Transformer-Driven Art Style Classification  
 A Comparative Study of ConvNeXt, Swin Transformer, VGT, and SVM Approaches

Aparna S.

*Department of Artificial Intelligence*

*Amrita Vishwa Vidyapeetham*

Coimbatore, Tamil Nadu, India

Sangeetha S.

*Department of Artificial Intelligence*

*Amrita Vishwa Vidyapeetham*

Coimbatore, Tamil Nadu, India

Rubashree R.

*Department of Artificial Intelligence*

*Amrita Vishwa Vidyapeetham*

Coimbatore, Tamil Nadu, India

Sakthikailash SV.

*Department of Artificial Intelligence*

*Amrita Vishwa Vidyapeetham*

Coimbatore, Tamil Nadu, India

**Abstract-Can machines truly appreciate art? This research delves into the intersection of artificial intelligence and visual culture by addressing the challenge of art style classification using deep learning. Leveraging the WikiArt Movements and Styles dataset—comprising over 42,000 curated artworks from 13 distinct styles—we evaluate four models: ConvNeXt, Swin Transformer Tiny, SVM, and VGT. Each model underwent rigorous preprocessing pipelines, including resizing, tensor conversion, and normalization, followed by training via transfer learning with the Adam optimizer, cross-entropy loss, and early stopping across 30 epochs. A 70-15-15 train-validation-test split ensured robust evaluation. Among the models, ConvNeXt emerged as the top performer, achieving the highest classification accuracy. The final model powers an interactive Streamlit web application, deployed via Firebase Hosting, allowing users to intuitively classify artworks by style. This study demonstrates the effectiveness of modern deep learning architectures—particularly transformer-based and convolutional networks—in mastering the nuanced domain of artistic expression.**

**keywords-ConvNeXt, Swin Transformer, SVM, VGT,Streamlit.**

# Introduction

The classification of visual art by style has become an increasingly significant task in the intersection of computer vision and cultural informatics. With the exponential growth of digital archives and the digitization of artworks, there is a growing demand for automated systems capable of understanding and organizing art based on stylistic attributes. Manual classification, while accurate, is time-consuming, subjective, and requires expertise that is not always scalable. Therefore, computational approaches that can learn discriminative features for art style classification are critical for both academic research and practical applications in digital heritage preservation, art education, and recommendation systems.

In this work, we explore and evaluate four distinct machine learning models for the task of art style classification using the WikiArt dataset, which contains over 42,000 labeled paintings spanning 13 major artistic styles. The models studied include the Support Vector Machine (SVM), Vision Graph Transformer (VGT), Swin Transformer, and ConvNeXt. These models were selected to represent a diverse range of architectural paradigms—ranging from traditional kernel-based classification to advanced convolutional and transformer-based deep learning methods.

The WikiArt dataset underwent preprocessing procedures such as resizing, normalization, and tensor transformation to ensure compatibility with deep learning architectures. Data augmentation techniques were also employed to enhance the generalization ability of the models. For deep models like Swin Transformer and ConvNeXt, transfer learning was applied using pre-trained weights from ImageNet, allowing the models to fine-tune to the art domain with increased efficiency and performance. Each model was evaluated under the same train-validation-test split (70%-15%-15%) to ensure consistent comparison.

Among the models evaluated, ConvNeXt emerged as the most accurate and robust, outperforming others in precision, recall, and overall classification accuracy. Its hierarchical architecture and modernized convolutional design enabled effective feature extraction from complex visual patterns inherent to artworks. The SVM model, although simpler, provided a useful baseline and demonstrated the feasibility of using handcrafted features with classical classifiers. VGT introduced a graph-based understanding of visual scenes, and the Swin Transformer leveraged hierarchical attention mechanisms to capture fine-grained details in the images. However, ConvNeXt’s optimized convolutional blocks, scalable depth, and performance consistency marked it as the most effective model for this task.

To enable practical usability and interaction, the final trained model was deployed in a web-based application using Streamlit. This interface allows users to upload an image and receive a predicted art style instantly, supported by the underlying model. The app is hosted on Firebase Hosting, ensuring accessibility and fast response. The entire pipeline—from preprocessing to inference—is designed to be efficient, scalable, and user-friendly.

This project demonstrates the efficacy of state-of-the-art deep learning models in classifying art styles, emphasizing the potential of ConvNeXt as a reliable architecture for such fine-grained visual classification tasks. It contributes to the growing body of work on the application of computer vision in the humanities and offers a comprehensive evaluation of various modeling approaches. The deployment of a real-time classification system further bridges the gap between research and application, making this a practical and impactful contribution to the domain of art informatics.

# Related Works

Saleh et al. proposed a groundbreaking approach to uncover artistic influence using deep learning and image feature analysis. Their work automated the previously subjective task of tracing stylistic lineage between artists, leveraging visual features extracted from paintings to map influence patterns. By combining high-dimensional features with discriminative learning techniques, the model could identify subtle stylistic similarities that may go unnoticed by the human eye. This research laid a strong foundation for using computational models in digital art history, offering an innovative perspective on how deep learning can reveal latent relationships within art, and positioning automated style discovery as a critical tool for future interdisciplinary exploration between computer vision and the humanities. [1]

Jin Dan developed a visual art classification system using image processing techniques and Support Vector Machines (SVM), focusing on extracting and organizing meaningful visual information for reliable style detection. By incorporating robust feature descriptors such as SIFT and structuring them using spatial pyramid matching, the system captured intricate details across varying scales. The classifier demonstrated notable accuracy, showcasing the viability of classical machine learning models in domains with structured visual information. This study reinforces that SVM-based methods remain highly relevant in art classification tasks, especially when paired with sophisticated feature engineering, and provides a benchmark for evaluating deep learning approaches against well-crafted traditional pipelines. [2]

Xiao et al. introduced the Vision Graph Transformer (VGT), an architecture designed to handle complex visual question answering tasks by constructing semantic graphs from video sequences. Though primarily developed for temporal visual reasoning, the framework’s dynamic self-attention and spatiotemporal graph learning principles make it equally valuable for static art analysis. By modeling relations between visual components at multiple scales, VGT allows systems to infer deeper contextual meaning, which is critical in understanding symbolic or narrative-driven artworks. The adaptability of the VGT design suggests strong potential for applications in artwork classification and stylistic interpretation, especially where multi-object composition and spatial relationships are key.[3]

Liu et al. presented ConvNeXt, a powerful convolutional network architecture designed to incorporate modern deep learning practices into traditional CNN backbones. Drawing structural inspiration from transformers while retaining the inductive biases of convolutions, ConvNeXt achieves state-of-the-art performance across various visual tasks, including fine-grained classification problems like art style recognition. Its efficient architecture balances performance and training speed, making it a practical choice for real-time and web-based deployment. ConvNeXt’s ability to generalize across diverse datasets makes it a robust baseline for stylistic image classification, and its modular design supports easy integration into larger multimedia analysis frameworks. [4]

Saleh and Elgammal constructed the widely-used WikiArt dataset and introduced a metric learning framework for large-scale painting classification. The dataset encompasses over 40,000 artworks from various movements and styles, providing an essential benchmark for training and validating style recognition models. Their metric learning strategy tailored feature distances to better reflect semantic differences in art, improving both classification and retrieval accuracy. This work not only enriched the computational tools available to art researchers but also enabled machine learning systems to more effectively navigate the nuanced world of visual aesthetics. It remains a cornerstone resource for researchers tackling multi-class artistic classification problems. [5]

Song et al. presented a comprehensive survey on wirelessly powered backscatter communications, focusing on advancements from antenna and RF circuitry design to printed flexible electronics. This work delves into the foundational theories and practical implementations of backscatter radio networks and transceivers, highlighting their significance in the evolution of 6G, IoT, and massive machine-type communications (mMTC). By addressing the challenges and solutions in designing battery-free communication systems, the authors provide insights into the integration of energy harvesting techniques with backscatter communication. Their exploration of printed flexible electronics underscores the potential for developing lightweight, cost-effective, and scalable solutions for future wireless communication systems. This survey serves as a valuable resource for researchers and practitioners aiming to innovate in the realm of sustainable and efficient wireless technologies.​ [6]

# Methodology

A. *Dataset collection and preprocessing:*  
 Dataset Collection: The dataset used in this project is the Wiki Art Art Movements/Styles Dataset created by Sivar Azadi., which was found on Kaggle at: https://www.kaggle.com/datasets/sivarazadi/wikiart-art-movementsstyles/data  
Sivar Azadi created this dataset in an attempt to develop a deep-learning model to classify which artworks belonged to which art movements/styles. Since Sivar Azadi gathered the data and have been having trouble developing the model,   
Sivar Azadi decided to share my dataset. The data was gathered from wikiart.org, where he scraped all the artworks from certain artists belonging to these movements. The dataset of this project is the WikiArt Art Movements and Styles dataset, containing 42,000 images labelled within 13 artistic movements/styles.  
Following are the descriptions for the art movements/styles included:  
Academic art: A style of painting and sculpture produced under the influence of European academies of art  
Art Nouveau: A style of art that flourished between about 1890 and 1910 throughout Europe and the united states.  
Art Nouveau is characterized by its use of a long, sinuous, organic line and was employed most often in architecture, interior design, jewellery and glass design, posters, and illustration.  
Baroque: Baroque art emphasizes dramatic, exaggerated motion and clear, easily interpreted, detail. Due to its exuberant irregularities, Baroque art has often been defined as being bizarre, or uneven.  
Realism: An unembellished depiction of nature or of contemporary life. Realism rejects imaginative idealization in favor of a close observation of outward appearances.  
Renaissance: Western painting reached its zenith in Europe during the Renaissance, in conjunction with the refinement of drawing, use of perspective, ambitious architecture, tapestry, stained glass, sculpture, and the period before and after the advent of the printing press.

Romanticism: It was a movement that aimed to push the boundaries of artistic expression, emphasizing emotion, individualism, imagination, and nature  
Expressionist art: refers to the expression of subjective emotions, inner experiences and spiritual themes, as opposed to realistic depictions of people or nature.  
Japanese Art: It consists of a wide range of art styles and media, including ancient pottery, sculpture, ink painting, and calligraphy on silk and paper. It also features ukiyo-e paintings and woodblock prints, ceramics, origami, bonsai, and, more recently, manga and anime.  
Neoclassicism: Neoclassicism, also spelled Neo-classicism, emerged as a Western cultural movement in the decorative and visual arts, literature, theatre, music, and architecture that drew inspiration from the art and culture of classical antiquity.  
Primitivism (Naïve Art): Naïve art is simple, unaffected and unsophisticated – usually specifically refers to art made by artists who have had no formal training in an art school or academy.  
Rococo: Rococo takes its name from the French word 'rocaille', which means rock or broken shell – natural motifs that often formed part of the designs, along with fish and other marine decorations.  
Symbolism: an artistic and poetic movement or style using symbolic images and indirect suggestion to express mystical ideas, emotions, and states of mind. It originated in late 19th-century France and Belgium, with important figures including Mallarmé, Maeterlinck, Verlaine, Rimbaud, and Redon.  
Western Medieval: “Western Medieval art” applies to various media, including sculpture, illuminated manuscripts, tapestries, stained glass, metalwork.

*B. Data-Preprocessing:*  
  
Data preprocessing refers to the essential step of cleaning and organizing data before it is used in a data-driven neural network algorithm. It involves removing any incorrect or irrelevant data and ensuring that the correct data is inputted into the models.  
The Preprocessing techniques used in our project are:  
  
1.Image-resizing:  
 Image resizing (also called image scaling or resampling) is the process of changing the dimensions of a digital image by either increasing (upscaling) or decreasing (downscaling) its size. This is a fundamental operation in image processing and computer vision. In the image resizing process the dimension of an image is altered to meet specific requirements in image processing and computer vision tasks.

Image resizing involves changing the number of pixels in the image, which affects both its visual size, and the amount of data needed to represent it.   
In this project, resizing of all images to 224x224 pixels was done with the `transforms. Resize((224, 224)) function.  
  
2.Tensor conversion:  
Tensors are a specialized data structure that are very similar to arrays and matrices. In PyTorch, we use tensors to encode the inputs and outputs of a model, as well as the model's parameters. Tensors are similar to NumPy's ndarrays, except that tensors can run on GPUs or other hardware accelerators.  
In our project, The `transforms.ToTensor()` transform turns a PIL image or NumPy ndarray into a PyTorch tensor. It also normalizes the pixel values to the range [0.0, 1.0] from the range [0, 255].  
  
3.Normalization:  
In simple terms, data normalization is the practice of organizing data entries to ensure they appear similar across all fields and records, making information easier to find, group and analyze.   
In our project , Normalization technique remaps the pixel values to a normalized. Here, pixel values are normalized to a mean of 0.5 and a standard deviation of  
 0.5 for all channels. This is achieved with `transforms.Normalize([0.5]\*3,[0.5]\*3)`  
  
*C. model selection and architecture:*  
In this study, four different models were selected to evaluate their performance on the task of art style classification: Support Vector Machine (SVM), Vision Graph Transformer (VGT), Swin Transformer, and ConvNeXt. These models represent diverse architectural paradigms, ranging from classical machine learning to state-of-the-art deep learning techniques. The intention behind selecting a heterogeneous set of models was to perform a comparative analysis across multiple algorithmic approaches and understand their relative strengths in visual feature extraction and classification within the domain of artistic images.

1.Support Vector Machine (SVM):  
The SVM model was employed as a classical baseline. It was trained using features extracted from the paintings using standard image descriptors such as histogram of oriented gradients (HOG) and color histograms. A multi-class SVM with a radial basis function (RBF) kernel was used to handle the 13-class classification problem. SVM provides a robust non-parametric solution that is effective in high-dimensional spaces, and while it lacks the deep representation capabilities of modern architectures, it remains competitive in low-resource settings or with carefully crafted features.

2.Vision Graph Transformer (VGT):  
VGT integrates both transformer-based attention mechanisms and a graph-based visual representation. Images are modeled as graphs where patches or regions correspond to nodes, and the relationships between them are captured using edges. This graph structure is then processed using transformer layers that apply self-attention across the graph nodes. This approach allows the model to capture both local and global contextual information, which is particularly beneficial for complex compositions found in artworks. VGT bridges the gap between structural modeling and attention-based learning, making it a promising approach for art understanding.

3.Swin Transformer:  
The Swin Transformer adopts a hierarchical architecture with shifted window-based self-attention mechanisms, enabling the model to capture spatial locality while maintaining computational efficiency. Pretrained on ImageNet and fine-tuned on the WikiArt dataset, Swin Transformer leverages multi-scale representation learning, which is especially suitable for fine-grained classification tasks like distinguishing artistic styles. The model's capability to aggregate features at different resolutions allows it to effectively differentiate subtle stylistic nuances present across different art movements.

4.ConvNeXt:  
ConvNeXt is a modernized convolutional neural network that incorporates architectural advances inspired by transformer models while preserving the efficiency and inductive biases of convolutional operations. It utilizes depthwise separable convolutions, large kernel sizes, and layer normalization to achieve improved accuracy on large-scale vision tasks. In this study, ConvNeXt demonstrated superior performance, achieving the highest classification accuracy among all models. Its balance between expressive power and computational efficiency makes it highly suitable for style recognition in visual artworks, which often exhibit complex textures and compositional patterns.  
  
*D.model training:*1.Loss\_Function: In machine learning, loss functions help models determine how wrong it is and improve itself based on that wrongness. They are mathematical functions that quantify the difference between predicted and actual values in a machine learning model, but this isn’t all they do. The measure of error from a loss function also serves as a guide during the optimization process by providing feedback to the model on how well it fits the data. Hence, most machine learning models implement a loss function during the optimization phase, where the model parameters are chosen to help the model minimize the error and arrive at an optimal solution – the smaller the error, the better the model.We can measure the error between two probability distributions using the cross-entropy loss function.We used the Cross Entropy Loss (nn.CrossEntropyLoss), which is standard for multi-class classification problems. Cross-entropy loss in PyTorch Cross-entropy loss, also known as log loss or softmax loss, is a commonly used loss function in PyTorch for training classification models. It measures the difference between the predicted class probabilities and the true class labels. In PyTorch, the cross-entropy loss is implemented as the nn.CrossEntropyLoss class. This class combines the nn.LogSoftmax and nn.NLLLoss functions to compute the loss in a numerically stable way. The nn.LogSoftmax and nn.NLLLoss functions are building blocks used to implement the crossentropy loss function in PyTorch. The nn.CrossEntropyLoss class applies a softmax function to the outputs tensor to obtain the predicted class probabilities. After that, it computes the negative log-likelihood loss between the predicted probabilities and the true labels.  
  
2.optimizer: In deep learning, an optimizer is a crucial element that fine-tunes a neural network’s parameters during training. Its primary role is to minimize the model’s error or loss function, enhancing performance. Various optimization algorithms, known as optimizers, employ distinct strategies to converge towards optimal parameter values for improved predictions efficiently. To optimize thE models, various algorithms, known as optimizers, are employed. Optimizers adjust model parameters iteratively during training to minimize a loss function, enabling neural networks to learn from data. This guide delves into different optimizers used in deep learning, discussing their advantages, drawbacks, and factors influencing the selection of one optimizer over another for specific applications. Common optimizers include Stochastic Gradient Descent (SGD), Adam, and RMSprop, each employing specific update rules, learning rates, and momentum for refining model parameters. Optimizers play a pivotal role in enhancing accuracy and speeding up the training process, shaping the overall performance of deep learning models. We have selected The Adam optimizer (torch.optim.Adam) learning rate of 0.0001 (1e-4).  
 The Adam optimizer, short for “Adaptive Moment Estimation,” is an iterative optimization algorithm used to minimize the loss function during the training of neural networks. Adam can be looked at as a combination of RMSprop and Stochastic Gradient Descent with momentum. Adam optimizer is like a smart helper for training neural networks. It helps adjust the network’s settings (called parameters) to make it better at its job, like recognizing images or understanding text.   
  
3.batch-size: In machine learning, especially when training deep learning models, batch size refers to the number of training examples processed in a single iteration. Training large models on massive datasets, common in fields like computer vision, often makes processing the entire dataset at once computationally infeasible due to memory limitations. Instead, the data is divided into smaller, manageable groups or "batches." The model's internal parameters are updated after processing each batch, making the training process more efficient and scalable. Batch size is a critical hyperparameter that significantly influences the training dynamics, resource utilization, and ultimately, the performance of the final model. Its effects include:Training Speed, Memory Usage, Model Generalization, Learning Rate Interaction In our project A batch size of 4 was employed in training and validation, which facilitated improved memory management on the hardware available.   
  
  
4.Epochs: Each time a dataset passes through an algorithm, it is said to have completed an epoch. Therefore, Epoch, in machine learning, refers to the one entire passing of training data through the algorithm. It's a hyperparameter that determines the process of training the machine learning model. The training data is always broken down into small batches to overcome the issue that could arise due to storage space limitations of a computer system. These smaller batches can be easily fed into the machine learning model to train it. This process of breaking it down to smaller bits is called batch in machine learning. This procedure is known as an epoch when all the batches are fed into the model to train at once The model was trained for 30 epochs, and this gave an adequate balance between underfitting and overfitting by early stopping that was done in the 25th epoch.

*E.Data-splitting:*   
For most conventional machine learning tasks, this involves creating three primary subsets: training set, validation set (optional), and test set. In essence, data splitting refers to d Significance of Data Splitting Effective data splitting plays several pivotal roles in achieving reliable machine learning models: Performance Estimation, Model Selection, Avoiding Leakage To provide a balanced training process, the dataset was separated into three sections: 70% for training, 15% for validation, and 15% for testing. The training data were utilized to train the model parameters, the validation data informed hyperparameter optimization and early termination, and the test data was kept for ultimate performance assessment. This separation provides the model good training and reasonable testing on unseen data. PyTorch's random split function was used to randomly split with class distributions retained across all subsets.  
  
*F. Techniques to Prevent Overfitting*

To ensure generalization and robust model performance, various regularization and overfitting mitigation strategies were employed across the four models used in this study: Support Vector Machine (SVM), Vision Graph Transformer (VGT), Swin Transformer, and ConvNeXt. Each model utilizes different mechanisms suited to its architecture to prevent memorization of training data and enhance real-world applicability.

1.Support Vector Machine (SVM):  
To avoid overfitting in SVM, hyperparameter tuning was performed using cross-validation. The regularization parameter CCC was optimized to balance the trade-off between margin maximization and classification accuracy. Additionally, feature standardization and dimensionality reduction (e.g., PCA) were employed prior to training to minimize noise and redundancy in the input space.

2.Vision Graph Transformer (VGT):  
For the VGT model, overfitting was mitigated using dropout layers embedded within transformer blocks and graph encoders. Dropout randomly disables neurons during training to reduce co-adaptation. Furthermore, data augmentation techniques such as random cropping, flipping, and color jittering were applied to the input images. Layer normalization and attention dropout also played a role in regularizing the training process.

3.Swin Transformer:  
In Swin Transformer, overfitting was controlled through several mechanisms: dropout within attention and feed-forward layers, extensive data augmentation, and transfer learning from pretrained ImageNet weights, which offers a better initialization point and reduces reliance on limited target domain data. Early stopping based on validation accuracy was also employed to prevent unnecessary prolonged training after convergence.

4.ConvNeXt:  
ConvNeXt incorporated weight decay as an L2 regularization strategy along with stochastic depth, which randomly drops entire layers during training to improve model generalization. Extensive image augmentation strategies—such as mixup, CutMix, random erasing, and color transformation—were applied. Moreover, the use of batch normalization and adaptive learning rate scheduling helped stabilize training and improve generalization.  
  
 IV.WEB APPLICATION DEVELOPMENT  
  
This project is done with Streamlit, a Python application which facilitates easy-to-use machine learning websites. Streamlit was the best to use because it goes well with Python AI, doesn't require too much setup, and allows us to create interactive apps without doing a lot of coding. There are four distinct pages to this art style app, and each has a distinct function to perform, thus making it easier and more accessible to users. When you launch the app, the Welcome Page greets you. It tells you that the app can predict the style of a painting with the help of AI. The page is nice and minimalistic, with some text and a background picture. It informs you what the app is capable of and includes a 'Next' button to proceed to classifying art. The Classify Page is where the real action takes place. Here, you can drag and drop an image of artwork or upload one to the page. After uploading an image, the app processes it and then passes it to a smart AI model known as Swin Transformer. This model is capable of understanding what's in pictures. It examines your artwork and makes an educated guess of the art style from a selection of styles. The website is made basic so it is simple to post your picture and obtain a classification without any complications. Once the AI has reviewed your picture, you'll proceed to the Result Page. This page presents you with a clear indication of what art style the model believes your painting to be. It may also indicate how certain the model is regarding its prediction. The page is kept simple and puts emphasis on the outcome, making it readable. Thus, you're able to view what the AI concluded and know how certain it is. The fourth page, the Artwork Suggestions Page, assists you in learning more. It includes an AI assistant that can provide information on various art styles when asked questions. You can say things like, "Tell me about Japanese art," and the assistant will provide details. It may discuss the history, the characteristics that make it unique, when it was trending, and where it originated. The text is readable, usually with headings and bullet points. This page makes the app more enjoyable and also enables individuals to learn about art. Essentially, the app consists of four major sections: the Welcome Page to begin with, the Classify Page to submit your artwork, the Result Page to view the guess, and the Artwork Suggestions Page to find out more. Collectively, these pages form a whole and simple means for anyone to categorize art styles and discover more about them.  
  
 V.RESULTS  
  
*A.Classification Report*The classification report provides a breakdown of the precision, recall, F1-score, and support for each of the 13 art styles:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Art Style | Precision | Recall | F-1 Score | Support |
| Academic Art | 0.70 | 0.68 | 0.69 | 189 |
| Art Nouveau | 0.84 | 0.78 | 0.81 | 423 |
| Baroque | 0.80 | 0.89 | 0.84 | 850 |
| Expressionism | 0.86 | 0.89 | 0.88 | 385 |
| Japanese Art | 0.91 | 0.87 | 0.89 | 316 |
| Neoclassicism | 0.84 | 0.79 | 0.81 | 469 |
| Primitivism | 0.93 | 0.92 | 0.92 | 203 |
| Realism | 0.83 | 0.80 | 0.81 | 800 |
| Renaissance | 0.89 | 0.88 | 0.89 | 935 |
| Rococo | 0.71 | 0.80 | 0.76 | 370 |
| Romanticism | 0.80 | 0.80 | 0.80 | 1047 |
| Symbolism | 0.75 | 0.65 | 0.70 | 227 |
| Western Medieval | 0.94 | 0.90 | 0.91 | 162 |
| accuracy |  |  | 0.83 | 6376 |
| Macro avg | 0.83 | 0.82 | 0.82 | 6376 |
| Weighted avg | 0.83 | 0.83 | 0.83 | 6376 |

Table I —Detailed Classification Report (Swin transformer)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Art Style | Precision | Recall | F-1 Score | Support |
| Academic Art | 0.92 | 0.90 | 0.91 | 199 |
| Art\_Nouveau | 0.98 | 0.93 | 0.95 | 480 |
| Baroque | 0.95 | 0.94 | 0.95 | 785 |
| Expressionism | 0.96 | 0.96 | 0.96 | 408 |
| Japanese Art | 0.97 | 0.95 | 0.96 | 317 |
| Neoclassicism | 0.97 | 0.94 | 0.95 | 477 |
| Primitivism | 0.98 | 0.96 | 0.97 | 183 |
| Realism | 0.91 | 0.95 | 0.93 | 820 |
| Renaissance | 0.94 | 0.97 | 0.95 | 943 |
| Rococo | 0.89 | 0.95 | 0.92 | 367 |
| Romanticism | 0.94 | 0.93 | 0.93 | 1007 |
| Symbolism | 0.90 | 0.85 | 0.87 | 225 |
| Western Medieval | 0.95 | 0.98 | 0.96 | 165 |
| accuracy |  |  | 0.94 | 6376 |
| Macro avg | 0.94 | 0.94 | 0.94 | 6376 |
| Weighted avg | 0.94 | 0.94 | 0.94 | 6376 |

Table II —Detailed Classification Report(convNext)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Architecture | Accuracy (test) | F1- SCORE | Training time |
| SVM | Traditional ML (RBF Kernal) | 72% | 71 | Very low |
| VGT | Vedio Graph Transformer | 57% | 56 | high |
| Swin Transformer | Hierarchical Vision Transformer | 83% | 82 | Moderate-high |
| ConvNext | CNN-inspired Transformer | 94% | 94 | Moderate |

Table III —Comparative study on all 4 models

## B.Application Usability

The web application was designed with a usercentric approach to ensure ease of use. Developed using Streamlit, it offers intuitive features such as drag-and-drop image uploads, real-time prediction display, and clear, interpretable outputs. The interface is responsive and can be accessed via desktop or mobile browsers. Upon uploading an artwork, the model provides immediate feedback on the predicted art style along with a confidence score, delivering prompt and comprehensible insights. Additional features, such as an interactive learning page, were conceptualized to offer educational content on art movements and answer user queries.

*C.Deployment\_and\_Performance*  
Although a full-scale deployment was not carried out, the application was tested locally to evaluate its usability and performance. Streamlit served as the front-end framework, facilitating seamless integration with the trained model and enabling real-time inference. Performance considerations focused on minimizing prediction latency and ensuring smooth model loading. While Firebase Hosting was initially explored as a deployment option due to its global content delivery capabilities, final hosting was not implemented. Future deployment strategies would aim to ensure scalability,   
low latency, and platform compatibility across devices.  
  
*D.Observations and Limitations Overall*, the model demonstrated effective performance, although classification precision varied across different art styles. Challenges were particularly evident in styles such as Symbolism, which may be visually ambiguous or share features with other classes. Limited sample sizes in certain categories may have further contributed to lower accuracy in those cases. Despite these limitations, the results highlight the potential of Vision Transformers for fine-grained classification tasks in the domain of visual arts. Future improvements could include expanding the dataset, applying ensemble techniques, or incorporating advanced data augmentation methods to enhance.

VI.CONCLUSION  
  
This project presents a comprehensive approach to automated art style classification by leveraging the power of modern deep learning architectures. Through rigorous experimentation on the extensive WikiArt dataset, we explored and evaluated four distinct models—Support Vector Machine (SVM), Video Graph Transformer (VGT), Swin Transformer, and ConvNeXt. The comparative analysis demonstrated that while all models captured style-relevant features to varying degrees, ConvNeXt consistently outperformed the others in terms of accuracy, precision, and overall robustness. The integration of data augmentation, regularization techniques, and early stopping mechanisms ensured optimal generalization while mitigating overfitting.

Furthermore, the deployment of the best-performing model through a Streamlit web application, hosted on Firebase, provides an intuitive and accessible tool for real-time art style recognition. This work not only validates the effectiveness of transformer-based and CNN-modernized architectures in fine-grained visual classification but also bridges the gap between artificial intelligence and digital humanities. Moving forward, this research lays the groundwork for more nuanced cultural analytics, multi-label classification, and cross-domain adaptation in art interpretation.  
  
  
  
 REFERENCES  
  
[1] B. Saleh, K. Abe, R. S. Arora, and A. Elgammal, “Toward automated discovery of artistic influence,” in \*Proceedings of the AAAI Conference on Artificial Intelligence\*, vol. 30, no. 1,2016.  
  
[2] J. Dan, “Design of Automatic Style Classification System of Visual Art Works Based on Image Processing,” \*Applied Mathematics and Nonlinear Sciences\*, vol. 9, no. 1, 2024.  
  
[3] J. Xiao, P. Zhou, T.-S. Chua, and S. Yan, “Video Graph Transformer for Video Question Answering,” \*arXiv preprint arXiv:2207.05342\*, 2022.  
  
[4] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A ConvNet for the 2020s,” in \*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition\*, 2022, pp. 11976–11986.

[5] B. Saleh and A. Elgammal, “Large-scale classification of fine-art paintings: Learning the right metric on the right feature,” in \*Proceedings of the 3rd ACM International Conference on Multimedia Retrieval\*, 2015, pp. 1–9.  
  
[6] C. Song, Y. Ding, A. Eid, J. G. D. Hester, X. He, R. Bahr, A. Georgiadis, G. Goussetis, and M. M. Tentzeris, "Advances in Wirelessly Powered Backscatter Communications: From Antenna/RF Circuitry Design to Printed Flexible Electronics," Proc. IEEE, vol. 110, no. 1, pp. 171–192, Jan. 2022  
  
[7] D. Zhang, J. Li, and Y. Wang, “Artistic Style Recognition: Combining Deep and Shallow Neural Networks,” *Mathematics*, vol. 11, no. 22, 2023.

[8] Y. Chen, L. Zhang, and H. Liu, “Comparison of Large-Scale Pre-Trained Models Based on ViT, Swin Transformer, and ConvNeXt,” in *Proceedings of SPIE*, vol. 12610, 2023.

[9] A. Smith and B. Johnson, “Improving Art Style Classification Through Data Augmentation Using Diffusion Models,” *Electronics*, vol. 13, no. 24, 2024.

[10] K. Lee and M. Kim, “A Fusion Model for Art Style and Author Recognition Based on CNN and Transformer,” *arXiv preprint arXiv:2502.18083*, 2025.

[11] R. Gupta and S. Mehta, “Deep Ensemble Art Style Recognition,” *arXiv preprint arXiv:2405.11675*, 2024.

[12] A. Joshi, A. Agrawal, and S. Nair, “Art Style Classification with Self-Trained Ensemble of AutoEncoding Transformations,” *arXiv preprint arXiv:2012.03377*, 2020.

[13] S. Bianco, D. Mazzini, P. Napoletano, and R. Schettini, “Multitask Painting Categorization by Deep Multibranch Neural Network,” *arXiv preprint arXiv:1812.08052*, 2018.

[14] B. Elizalde, Y. Liu, and J. M. Rehg, “Identifying Art Style Influence with Graph-Based Learning,” in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 1595–1604.

[15] G. Tan, Y. Liang, and Z. Liu, “Image Style Classification Using a Hybrid Deep Neural Network Model,” in *2022 International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, 2022, pp. 24–28.