Elgammal. Large-scale classification of fine-art paintings: Learning the right metric on the right feature. 2015. 2

[18] Ludovica Schaerf, Carina Popovici, and Eric Postma. Art authentication with vision transformers, 2023. 2

Appendix A: Art Style Taxonomy

In our experiments, we categorized artworks into broader style groups, each encompassing several sub-styles. This classification is based on our interpretation and available literature. Any corrections and advices are welcomes.

DenseNet-121

The DenseNet-121 model classifies artworks into the following 5 style groups:

- Cubism: Cubism, Tubism, Cubo-Expressionism, Mechanistic Cubism, Analytical Cubism, Cubo-Futurism, Synthetic Cubism
- Impressionism: Impressionism, Post-Impressionism, Synthetism, Divisionism, Cloisonnism
- Expressionism: Expressionism, Neo-Expressionism, Figurative Expressionism, Fauvism
- Realism: Realism, Hyper-Realism, Photorealism, Analytical Realism, Naturalism
- Abstract: Abstract Art, New Casualism, Post-Minimalism, Orphism, Constructivism, Lettrism, Neo-Concretism, Suprematism, Spatialism, Conceptual Art, Tachisme, Post-Painterly Abstraction, Neoplasticism, Precisionism, Hard Edge Painting

VGG-19

The VGG-19 model classifies artworks into the following 9 style groups:

- Art Nouveau (Modern): Art Nouveau (Modern)
- Baroque: Baroque
- Expressionism: Expressionism, Neo-Expressionism, Figurative Expressionism, Fauvism, Cubism, Tubism, Cubo-Expressionism, Mechanistic Cubism, Analytical Cubism, Cubo-Futurism, Synthetic Cubism
- Impressionism: Impressionism, Synthetism, Divisionism, Cloisonnism
- Post-Impressionism: Post-Impressionism
- Rococo: Rococo
- Romanticism: Romanticism, Realism, Hyper-Realism, Photorealism, Analytical Realism, Naturalism
- Surrealism: Surrealism, Abstract Art, New Casualism, Post-Minimalism, Orphism, Constructivism, Lettrism, Neo-Concretism, Suprematism, Spatialism, Conceptual Art, Tachisme, Post-Painterly Abstraction, Neoplasticism, Precisionism, Hard Edge Painting
- Symbolism: Symbolism

ResNet-50

The ResNet-50 model classifies artworks into the following 5 style groups:

- Cubism: Cubism, Tubism, Cubo-Expressionism, Mechanistic Cubism, Analytical Cubism, Cubo-Futurism, Synthetic Cubism
- Impressionism: Impressionism, Post-Impressionism, Synthetism, Divisionism, Cloisonnism
- Expressionism: Expressionism, Neo-Expressionism, Figurative Expressionism, Fauvism
- Realism: Realism, Hyper-Realism, Photorealism, Analytical Realism, Naturalism
- Abstract Art: Abstract Art, New Casualism, Post-Minimalism, Orphism, Constructivism, Lettrism, Neo-Concretism, Suprematism, Spatialism, Conceptual Art, Tachisme, Post-Painterly Abstraction, Neoplasticism, Precisionism, Hard Edge Painting

Appendix B: Additional Results

In this part, we present the evaluate results for each classes in each experiment settings.

DenseNet-121

Class	Accuracy	Precision	Recall	F1 Score
Cubism	49%	0.8033	0.4900	0.6087
Expressionism	11%	0.2500	0.1100	0.1528
Impressionism	34%	0.7556	0.3400	0.4690
Realism	93%	0.4227	0.9300	0.5813
Abstract	72%	0.5538	0.7200	0.6261

Table 5. Per-class results for the Baseline DenseNet-121 model

Class	Accuracy	Precision	Recall	F1 Score
Cubism	54%	0.8852	0.5400	0.6708
Expressionism	49%	0.3740	0.4900	0.4242
Impressionism	59%	0.5728	0.5900	0.5813
Realism	84%	0.6942	0.8400	0.7602
Abstract	65%	0.7738	0.6500	0.7065

Table 6. Per-class results for the Basic SNN adapter

Class	Accuracy	Precision	Recall	F1 Score
Cubism	53%	0.8833	0.5300	0.6625
Expressionism	45%	0.3814	0.4500	0.4128
Impressionism	55%	0.5978	0.5500	0.5729
Realism	90%	0.6818	0.9000	0.7759
Abstract	71%	0.7245	0.7100	0.7172

Table 7. Per-class results for the Hierarchy SNN architecture ($\lambda = 10$)

Class	Accuracy	Precision	Recall	F1 Score
Cubism	70%	0.8861	0.7000	0.7821
Expressionism	52%	0.4094	0.5200	0.4581
Impressionism	61%	0.6224	0.6100	0.6162
Realism	81%	0.6378	0.8100	0.7137
Abstract	60%	0.8696	0.6000	0.7101

Table 8. Per-class results for Object-Patch SNN adapter

VGG-19

Class	Accuracy	Precision	Recall	F1 Score
Art Nouveau (Modern)	nan*	0.0000	0.0000	0.0000
Baroque	6%	0.6364	0.0700	0.1261
Expressionism	31%	0.3605	0.3100	0.3333
Impressionism	42%	0.3165	0.4237	0.3623
Post-Impressionism	15%	0.1818	0.1463	0.1622
Rococo	nan*	0.0000	0.0000	0.0000
Romanticism	nan*	0.0000	0.0000	0.0000
Surrealism	60%	0.3797	0.6000	0.4651
Symbolism	0%	0.0000	0.0000	0.0000

Table 9. Per-class results for the Baseline VGG-19 model

* There is no painting in this class in the test set.

Class	Accuracy	Precision	Recall	F1 Score
Art Nouveau (Modern)	nan	0.0000	0.0000	0.0000
Baroque	6%	0.4615	0.0600	0.1062
Expressionism	47%	0.3333	0.4700	0.3900
Impressionism	56%	0.3204	0.5593	0.4074
Post-Impressionism	27%	0.2619	0.2683	0.2651
Rococo	nan	0.0000	0.0000	0.0000
Romanticism	nan	0.0000	0.0000	0.0000
Surrealism	43%	0.4019	0.4300	0.4155
Symbolism	5%	0.2632	0.0500	0.0840

Table 10. Per-class results for the Basic SNN adapter

Class	Accuracy	Precision	Recall	F1 Score
Art Nouveau (Modern)	nan	0.0000	0.0000	0.0000
Baroque	8%	0.7273	0.0800	0.1441
Expressionism	45%	0.3879	0.4500	0.4167
Impressionism	51%	0.3529	0.5085	0.4167
Post-Impressionism	22%	0.2727	0.2195	0.2432
Rococo	nan	0.0000	0.0000	0.0000
Romanticism	nan	0.0000	0.0000	0.0000
Surrealism	55%	0.3929	0.5500	0.4583
Symbolism	1%	0.0476	0.0100	0.0165

Table 11. Per-class results for the Hierarchy SNN architecture ($\lambda = 10$)

Class	Accuracy	Precision	Recall	F1 Score
Art Nouveau (Modern)	nan	0.0000	0.0000	0.0000
Baroque	12%	0.5714	0.1200	0.1983
Expressionism	37%	0.3592	0.3700	0.3645
Impressionism	45.76%	0.4500	0.4576	0.4538
Post-Impressionism	41.46%	0.3400	0.4146	0.3736
Rococo	nan	0.0000	0.0000	0.0000
Romanticism	nan	0.0000	0.0000	0.0000
Surrealism	53%	0.4015	0.5300	0.4569
Symbolism	2%	0.0377	0.0200	0.0261

Table 12. Per-class results for Object-Patch SNN adapter

ResNet-50

Table 13. ResNet50 Baseline (Accuracy 20.0%)

Class	Accuracy	Precision	Recall	F1 Score
Abstract Art	0%	0.0000	0.0000	0.0000
Cubism	91%	0.2004	0.9100	0.3285
Expressionism	4%	0.2667	0.0400	0.0696
Impressionism	0%	0.0000	0.0000	0.0000
Realism	5%	0.2174	0.0500	0.0813

Table 14. ResNet50 ShallowNN (Accuracy 29.0%)

Class	Accuracy	Precision	Recall	F1 Score
Abstract Art	21%	0.5000	0.2100	0.2958
Cubism	0%	0.0000	0.0000	0.0000
Expressionism	26%	0.2857	0.2600	0.2723
Impressionism	74%	0.2624	0.7400	0.3874
Realism	24%	0.2892	0.2400	0.2623

Table 15. ResNet50 Hierarchy (Accuracy 25.8%)

Class	Precision	Recall	F1 Score	Support
Abstract Art	0.3305	0.3900	0.3578	100
Cubism	0.2759	0.2400	0.2567	100
Expressionism	0.2481	0.3300	0.2833	100
Impressionism	0.2600	0.1300	0.1733	100
Realism	0.1786	0.2000	0.1887	100

Table 16. ResNet50 Object-Patch SNN (Accuracy 31.60%)

Class	Precision	Recall	F1 Score	Support
Abstract Art	0.6190	0.2600	0.3662	100
Cubism	0.0000	0.0000	0.0000	100
Expressionism	0.2357	0.6600	0.3474	100
Impressionism	0.4316	0.4100	0.4205	100
Realism	0.3012	0.2500	0.2732	100