Classifying Art Styles!

DATASCI 281

What did we classify?

Link Here: https://www.kaggle.com/datasets/steubk/wikiart?resource=download&select=classes.csv

Dataset: 80k unique images, ranging across 27 different art genres (*Cubism, Impressionism, Pop Art, etc*)

Areas of interest: We hope to explore what visual information actually drives art-style classification and if we can potentially find it through only looking at an images' edges/lines/structure or color and texture alone.

Goal: Accurately label within correct art genre

EDA

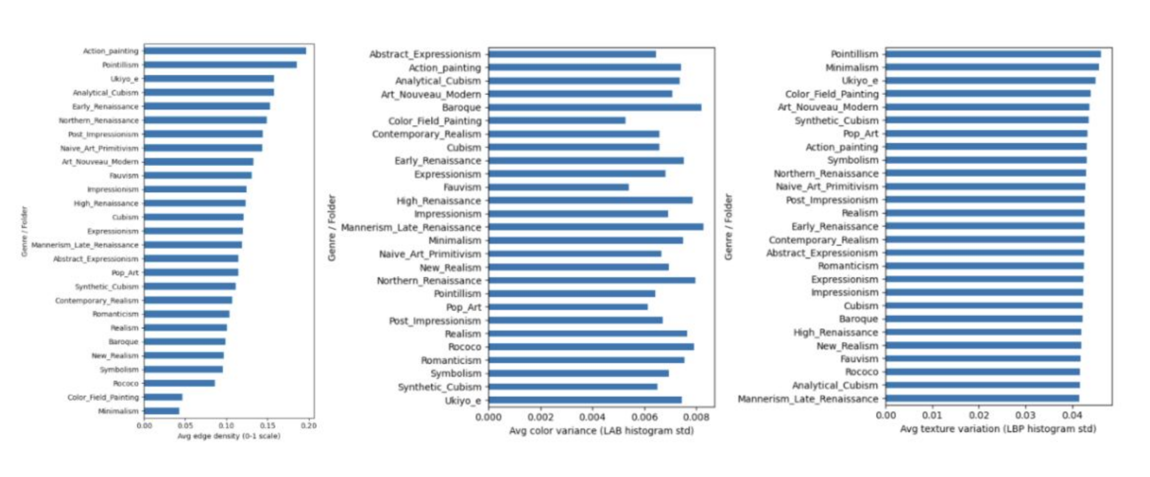
The initial Kaggle dataset was incredibly large, we first needed to make the dataset smaller. To ensure interpretability and computational efficiency of our investigation, we plan to restrict the dataset down to six art genres. Our data filtering process consisted of the following considerations:

- **Computation Efficiency**: We remove genres with fewer than 1k images as small sample sizes can introduce instability/noise. By focusing on balanced and sufficiently large classes, we can train faster and produce consistent results.

EDA (Cont)

Evaluation and Interpretability: Our goal generally is to understand whether classical structure, edges, color, or texture, drives stylistic differences across genres. We choose to limit the number of genres to those that on average express a higher magnitude to those categories.

Selected Genres: Ukiyo-e and Early Renaissance. Pop Art and Baroque. Rococo and Mannerism Late Renaissance.



Feature Extraction (Edges/ Edge Direction)

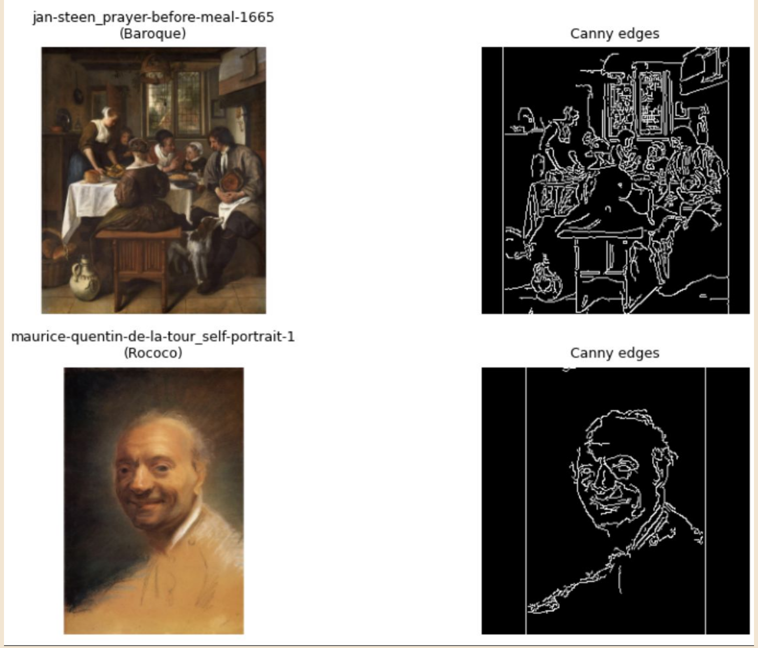
	filename	image_id	genre	orig_path	std_shape_h	std_shape_w	std_channels	edge_density	edge_orient_bin0	edge_orient_bin1
0	maerten-van-heemskerck_self-portrait.jpg	maerten-van-heemskerck_self-portrait	Mannerism_Late_Renaissance	/Users/austinblee/github/wikiart/Mannerism_Lat...	256	256	3	0.048721	0.475222	0.299861
1	agnolo-bronzino_portrait-of-ferdinando-de-medi...	agnolo-bronzino_portrait-of-ferdinando-de-medici	Mannerism_Late_Renaissance	/Users/austinblee/github/wikiart/Mannerism_Lat...	256	256	3	0.058380	0.518963	0.292354
2	tintoretto_autumn.jpg	tintoretto_autumn	Mannerism_Late_Renaissance	/Users/austinblee/github/wikiart/Mannerism_Lat...	256	256	3	0.117386	0.360474	0.318349
3	hans-von-aachen_venus-cupid-and-a-satyr-1598.jpg	hans-von-aachen_venus-cupid-and-a-satyr-1598	Mannerism_Late_Renaissance	/Users/austinblee/github/wikiart/Mannerism_Lat...	256	256	3	0.110001	0.467689	0.325112
4	tintoretto_portrait-of-a-white-bearded-man.jpg	tintoretto_portrait-of-a-white-bearded-man	Mannerism_Late_Renaissance	/Users/austinblee/github/wikiart/Mannerism_Lat...	256	256	3	0.029678	0.479910	0.326697

edge_orient_bin0	edge_orient_bin1	edge_orient_bin2	edge_orient_bin3	edge_orient_bin4	edge_orient_bin5	edge_orient_bin6	edge_orient_bin7
0.475222	0.299861	0.300953	0.282075	0.240887	0.221697	0.284728	0.441055
0.518963	0.292354	0.255518	0.272149	0.293597	0.221325	0.225677	0.466896
0.360474	0.318349	0.323012	0.360940	0.316173	0.199745	0.286949	0.380837
0.467689	0.325112	0.279090	0.307387	0.242707	0.181603	0.273026	0.469865
0.479910	0.326697	0.354983	0.254726	0.210727	0.207270	0.286155	0.426011

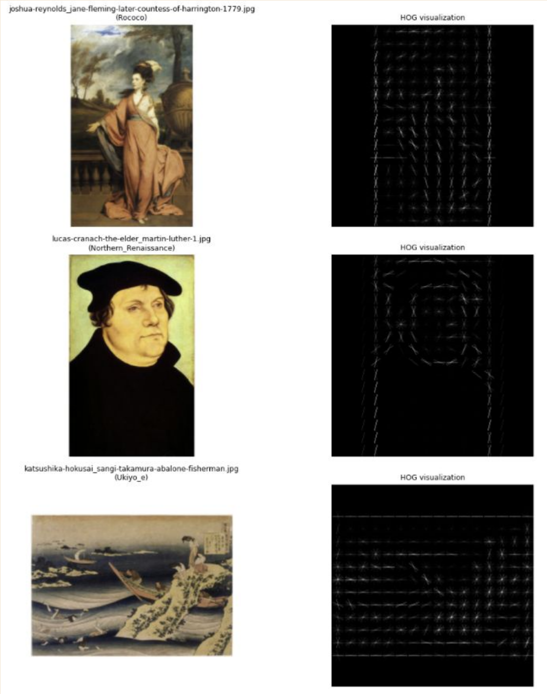
Bins 0-4 representative of leftward horizontal edges
Bin 5, representative of vertical
Bin 6-9 representative of rightward horizontal edges

We add 11 unique distinct edge features

Feature Extraction (Edges density) - cont.



Histogram of Gradients (HOG)



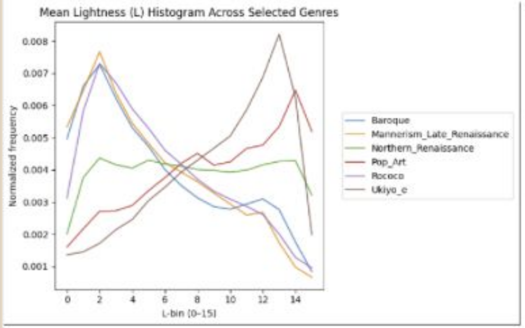
Given our configuration of:

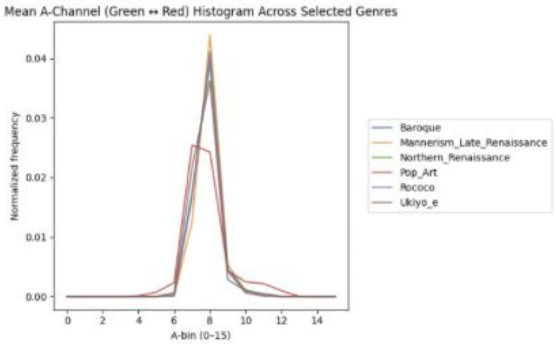
- 256x256 standardized images;
- 16x16 pixels per cell;
- 2x2 cells per block; and
- 9 orientation bins,

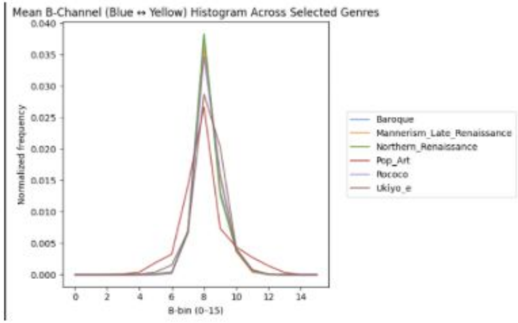
—

- Each image is divided into 16x16 grids of spatial cells. Because each block spans 2 cells and slides across the grid one cell at a time, there are 15 possible block positions along each dimension, yielding 15x15 = 225 total blocks.
- Each block contains four cells and each cell contributes a 9-bin orientation histogram, resulting in 36 features per block.
- **Across 225 blocks and 36 features per block we produce 8,100 features per image.**

Feature Extraction (color features - lab)







Feature Extraction (color features - lab) - cont

Feature Extraction (lbp)

Feature Name	Meaning / Description	What It Captures
LBP_0	Flat or uniform texture pattern	Smooth areas with little change in brightness
LBP_1	Small edge or corner pattern	Slight transitions or soft brushstrokes
LBP_2	Fine texture or micro-edge pattern	Repeated textures or dotted brushwork
LBP_3	Moderate texture variation	Areas with clear brush direction or mixed tones
LBP_4	Strong edge pattern	Sharp boundaries or structured paint lines
LBP_5	Complex light-dark mix	Varied texture patterns and expressive strokes
LBP_6	Complex light-dark mix	Varied texture patterns and expressive strokes
LBP_7	Complex light-dark mix	Varied texture patterns and expressive strokes
LBP_8	Complex light-dark mix	Varied texture patterns and expressive strokes
LBP_9	Complex light-dark mix	Varied texture patterns and expressive strokes

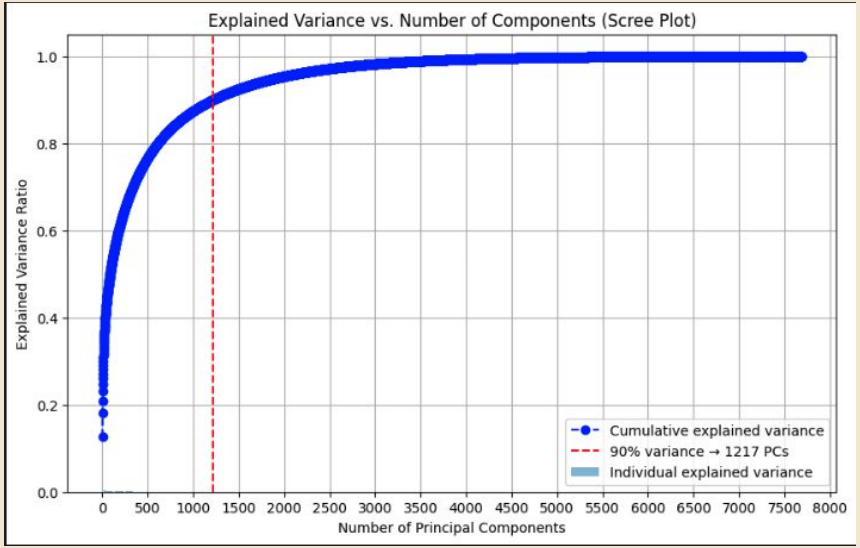
Resnet-50

- To complement our classical feature set, we leveraged Resnet-50 as our pre-trained neural network model.
- ResNet-50 consists of 50 layers organized into stacked residual blocks, each containing convolutional layers that learn increasingly abstract visual representations. ResNet50 has demonstrated great performance on various image classification benchmarks, making it an excellent choice to complement our classical feature set.
- Each image is passed through the network with the final classification layer removed, resulting in a 2,048-dimensional embedding vector.

Combined set roughly sized up to 10,191 features with resnet + feature extraction

PCA

- Training set is shaped: 7686, 10191
  - Each painting in the data set has 10191 dimensional vector of hand-crafted + resnet features. This means each painting is essentially represented by a 10k~ dim vector
  - This is extremely high dimensional space, requiring each training step update over 10k coefficients; each gradient update involves a dot product in 10k-dimensional space.
  - Convergence slower b/c multicollinearity + curse of dimensionality.
  - We found 1217 number of components that capture 90% of the variance within the feature set. This is 10x compression in the feature space. We can use these vectors on our classifying set, which reduces issues mentioned above.

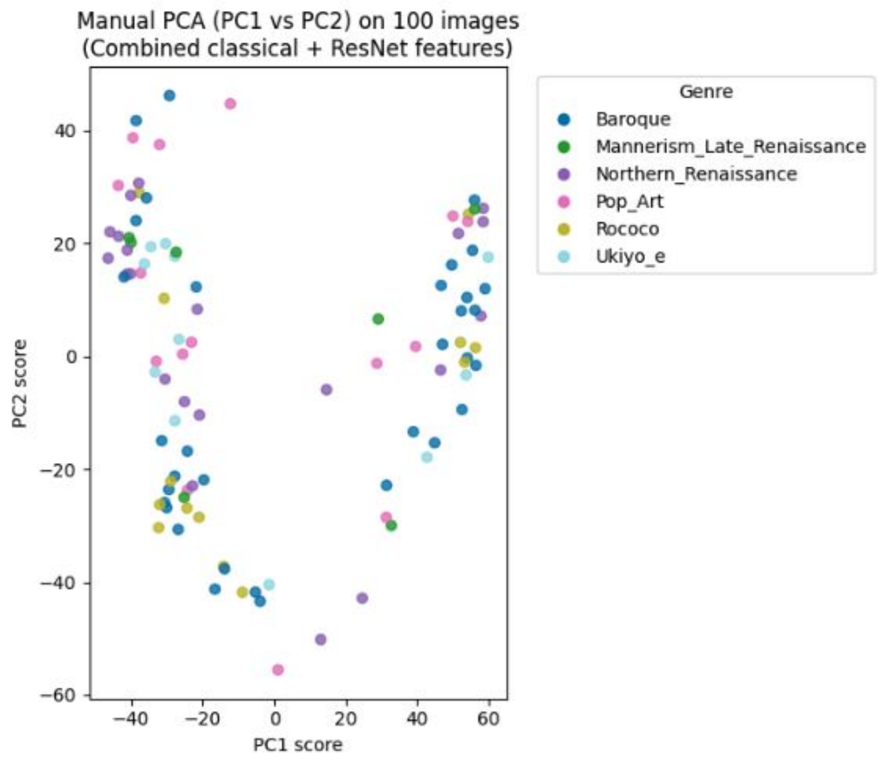


Manual PCA

Manually Computed PCA ->

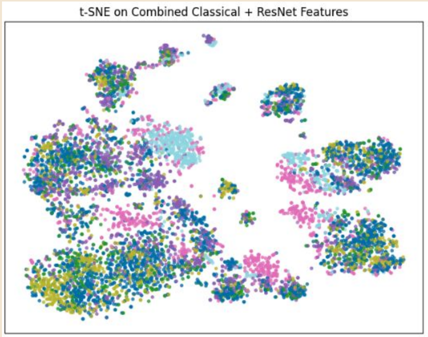
1.) Centered Data
2.) Cov Matrix Calc
3.) Found eigenvalue/vector decomposition
4.) leveraged top 2 components
5.) projected down to 2 dimensions for visualization purposes

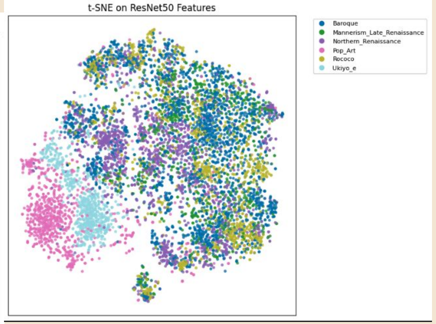
Note: PCA provides a linear embedding of the data, which is useful for global variance structure, but cannot capture non-linear structure. Next we leverage t-SNE, which preserves local + global nonlinear relationships within the data.

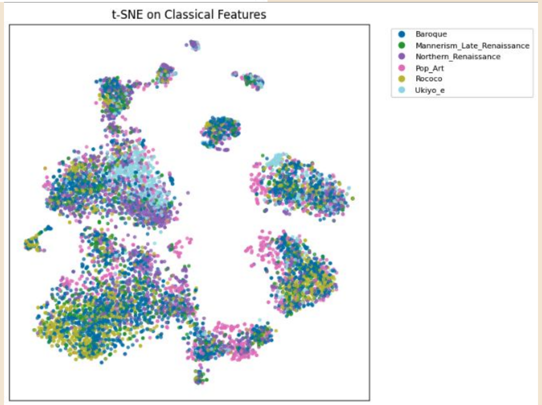


t-SNE

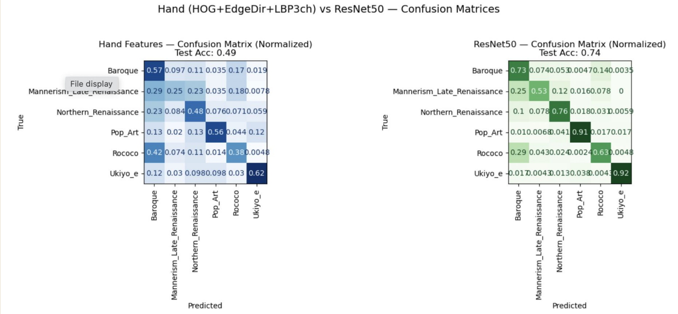
- t-SNE map local neighborhoods from high dimensional space to low dimensional space.
- **resnet and classical**
  - Certain labels have consistent patterns across our 10k dimensional feature space, which t-SNE is able to preserve over 2-D space
  - Far away clusters leverage different feature vectors and push apart into different neighborhoods
  - We see some very heavy overlap with the feature set, this suggests that it is actually quiet difficult to distinguish certain labels with our features.
- When we break apart resnet and classical we can see classical features have many messy/overlapping labels with no strong separation.
- Seems like resnet features dominate combined space

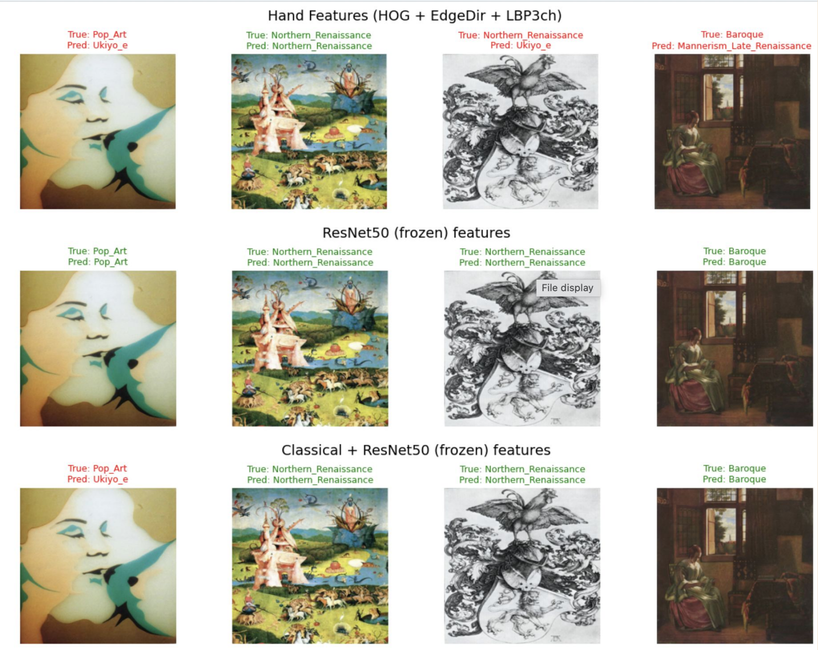






Hand Features
Vs
Complex Feature



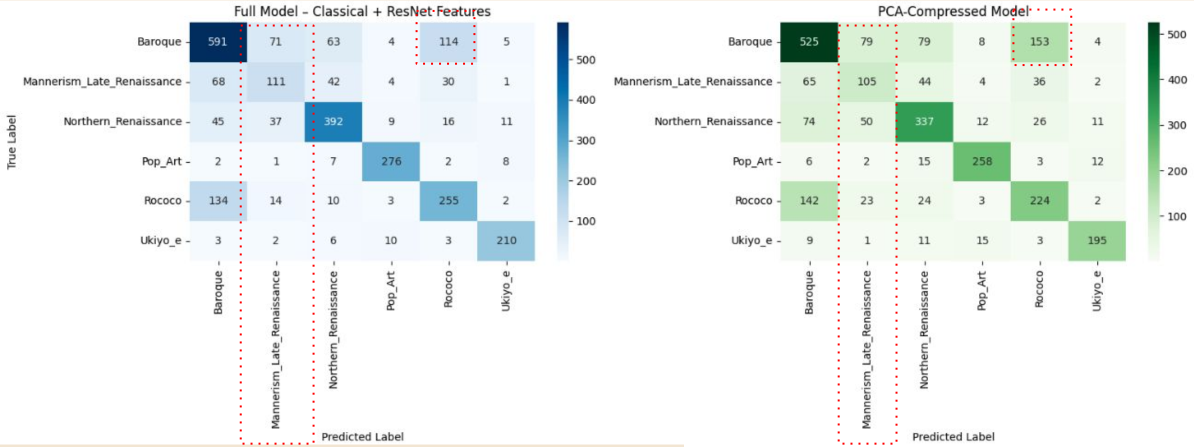


Performance Summary Table

	Model	Validation Accuracy	Test Accuracy	Macro F1	Top 3 Genres (Best Recall)
0	Hand Features (HOG + EdgeDir + LBP3ch)	0.486	0.543	0.461	Pointillism, Contemporary_Realism, Fauvism
1	ResNet50 (Pretrained, Frozen)	0.829	0.865	0.868	Synthetic_Cubism, Fauvism, Analytical_Cubism

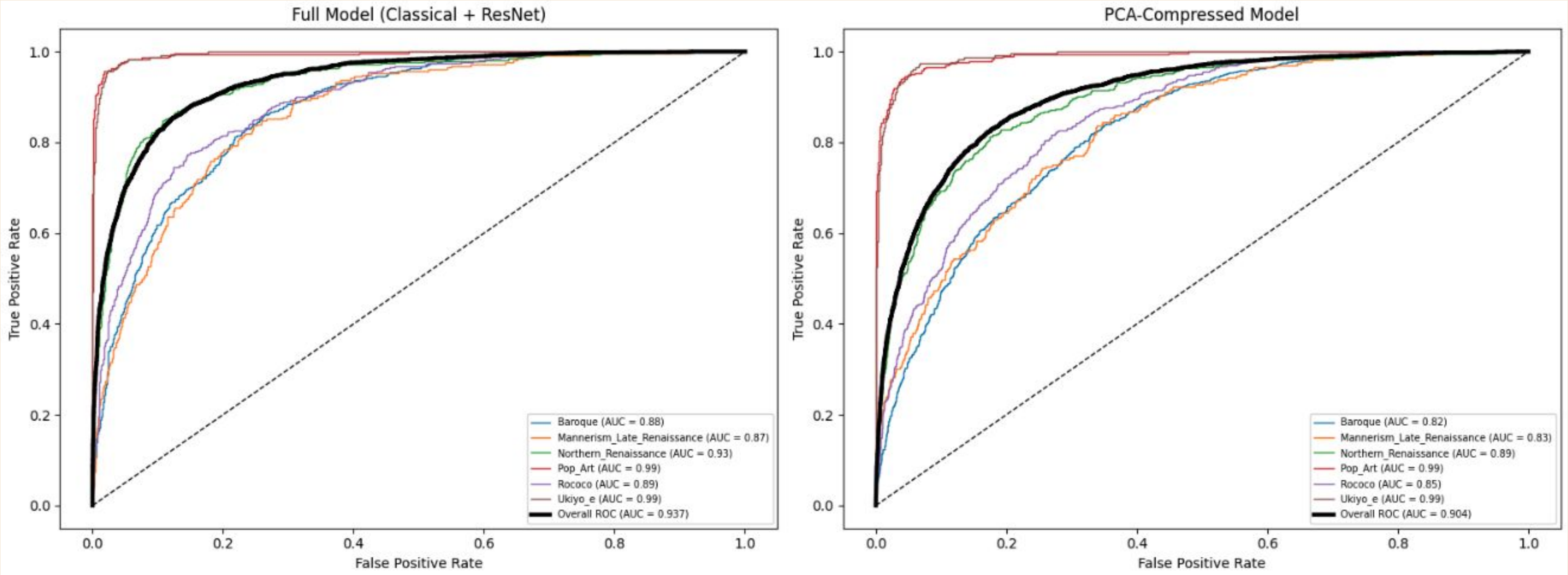
Confusion Matrix - log

	Model	Feature Set	Val Accuracy	Test Accuracy	Micro AUC
0	LogReg Full	Full (Hand + ResNet)	0.719	0.706	N/A
1	LogReg PCA	PCA 90% var	0.655	0.646	N/A
2	CNN Full	Full (Hand + ResNet)	0.704	0.705	0.937
3	CNN PCA	PCA 90% var	0.715	0.722	0.946



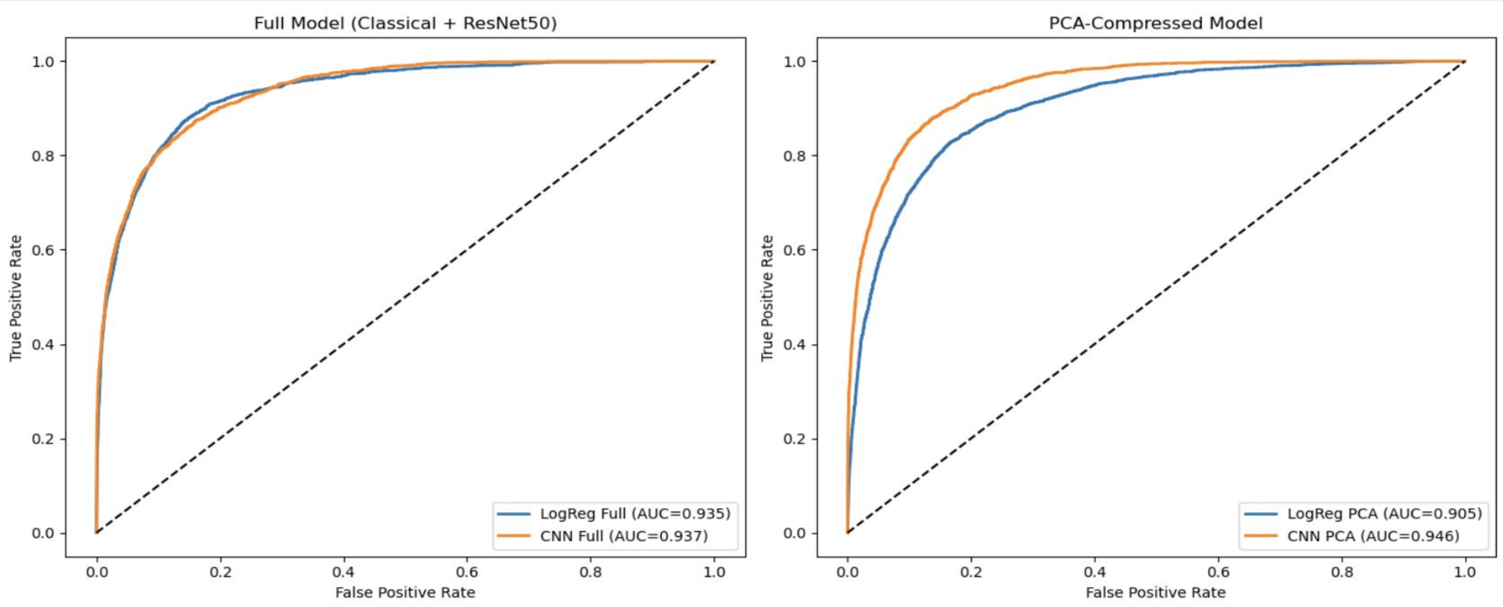
- Baroque and rococo exhibit high degree of overlap
    - t-SNE seem to support this

ROC - log regression



- Inter-genre classification doesn't seem materially worse when comparing between compressed model and full model
- Aggregated AUC is .93 for the full model vs .9 for the compressed model

ROC Comparison: LogReg & CNN Across Full and PCA Features



- Full vs. PCA models show **very similar ROC behavior**.
- Overall AUC stays strong: **0.93 full vs. 0.90 PCA**.
- Compression keeps ranking quality high across genres.

Efficiency Tradeoff - PCA vs Full

	Model	Feature Set – Dimension	Val Accuracy	Test Accuracy	Micro AUC	Training Time (s)
0	LogReg Full	Full (Hand + ResNet) – 10,191	0.719	0.706	N/A	19.151
1	LogReg PCA	PCA 90% var – 1,217	0.655	0.645	N/A	34.682
2	CNN Full	Full (Hand + ResNet) – 10,191	0.726	0.731	0.947	41.990
3	CNN PCA	PCA 90% var – 1,217	0.717	0.722	0.941	7.660

Computational Impact (CNN Classifier):

Full model is heavy to train (~80M ops).

- Training data: 7866 x 10191= 80M~ ops

PCA cuts features sharply (~9.5M ops).

- Trimmed down training: 7866 x 1217 = 9.5M ops

About 6× faster with only a small accuracy drop.

- 8.3x~ times more space efficient and ~6x more time efficient, sacrificing about 1%~ in accuracy overall

Efficiency Tradeoff - log regression

Log Regression

Val Acc: 0.719 | Test Acc: 0.706

(Hand + ResNet):

	precision	recall	f1-score	support
Baroque	0.68	0.68	0.68	848
Mannerism_Late_Renaissance	0.43	0.41	0.42	256
Northern_Renaissance	0.75	0.75	0.75	510
Pop_Art	0.91	0.94	0.92	296
Rococo	0.62	0.62	0.62	418
Ukiyo_e	0.89	0.89	0.89	234
accuracy			0.71	2562
macro avg	0.71	0.71	0.71	2562
weighted avg	0.70	0.71	0.70	2562

PCA-Compressed Model: Val 0.655, Test 0.646

Classification Report (PCA-Compressed Model):

	precision	recall	f1-score	support
Baroque	0.64	0.62	0.63	848
Mannerism_Late_Renaissance	0.37	0.40	0.38	256
Northern_Renaissance	0.68	0.66	0.67	510
Pop_Art	0.89	0.90	0.89	296
Rococo	0.53	0.55	0.54	418
Ukiyo_e	0.83	0.83	0.83	234
accuracy			0.65	2562
macro avg	0.66	0.66	0.66	2562
weighted avg	0.65	0.65	0.65	2562

Computational Impact (log regression):

- Training data: 7866 x 10191=80M~ ops
- Trimmed down training: 7866 x 1217 =9.5M ops
- 8.3x~ times more efficient, sacrificing about 7%~ in accuracy overall

Efficiency Tradeoff - CNN Classifier

Log Regression

CNN FULL | Val Acc: 0.704 | Test Acc: 0.705

Classification Report (CNN FULL):

	precision	recall	f1-score	support
Baroque	0.68	0.69	0.69	848
Mannerism_Late_Renaissance	0.45	0.45	0.45	256
Northern_Renaissance	0.70	0.78	0.74	510
Pop_Art	0.90	0.93	0.91	296
Rococo	0.68	0.52	0.59	418
Ukiyo_e	0.84	0.91	0.88	234
accuracy			0.71	2562
macro avg	0.71	0.71	0.71	2562
weighted avg	0.70	0.71	0.70	2562

CNN PCA | Val Acc: 0.715 | Test Acc: 0.722

Classification Report (CNN PCA):

	precision	recall	f1-score	support
Baroque	0.71	0.71	0.71	848
Mannerism_Late_Renaissance	0.50	0.40	0.45	256
Northern_Renaissance	0.70	0.79	0.74	510
Pop_Art	0.87	0.91	0.89	296
Rococo	0.68	0.63	0.65	418
Ukiyo_e	0.89	0.89	0.89	234
accuracy			0.72	2562
macro avg	0.73	0.72	0.72	2562
weighted avg	0.72	0.72	0.72	2562

Computational Impact (CNN Classifier):

Full model is heavy to train (~80M ops).

- Training data: 7866 x 10191= 80M~ ops

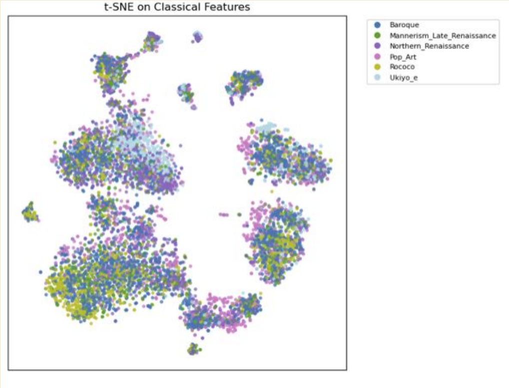
PCA cuts features sharply (~9.5M ops).

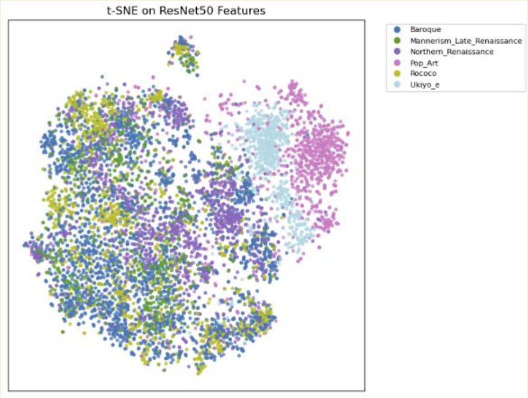
- Trimmed down training: 7866 x 1217 = 9.5M ops

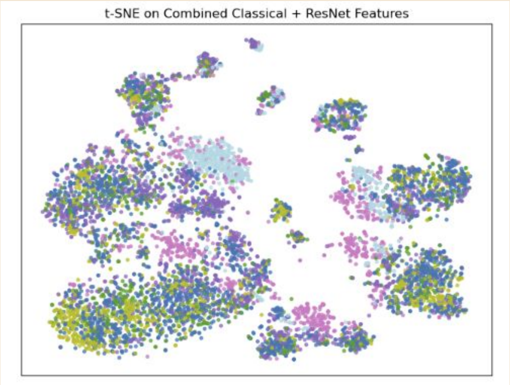
About 6× faster with only a small accuracy drop.

- 8.3x~ times more space efficient and ~6x more time efficient, sacrificing about 1%~ in accuracy overall

Extra


t-SNE on Classical Features
Baroque
Mannerism_Late_Renaissance
Northern_Renaissance
Pop_Art
Rococo
Ukiyo_e


t-SNE on ResNet50 Features
Baroque
Mannerism_Late_Renaissance
Northern_Renaissance
Pop_Art
Rococo
Ukiyo_e


t-SNE on Combined Classical + ResNet Features