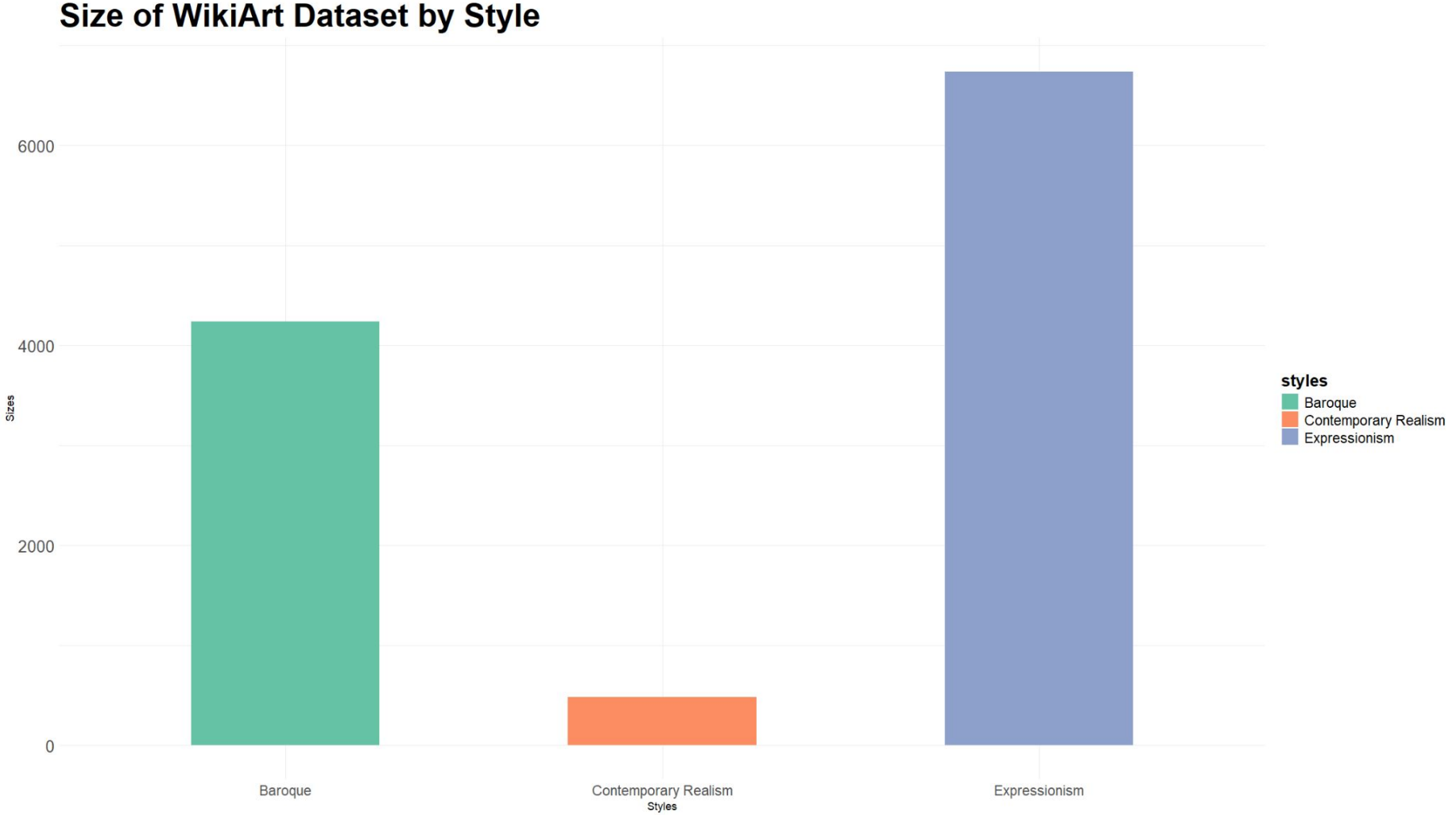
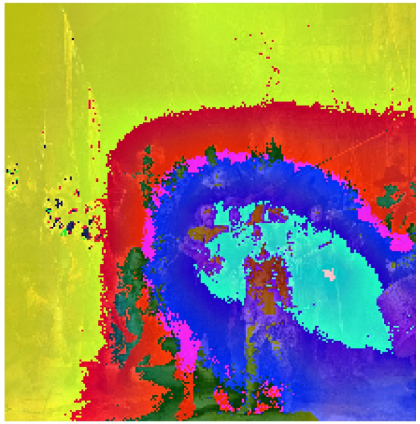
Image Classification for Artistic Style

John Bute, Kelly Gao, Anthony Le.


Size of WikiArt Dataset by Style
6000
4000
2000
0
Sizes
Baroque
Contemporary Realism
Expressionism
Styles
styles
Baroque
Contemporary Realism
Expressionism

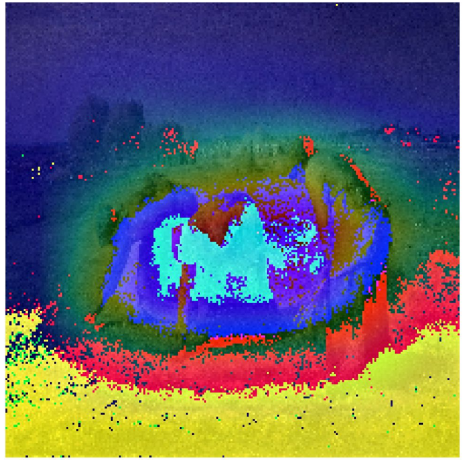
Original Image

CAM Overlay



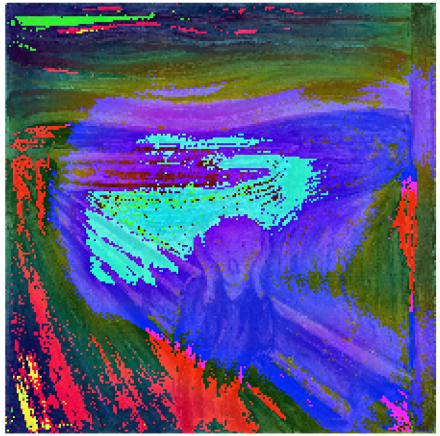
Original Image

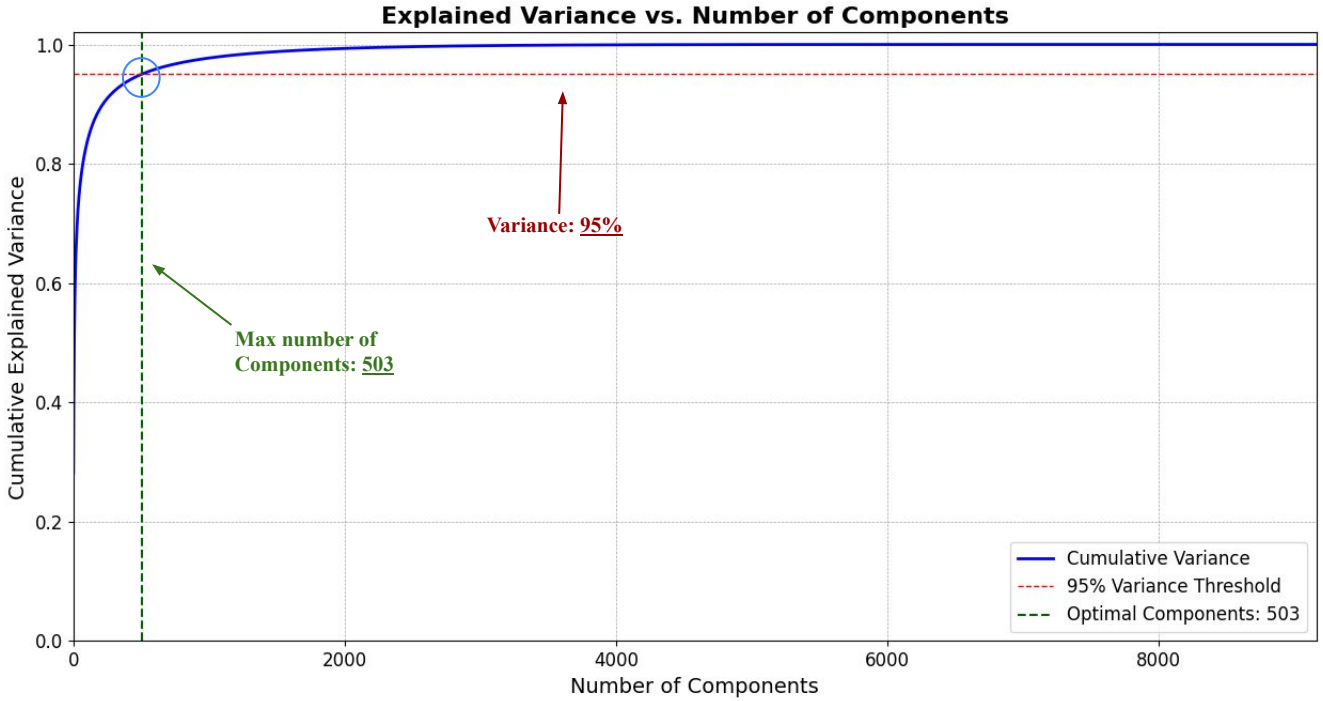
CAM Overlay



Original Image

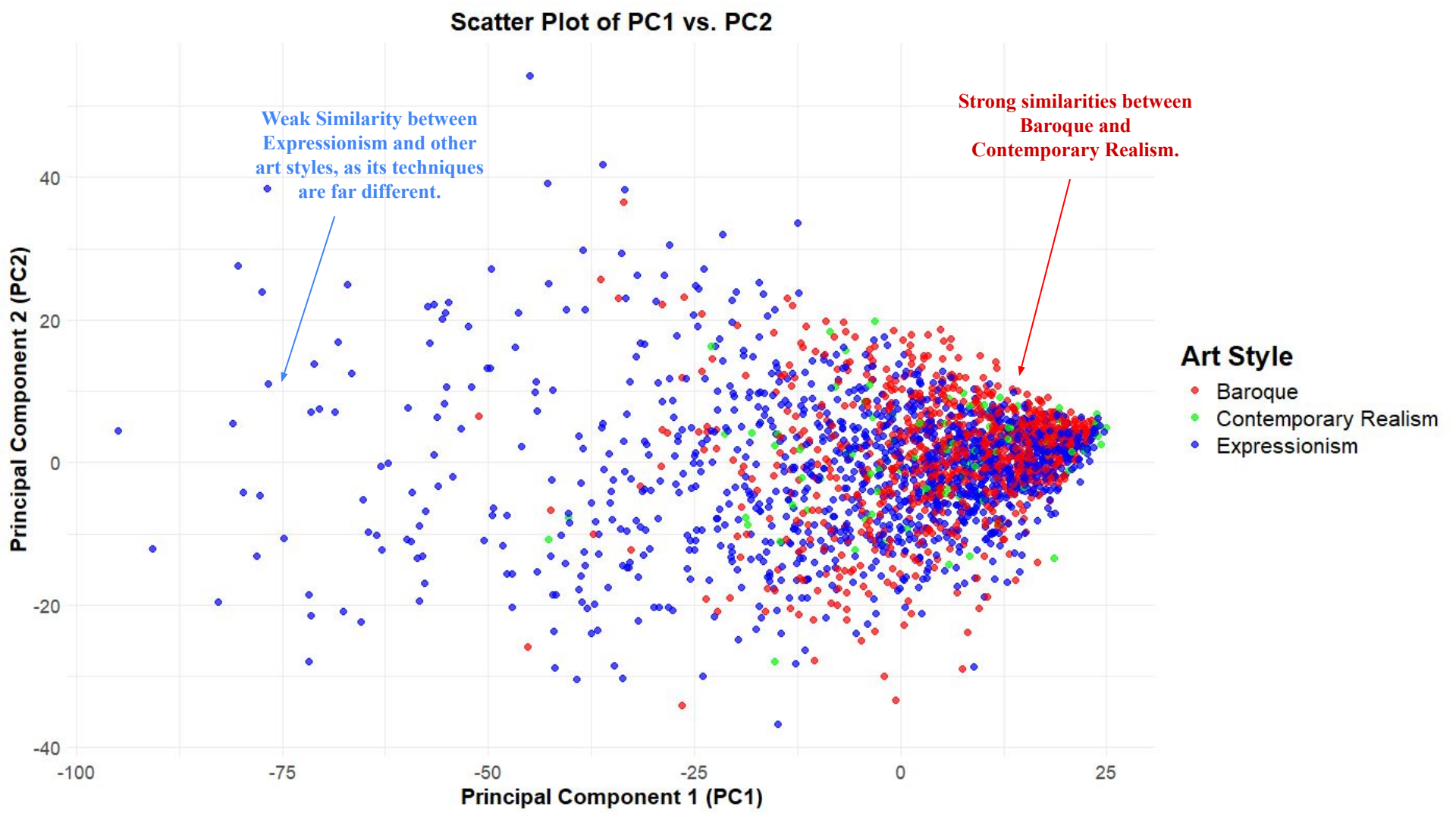
CAM Overlay

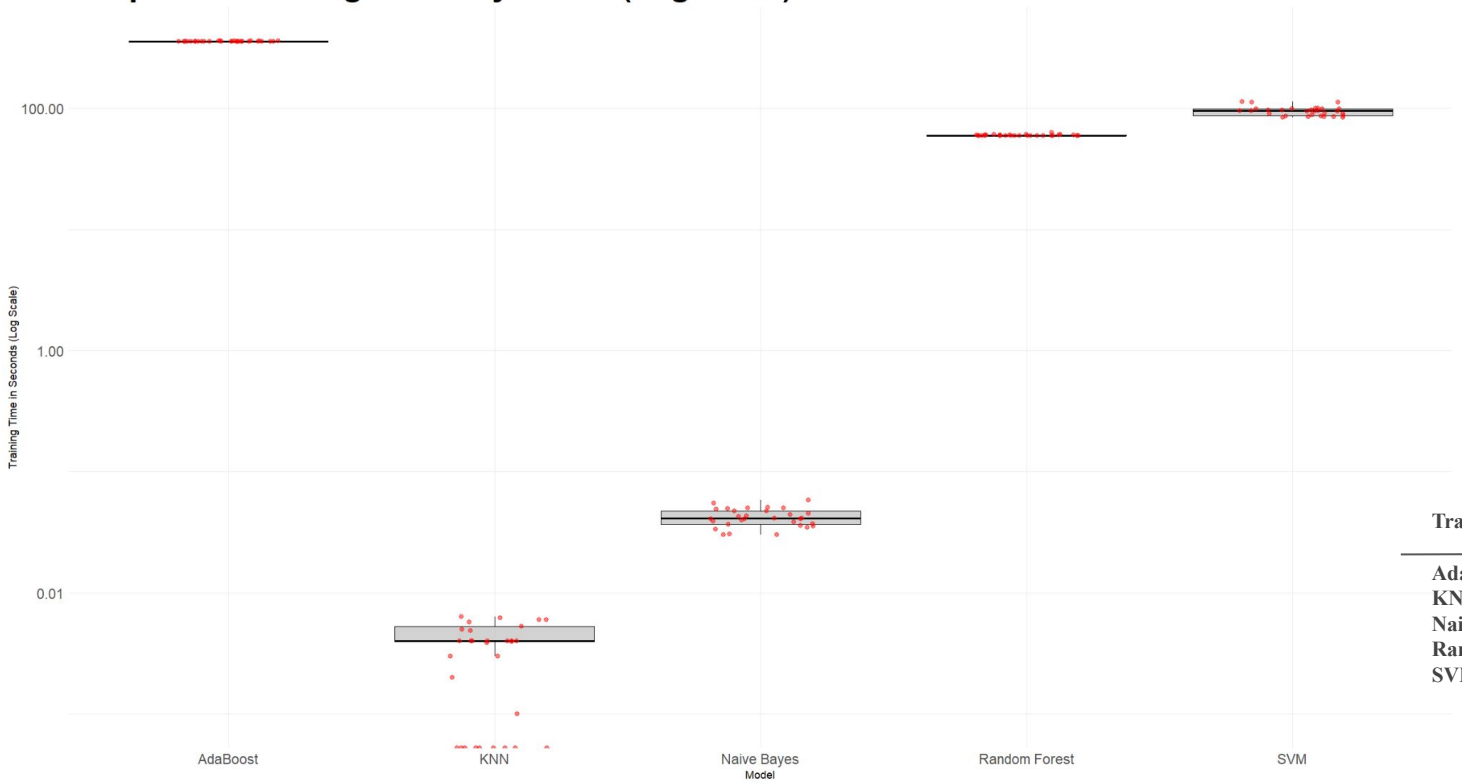




Following a Principal Component Analysis (PCA), we compared the cumulative explained variance against the number of components to determine how many components we need to still preserve a 95% threshold. Determining this allows us to reduce the dimensionality of the dataset and improve computational efficiency while not losing a significant amount of information.

Observing the graph, we can see that to retain the 95% variance we would have to keep 503 components.

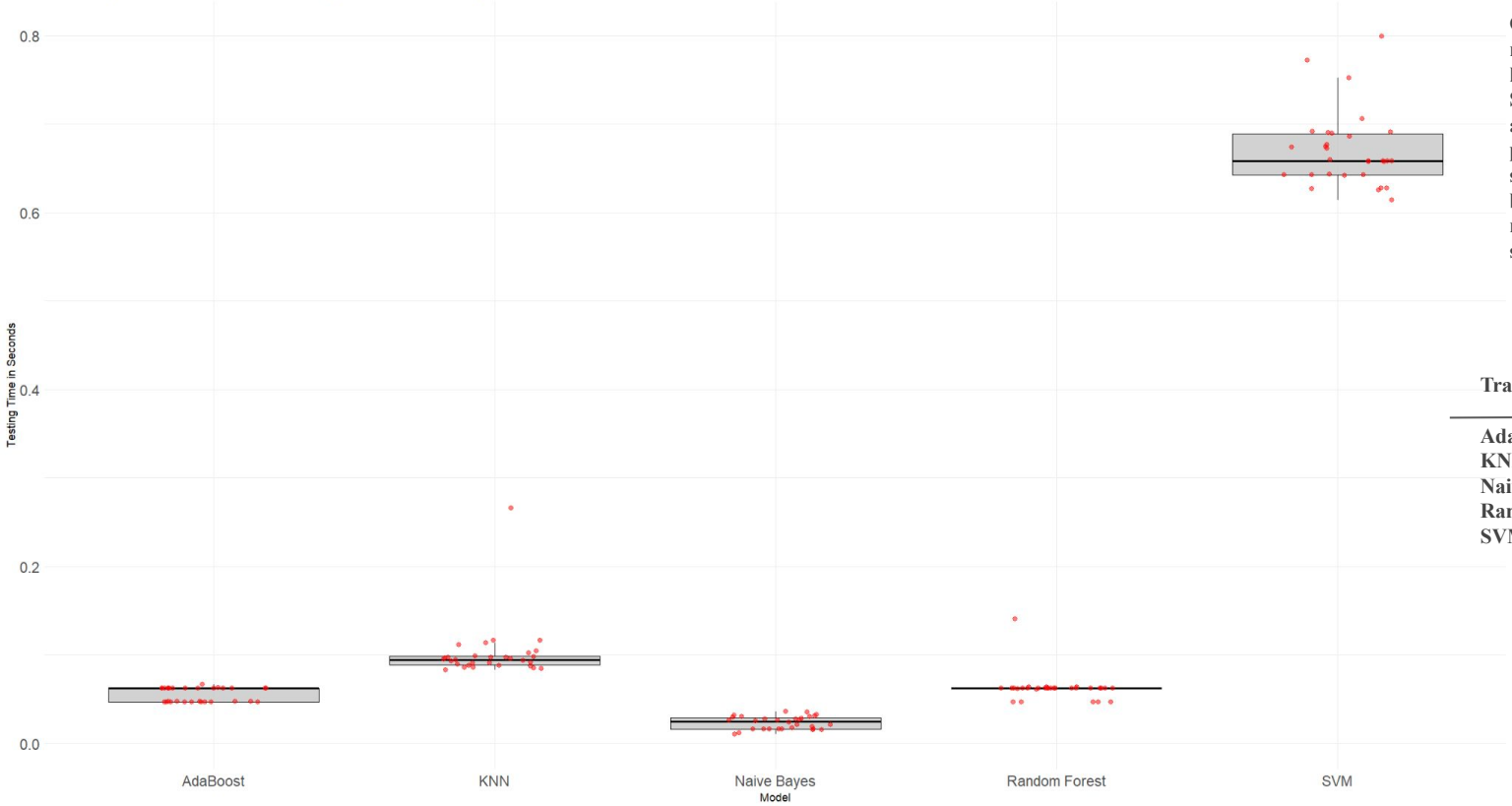

Scatter Plot of PC1 vs. PC2
Weak Similarity between Expressionism and other art styles, as its techniques are far different.
Strong similarities between Baroque and Contemporary Realism.
Art Style
Baroque
Contemporary Realism
Expressionism
Principal Component 2 (PC2)
40
20
0
-20
-40
-100
-75
-50
-25
0
25
Principal Component 1 (PC1)



Observing the training times of each model, AdaBoost, Random Forest and SVM were projected to have the highest average runtimes, with AdaBoost being the model with the highest average runtime. Conversely, Naive Bayes and SVM were the only models that were projected with the lowest average runtime. This indicates that compared to the other models, Naive Bayes and SVM are the weakest performers for training overall.

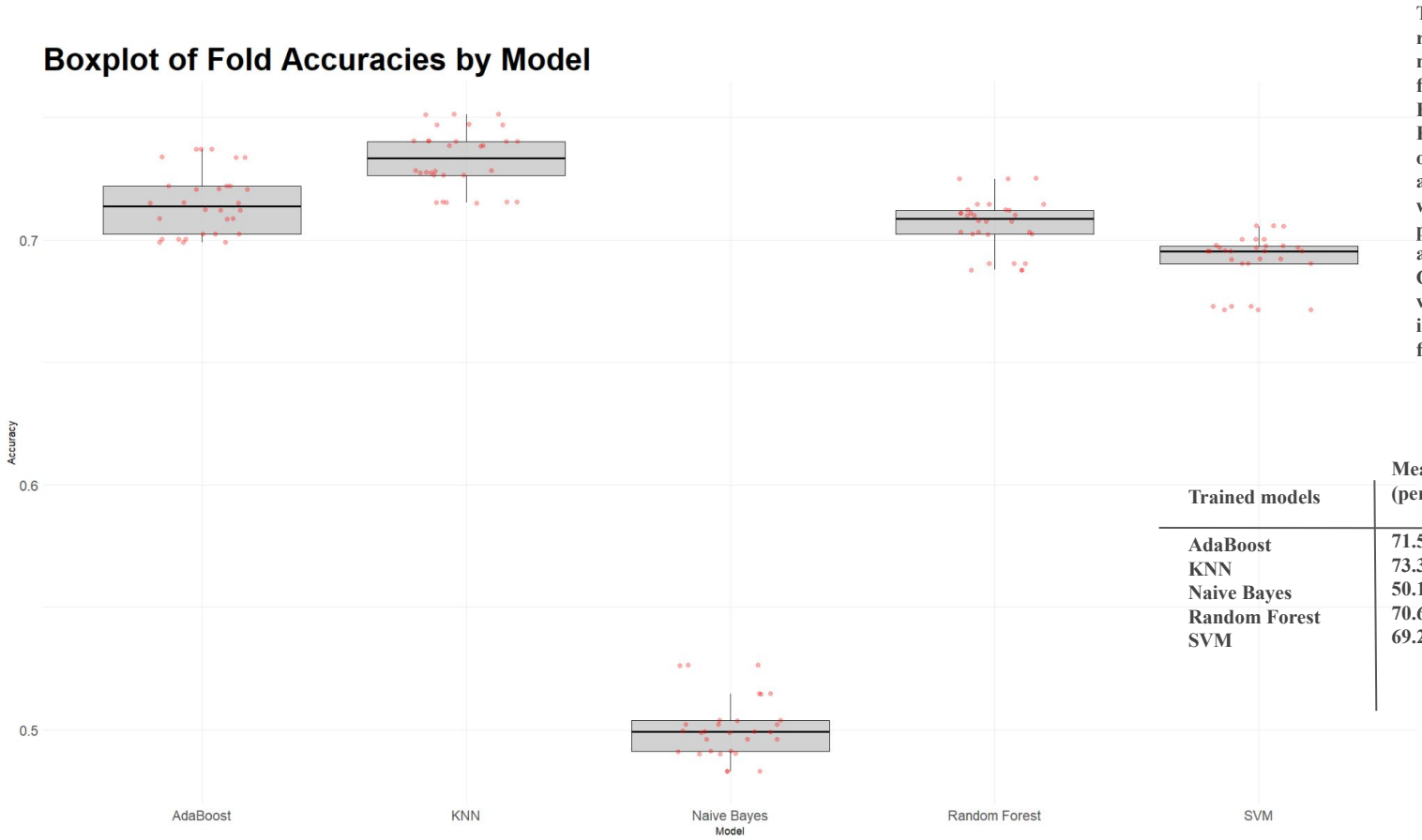
Trained models	Unlogged average runtimes (seconds)
AdaBoost	359.00
KNN	94.80
Naive Bayes	0.0420
Random Forest	60.0
SVM	0.0301

Boxplot of Testing Times by Model



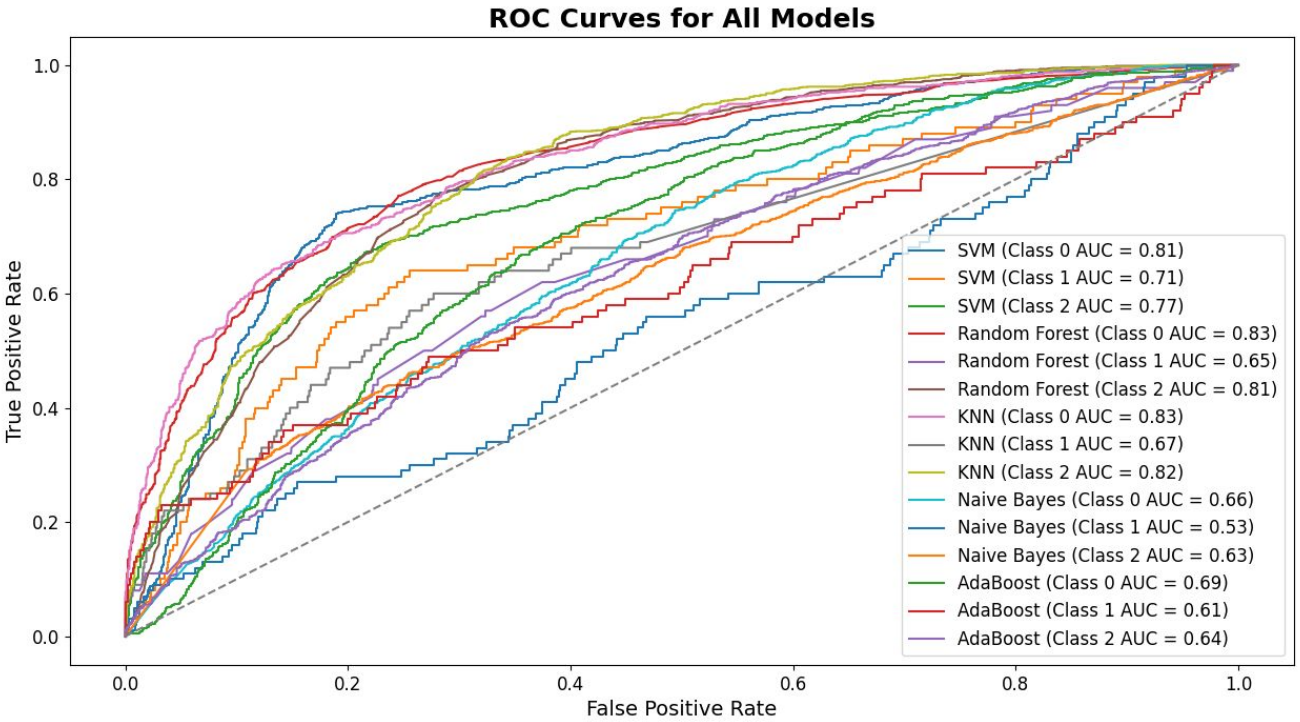
Observing the testing times for each model, we observed all models to have a relatively low runtime, with SVM being the only model seen with an obscenely high runtime. This posits an idea of a trade-off with some of models that performed better in their training times; certain models will be faster at training, but slow at testing and vice versa.

Trained models	average runtimes (seconds)
AdaBoost	0.0554
KNN	0.101
Naive Bayes	0.0234
Random Forest	0.0626
SVM	0.671



The fold accuracies of each model represent the consistency of each model across the cross validation folds with three separate repetitions. From the cross validation, AdaBoost, KNN, Random Forest and SVM are observed to achieve a high mean accuracy. Conversely, Naive Bayes was observed to be the worst performer, likely due to its assumption of feature independence. Overall, each model achieved a low variance in accuracy, nonetheless indicating a consistent performance from all models.

Trained models	Mean Accuracy (percentage)	Variance in Accuracy (percentage)
AdaBoost	71.5	0.0165
KNN	73.3	0.0143
Naive Bayes	50.1	0.0144
Random Forest	70.6	0.0114
SVM	69.2	0.0116



This graph observes the mean performance of each model, specifically when handling imbalanced datasets. Observing the graph itself, we can see that KNN outputs the highest AUC value even by class, validating its consistent outperformance against other models, though at a small margin.

Additionally, we can observe that SVM and Random Forest achieves a similar performance, with AdaBoost and Naive Bayes being the lowest performing models of the group. These low values from AdaBoost and Naive Bayes indicate that these models may not be suitable for art classification.

Models	Mean Area Under Curve
AdaBoost	0.65
KNN	0.77
Naive Bayes	0.61
Random Forest	0.76
SVM	0.76

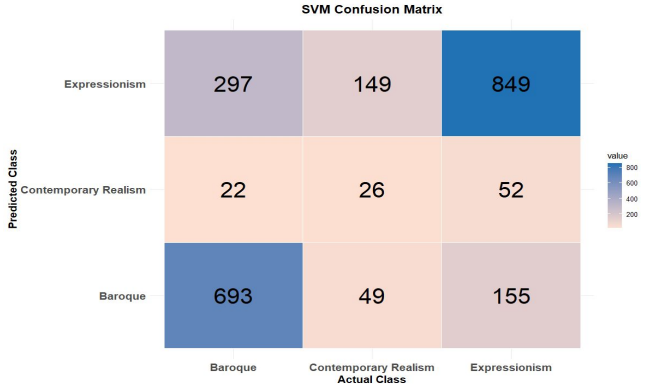
Pairwise Comparisons of Model Performances With Statistical Significance

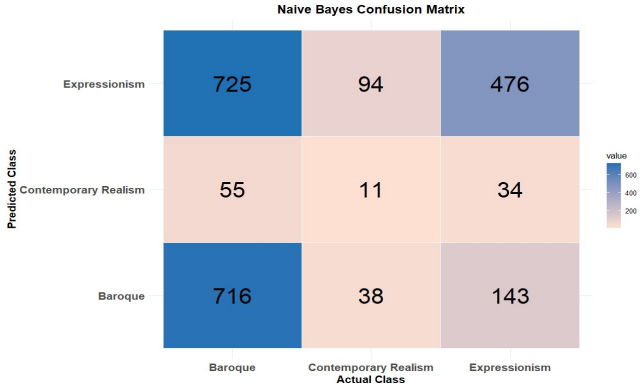
Model 1	Model 2	P-Value	Significant?
SVM	Random Forest	1.00	No
SVM	KNN	0.7418	No
SVM	Naive Bayes	0.0058	Yes
SVM	AdaBoost	0.0056	Yes
Random Forest	KNN	0.2254	No
Random Forest	Naive Bayes	0.0137	Yes
Random Forest	AdaBoost	0.0972	No
KNN	Naive Bayes	0.0075	Yes
KNN	AdaBoost	0.0695	No
Naive Bayes	AdaBoost	0.1946	No

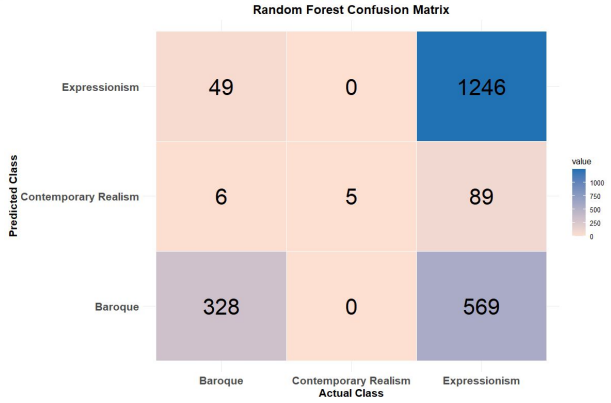
This table which shows the individual differences between the performance of each model further supports our findings on each of the model's performances seen in the previous graph.

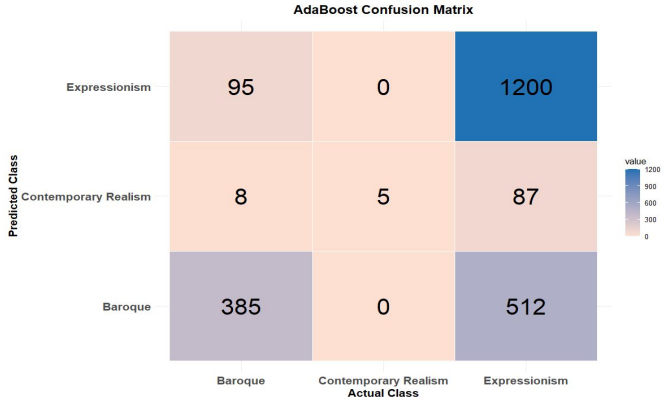
The p-value observed within the table shows the performance similarity between the models, where if a model is statistically significant ($p < 0.05$), then one of the models is outperforming the other.

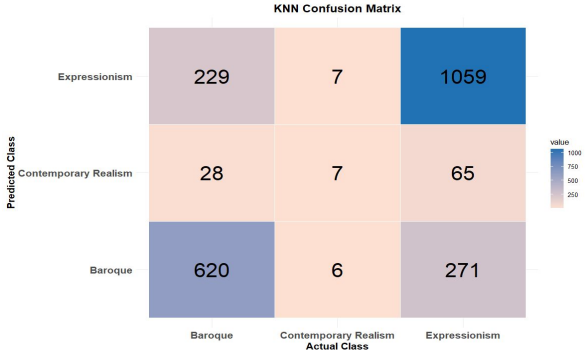
According to our table, we are not able to view a significant difference in performance between SVM, Random Forest and KNN and that all three models are able to outperform AdaBoost and Naive Bayes. This confirms that SVM, KNN and Random Forest are the most suitable choices for this task in terms of accuracy.


SVM Confusion Matrix
Expressionism
Contemporary Realism
Baroque
297
149
849
22
26
52
693
49
155
Predicted Class
Baroque
Contemporary Realism
Expressionism
Actual Class
value
800
600
400
200


Naive Bayes Confusion Matrix
Expressionism
Contemporary Realism
Baroque
725
94
476
55
11
34
716
38
143
Predicted Class
Baroque
Contemporary Realism
Expressionism
Actual Class
value
600
400
200


Random Forest Confusion Matrix
Expressionism
Contemporary Realism
Baroque
49
0
1246
6
5
89
328
0
569
Predicted Class
Baroque
Contemporary Realism
Expressionism
Actual Class
value
1000
750
500
250
0


AdaBoost Confusion Matrix
Expressionism
Contemporary Realism
Baroque
95
0
1200
8
5
87
385
0
512
Predicted Class
Baroque
Contemporary Realism
Expressionism
Actual Class
value
1200
900
600
300
0


KNN Confusion Matrix
Expressionism
Contemporary Realism
Baroque
229
7
1059
28
7
65
620
6
271
Predicted Class
Baroque
Contemporary Realism
Expressionism
Actual Class
value
1000
750
500
250

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

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

Questions? Comments?