Group 4 Final Report - Image Classification for Artistic Style.

Anthony Le, John Bute, Kelly Gao
SDS 3786
Patrick Boily
2024-12-18

1. Introduction

In the world of visual arts, an artistic style is a central component in identifying a work of art in terms of art history. The term refers to particular techniques and skills that characterize an artist's approach to creating art, including its similarities among artists with comparable works. Art styles place pieces and artists within specific cultural and historical contexts based on their visual attributes. Conversely, understanding these historical and cultural contexts, combined with visual details, allows historians to determine and classify artistic styles.

In a previous study, Mazzone *et al.* [1] demonstrated the feasibility of using machine learning to classify various art styles. Their work employed deep learning models to analyze a large dataset of paintings, utilizing quantifiable style patterns, methodologies, and principles rooted in traditional art history practices. The study leveraged an altered dataset of over 80,000 images among 27 artistic styles. Convolutional Neural Networks (CNNs) and Support Vector Machines (SVMs) were employed as feature extractors and classifiers, ultimately achieving an accuracy of 63%.

However, access to such extensive datasets and the computational power required to train deep learning models on them remains challenging for many researchers. To address this, we propose a focused approach that reduces the prediction classes to three specific art styles: Baroque, Contemporary Realism, and Expressionism. By narrowing the scope, we aim to demonstrate a framework for building a fast and specialized art classifier that operates effectively within constrained computational resources. We utilize ResNet50 for feature extraction and Principal Component Analysis (PCA) for dimensionality reduction. Furthermore, our paper explores various classifiers to identify the most optimal model for this task. This paper's results and findings highlight the potential of specialized art classification frameworks, which can achieve high accuracy while maintaining computational efficiency.

1. *Mazzone, M., Elgammal, A., Liu, B., & Kim, D. (2018). The shape of art history in the eyes of the machine. Retrieved from ResearchGate.*

2. Dataset and Preprocessing

Before classifying artwork, it is essential to gather paintings that correspond to each style. We utilized an online dataset, as shown below:

2.1 WikiArt-WikiPaintings dataset:

The WikiArt dataset was obtained from Kaggle. Although it has not been updated in over two years, this does not affect our analysis, as the art styles we are exploring are historical. Only if newly discovered paintings emerge will a more recent dataset version be necessary.

The WikiArt dataset contains over 80,000 high-quality art images with over 1,000 artists, 27 styles, and 45 genres. Fortunately, the dataset has also been organized into folders based on style, simplifying our work. Our styles have the following image count:

- **Expressionism**: 6,737 paintings
- **Baroque**: 4,240 paintings
- **Contemporary Realism**: 481 paintings

We have dissociated images from metadata such as artist name, year, and genre to ensure an unbiased evaluation. This separation allows the classification models to rely solely on the visual content of the paintings to distinguish between art styles. It should be noted that we have an imbalanced dataset (Contemporary Realism with 481 paintings) as we want to see how well models fare with rarer styles.

2.2 Preprocessing

Art consists of patterns, textures, shapes, and colors. These features are key in distinguishing styles like Baroque, Contemporary Realism, and Expressionism. While the human eye can identify them, replicating this process using machine learning requires the transformation of paintings into numerical features that capture these visual elements. Training a deep network from scratch requires significant computing power and data. To address this, we utilize ResNet50, a 50-layer deep residual network pre-trained on ImageNet, a dataset with millions of labeled images. Although ImageNet may focus more on objects than paintings, ResNet50 captures general visual features. These features often correspond to brushwork, color palettes, and themes inherent to different art styles. To leverage ResNet50, we resize our input images to 224x224 pixels for compatibility and remove the final classification layer of ResNet50 to transform it into a feature extractor. The output is a high-dimensional feature map (which we crushed into a 1D vector space) that contains essential visual features for each painting.

2.3 Class Classification Map

Our extracted features enable us to train our classification models. However, we must interpret the feature extraction process to understand what ResNet50 detects since a neural network acts as a black box where the decision-making process is not inherently transparent. Thus, we apply Class Activation Maps (CAM) directly to the output of the feature extractor. CAM generates heat maps highlighting regions in the paintings where ResNet detects the most relevant features. Firstly, this approach allows us to understand what areas of an image influence the model's feature extraction and what visual characteristics ResNet50 prioritizes depending on the style of the painting.

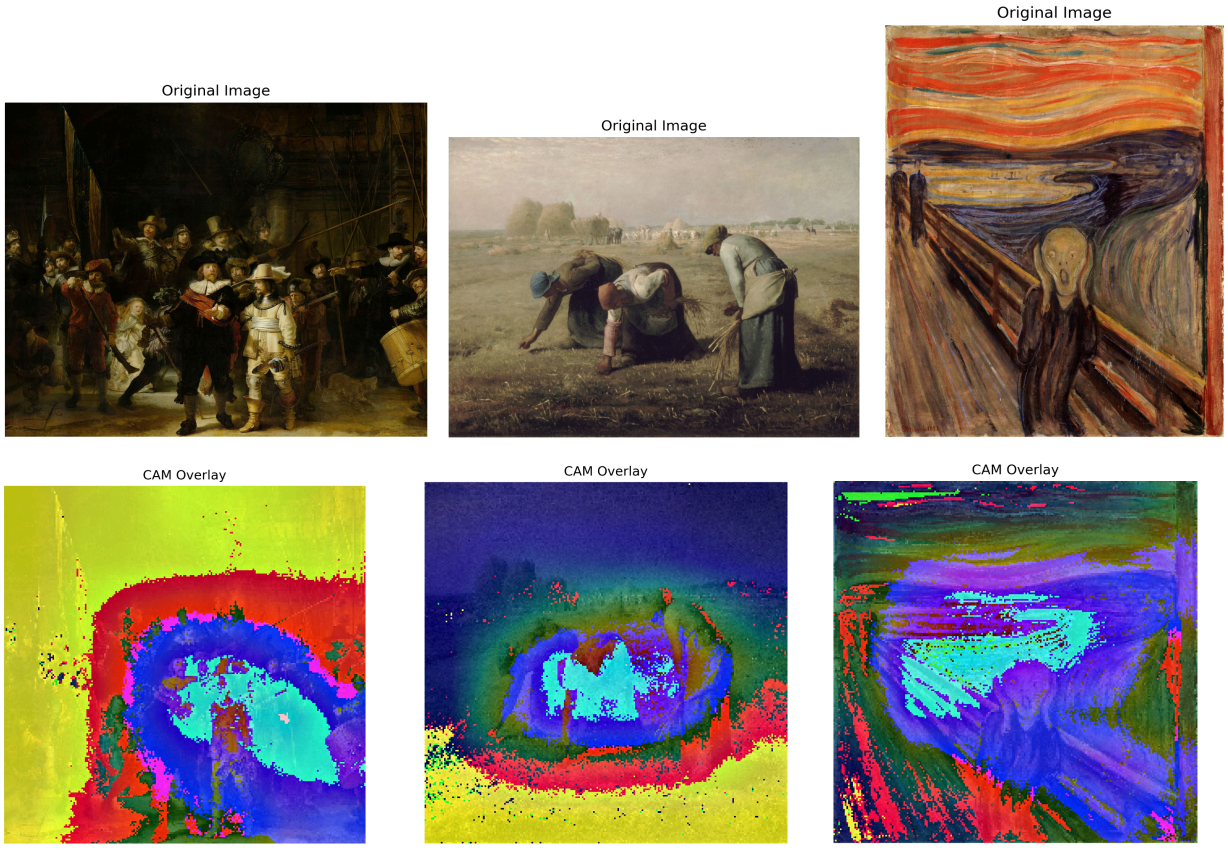

Figure 1: From left to right – Baroque (The Night Watch), Contemporary Realism (The Gleaners), and Expressionism (The Scream) with corresponding CAM overlays.

Visualizing the CAM overlays on three representative paintings— *The Night Watch* (Baroque), *The Gleaners* (Contemporary Realism), and *The Scream* (Expressionism)— reveals the features prioritized by ResNet50 and the corresponding stylistic significance. Red, yellow, and cyan indicate areas of heightened importance (in respective order). Meanwhile, blue areas denote regions ignored by our model. In the Baroque painting, the overlay focused on areas with strong contrast and dramatic lighting, mainly where chiaroscuro techniques are prominent. These features align with Baroque's emphasis on dynamic compositions and visual drama. In the Gleaners, CAM highlights fine details and crisp edges, such as the figure's clothing and precision of texture. We also notice that the feature extractor paid close attention to the grass, which has almost invisible brushstrokes. Finally, ResNet50 extracted features of the subjects of the painting, as evidenced by the cyan engulfing them. These attributes reflect Contemporary Realism's pursuit of photorealism and detailed brushwork while placing subjects in realistic settings. The CAM overlay of The Scream concentrates on broader, less structured regions. In particular, the swirling sky in the background and the loose brushwork of the water capture Expressionist painters' use of bold colours, abstract forms, and emotional intensity.

By visualizing these CAM overlays, we better understand the stylistic features prioritized by our feature extractor and how they relate to the distinct characteristics of each style. However, it is essential to note that this analysis was conducted using only one painting from each style. Oftentimes, these art styles exhibit overlapping visual features. Consequently, it will be the role of our classifiers to resolve such ambiguities during the classification process.

3. Models Used

We tested five machine learning models for this task, selected based on their suitability for high-dimensional data and capacity to handle imbalanced data:

K-Nearest Neighbors (KNN): A non-parametric algorithm that assigns labels based on the majority vote of nearest neighbors. KNN leverages similarities between feature embedding, which is ideal for PCA-reduced spaces. Furthermore, KNN's capacity for drawing non-linear decision boundaries in high-dimensional data serves well for classifying art, where decision boundaries are not linear.

Support Vector Machine (SVM): A linear classifier that maximizes the margin between classes using a hyperplane. It was chosen due to the fact that SVM effectively reduced feature spaces, and its "balanced" parameter allows it to undersample majority classes, which may improve predictions for the Contemporary Realism class, our minority class. Finally, the linear kernel enables an SVM to train faster.

Random Forest: An ensemble of decision trees trained on bootstrapped subsets of data and features. Random Forest handles noisy and imbalanced data well while capturing feature interactions.

AdaBoost: An ensemble method that iteratively boosts weak classifiers by focusing on misclassified samples. Its ability to emphasize hard-to-classify examples helps address class imbalances. By focusing on its mistakes and assigning gross misclassifications high priority, AdaBoost should be capable of overcoming the imbalance in our dataset.

Naive Bayes: A probabilistic classifier that assumes conditional independence between features. It serves as a baseline due to its computational efficiency and scalability.

3.1 RandomizedSearchCV:

We used RandomizedSearchCV with 5-fold cross-validation to optimize the models, evaluating ten random combinations of hyperparameters for each model. Unlike the more exhaustive grid search, which evaluates all possible hyperparameter combinations, RandomizedSearchCV was chosen for its efficiency in exploring larger hyperparameter spaces by randomly sampling a fixed number of hyperparameters. Below is a table of the best hyperparameters found for each model.

Model	**HyperParameters**
Random Forest	n_estimators=300, max_depth=30, min_samples_split=2, max_features="sqrt"
KNN	n_neighbors=25, weights="distance", metric="minkowski"
AdaBoost	n_estimators=200, learning_rate=0.1, estimator=DecisionTree(max_depth=3)

Table 1: Classifier Models and Hyperparameter Configurations

4. Results and Discussion

We evaluated the models using 10-fold cross-validation, running each iteration three times to ensure consistency of results. Performance was measured in terms of and measured training/testing times, accuracy, and ROC-AUC scores.

4.1 Principal Component Analysis

Firstly, we applied Principal Component Analysis (PCA) to reduce the dimensionality of our extracted feature space, which initially contained over 10,000 features. To retain 95% of the total variability in the data, we identified 503 principal components as the optimal number. Additionally, we plotted the first two principal components to visualize and better understand the structure of our reduced feature space.

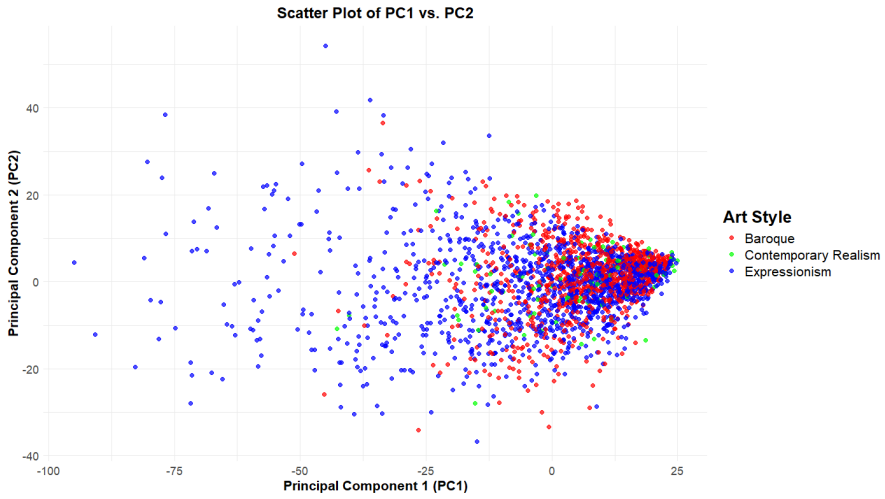

Figure 2: Scatter Plot of PC1 vs. PC2, categorized by art style

The first two principal components of our PCA reduced feature space are the most significant ones, capturing the most variability in the visual features. In Figure 2, Expressionism is far more spread out than Baroque and Contemporary Realism. As we know, Expressionist paintings are more known for loose brushwork, with muted tones, compared to the rigidness and emphasis on the contrast of lighting that characterizes Baroque and Contemporary Realism. Furthermore, Contemporary Realism and Baroque are closely clustered as both styles emphasize realistic depictions, fine detail, and precise brushstrokes, making their visual features harder to distinguish.

4.2 Model Training Times

The training times for each model are compared using a log-transformed boxplot due to the wide variability. KNN scored the lowest, as expected, since the model has no training phase. It simply stores the training data. Meanwhile, Naive Bayes' training time is short due to only needing to compute metrics such as mean and variance for each feature per class. In contrast, Random Forests take substantially more time because they build multiple decision trees through bootstrapping and random feature selection, involving frequent computations on data subsets. SVMs involve maximizing the margin between classes,

utilizing a one-class versus all approach, leading us to create three classifiers instead each time our model is trained. Finally, AdaBoost takes the longest training time as it is an iterative algorithm that trains weak classifiers sequentially. However, the training times of Support Vector Machines, AdaBoost, and Random Forests vary very little. Adaboost and Random Forest train the same number of weak classifiers every time, ensuring consistent training times. At the same time, SVMs with a linear kernel are deterministic, ensuring the optimization process converges to the same solution every time, explaining the lack of variety.

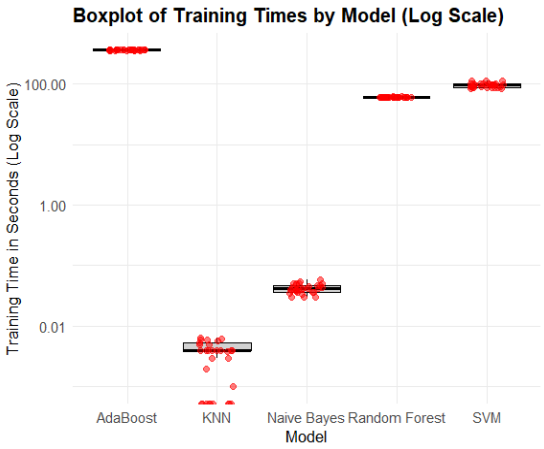

Figure 3: Training Times by Model (log scale)

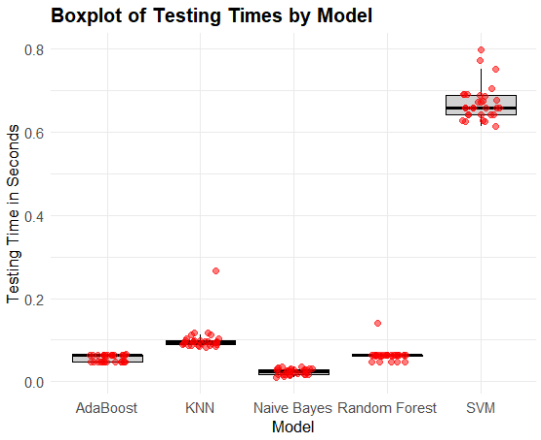

Figure 4: Test times by Model

4.3 Model Testing Times

Testing times provide insight into each model's efficiency when we need real-time predictions. All models other than SVM had considerably fast testing times. This can be explained by Support Vector Machines needing to run three classifiers every time and then calculate the dot product of test points and the support vectors, thus slowing down test time.

4.4 Model Accuracy

As shown in Table 2, Naive Bayes stands out as the worst performer, likely due to its feature independence assumption, which does not hold for paintings. In contrast, KNN achieved the highest accuracy, followed by Adaboost and Random Forest. All models exhibited low standard deviation values, indicating consistent performances across the cross-validation folds with three repetitions.

Model	Mean Accuracy	Standard Deviation
AdaBoost	0.715	0.0128
KNN	0.733	0.0119
Naive Bayes	0.501	0.0120
Random Forest	0.706	0.0107
SVM	0.692	0.0107

Table 2: Model Performance metrics

Model	Mean Area Under Curve
SVM	0.76
Random Forest	0.76
KNN	0.77
Naive Bayes	0.61
Adaboost	0.65

Table 3: Model Performance Based on Mean AUC Scores

4.5 Area Under the Curve (AUC)

Although each model's mean accuracy offers us a general insight into their performance, the mean area under the curve values provides a deeper understanding of the models' performances, particularly when handling imbalanced datasets. While accuracy measures the proportion of correct predictions, the AUC evaluates a model's true positive and false positive rates, providing a measure of the model's ranking ability rather than relying on a single threshold. Furthermore, we should note that SVM and K-nearest neighbors do not give probabilistic outputs by default. However, we can approximate these, as KNN can produce pseudo-probabilities by calculating the number of neighbors in the class divided by the total number of neighbors (25), which we will use as an approximation. Meanwhile, SVMs, by design, focus on finding hyperplanes that maximize the margin between classes. As a workaround, we utilize logistic regression on the decision function scores to approximate probabilities (although it should be noted that these probabilities are not directly computed from the SVM's optimization.

Table 3 demonstrates similar results to Table 2, with KNN being the highest AUC value, confirming that it consistently outperforms the other models albeit by a very small margin, while SVM and Random Forest achieve similar AUC means. Yet, Adaboost and Naive Bayes stand out with very low scores, signaling that these models may not be suitable for art classification.

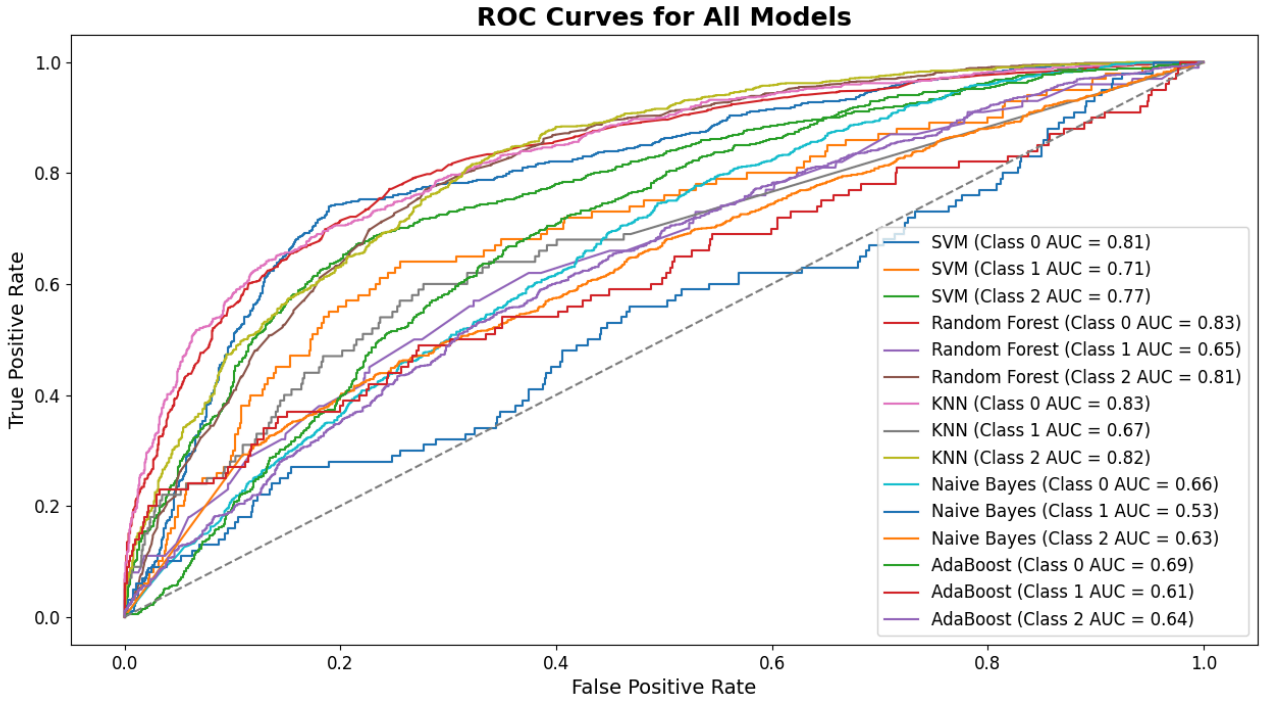

Figure 5: ROC Curves and AUC Scores for each Model

When we look at each model's ROC curve by class, we notice that every model performs better than random guessing, as an area under the curve higher than 0.5 states that the model performs better at distinguishing between positive and negative cases. At first glance, our SVM classifier truly excelled at predicting our less frequent class (Class 1 = Contemporary Realism). Yet, it seems to be outshined by KNN, which was way more adept at predicting the majority classes. Finally, Adaboost was disappointed,

as it could not predict our minority class, as we expected, suggesting that our minority class samples may not have received sufficient weight during the boosting process.
Our pairwise comparisons table further validates these observations, which provides insights into the statistical significance of differences between model performances. If a model is statistically significant (p-value <0.05), it signifies that this model is performing significantly better, indicating that it is a much more suitable choice. According to our table, there is statistically no significant difference between SVM, KNN, and Random Forest models, yet they all seem to significantly outperform AdaBoost and Naive Bayes, making them the most suitable choices in terms of accuracy

Model 1	**Model 2**	**p-value**	**Significant?**
SVM	Random Forest	1.0000	No
SVM	KNN	0.7418	No
SVM	Naive Bayes	0.0058	Yes
SVM	AdaBoost	0.0056	Yes
Random Forest	KNN	0.2254	No
Random Forest	Naive Bayes	0.0137	Yes
Random Forest	AdaBoost	0.0972	No
KNN	Naive Bayes	0.0075	Yes
KNN	Adaboost	0.0695	No
Naive Bayes	AdaBoost	0.1946	No

Table 4: Pairwise Comparisons of Model Performances with Statistical Significance (p-values)

4.6 Confusion Matrices

The confusion matrices provide further insights into how well each model performs across our classes and highlight key areas of strength and weaknesses for each model. Firstly, the SVM performs well overall, particularly in predicting Baroque and Expressionism (693 for Baroque, 849 for Expressionism). However, it struggles with Contemporary Realism, misclassifying many of its samples in Expressionism. This aligns with the model's low AUC score. Adaboost, however, performs poorly, as it seems to have focused its efforts on classifying Expressionism correctly while severely underperforming in the two other classes. Random Forest performs slightly better than Adaboost but still suffers from similar problems. Meanwhile, KNN seems to be a top performer, garnering solid results across all classes, thus confirming its assumption of similar paintings being often clustered together. Finally, Naive Bayes performs poorly, particularly for expressionism (476 correct predictions) and contemporary realism (11 correct predictions), proving that paintings' visual features are not independent.

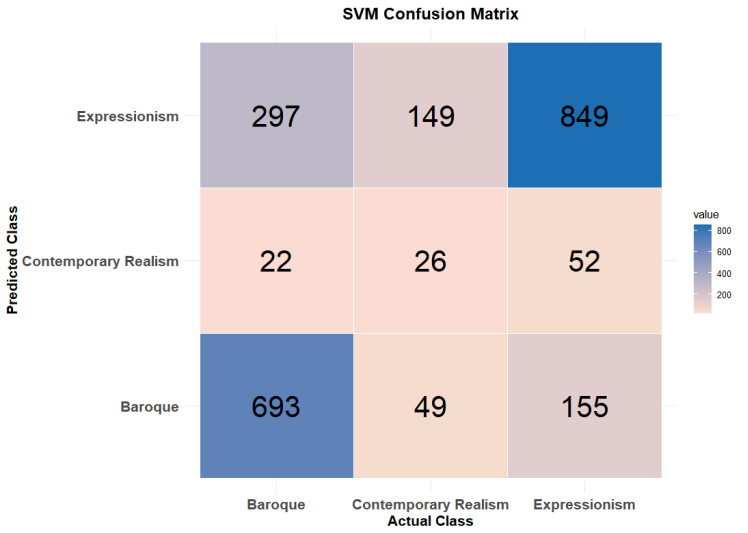

Figure 6: Confusion Matrix of SVM Classifier

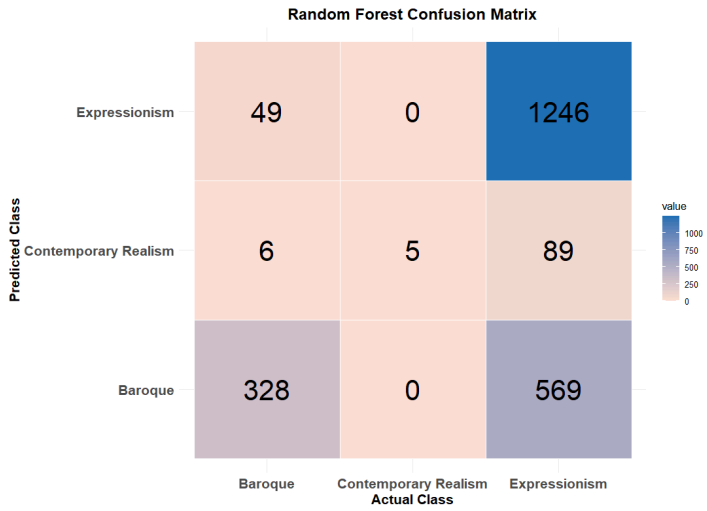

Figure 7: Confusion Matrix of Random Forest Classifier

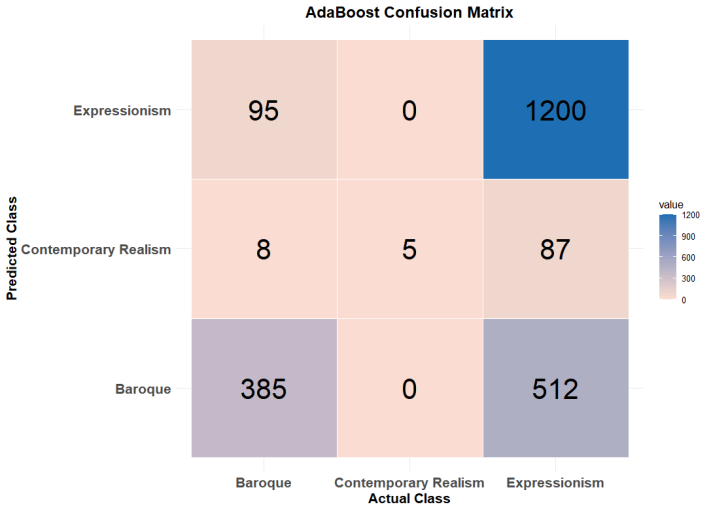

Figure 8: Confusion Matrix of AdaBoost Classifier

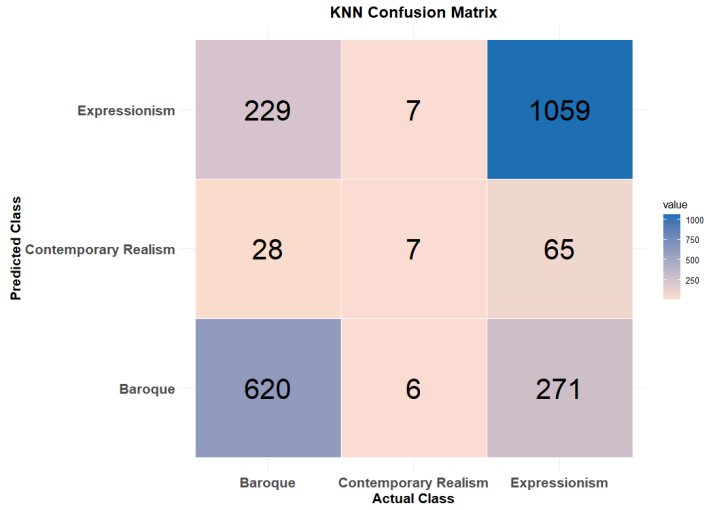

Figure 9: Confusion Matrix of KNN Classifier

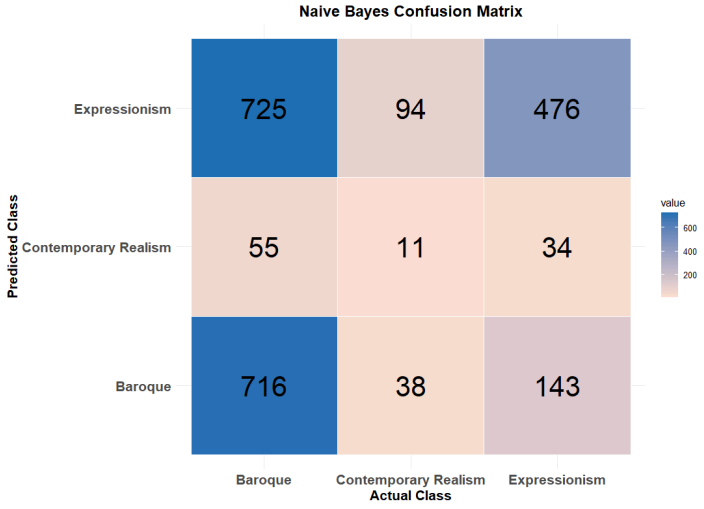

Figure 9: Confusion Matrix of Naive Bayes Classifier

5. Final Model Selection

At this point, we know that SVM, KNN, and Random Forest are not significantly different in accuracy to warrant a selection between one of the three. Thus, we include testing and training times and their respective f1-score for each class. With the f1-score, we can calculate accuracy while taking into account false positives and negatives through its inclusion of precision and recall. In doing so, the f1-score allows an unskewed measure of model performance when evaluating a specific class. In addition, training and testing times are incredibly important since our goal was to find a computationally inexpensive and accurate workflow when predicting a small range of classes. Thus, we decided to rank our KNN, SVM, and Random Forest models based on these 5 criteria and find their average rank across these categories.

Model	Training Time Rank	Testing TIme Rank	Baroque F1- score Rank	Contemporary Realism F1- score Rank	Expressionism F1-score rank	Average Rank
KNN	1	2	2	2	1	1.6
SVM	3	3	1	1	3	2.2
Random Forest	2	1	3	3	2	2.2

Table 5: Model Performance Rankings across Training Time, Testing Time, and F1-Scores

According to our table, KNN is the best model, as it is incredibly fast and accurate when predicting classes. Its only weakness is the contemporary realism f1-score, where KNN struggled compared to SVM (see AUC graph). Meanwhile, SVM and Random Forest seem polar opposites, as the former is far more accurate, but the latter is faster for training and testing times. In the case of predicting minority classes and having the time necessary to wait for predictions, SVM is a great tool.

6. Conclusion

Our approach to classifying three art styles, where one was severely underrepresented, utilized ResNet50 for feature extraction and PCA for dimensionality reduction. Class activation maps confirmed that ResNet50 focused on meaningful regions that displayed distinguishing features of each class. Furthermore, we ran RandomizedGridCV to determine an optimal set of hyperparameters for KNN, Random Forest, and Adaboost. Among the 5 models tested with 10-fold cross-validation times repeated three times, K-Nearest Neighbors emerged as the most suitable due to its speed and balanced accuracy. However, SVM excelled in minority class prediction. These findings demonstrate the effectiveness of narrowing classification tasks to fewer styles while optimizing computational resources. Future work could expand the framework to include additional styles, deep learning models, comprehensive hyperparameter tuning, or make models handle higher dimensional data to classify art styles.