The Stylistic Fingerprints of Art Genres

Ofek Peres
Department of Mechanical Engineering
Princeton University
operes@princeton.edu

Divyanshu Pachisia
Department of Mechanical Engineering
Princeton University
divyanshupachisia@princeton.edu

Abstract

Given the wide variety of paintings throughout history, the stylistic differences between painting genres is highly subjective and often requires the judgment of an 'expert'. Recently, there have been attempts at quantifying painting style using Convolutional Neural Networks (CNN), indicating some quantitative representation is possible. Using the dataset *Best Artworks of All Time* and extracting features from paintings using CNN's, this project measures the effectiveness of more traditional supervised classifiers to identify a genre within Impressionism, Post-Impressionism, Renaissance and Baroque. Additionally, clustering is performed to find whether the latent structure of the paintings coincides with the categorization of genre. Overall, genre could be successfully predicted from extracted features, with an average AUC of 0.95 in one vs. all classification. Additionally, multi-class classification was also performed with a high accuracy of 0.90. Interestingly, linear models outperformed non-linear models indicating that the classification problem is a linear one. However, the latent structure revealed through clustering suggests that there may be a different underlying structure to the paintings that could transcend genres.

1 Introduction

The study of artwork is a very subjective field with multiple, diverging interpretations of a given artist's and genre's style. The subjective nature of the field often requires in depth immersion to acquire an understanding, making it inaccessible. By approaching this esoteric field from an analytic lens, we hope to demystify some of the ambiguity surrounding art. For instance, if art could be successfully classified by genre using quantitative approaches, then a definitive underlying style can be expressed. Additionally, it opens up the possibility for tools to be built that make art more accessible, such as a virtual assistant that allows users to interact with art and learn about it in a more structured manner.

Furthermore, a quantitative approach to art could provide a tool by which new artists' works are examined, interpreted and displayed. For example, new paintings could be run through a quantitative model to help inform price. When encountered with a new painting, a machine learning model could show similar and different pieces of work which may be helpful to experts during evaluation. Similar models could be trained to benefit museums by generating exhibit options after surveying thousands of paintings and past exhibits.

Finally, from a computer science perspective, using supervised classification methods on features extracted from unsupervised learning is an interesting approach. It is possible to use Convolutional Neural Networks (CNN's) for both feature extraction and classification [2]. CNN's, often referred to as "black boxes", generally perform well with image recognition and processing but offer less interpretability than traditional supervised classification methods do [3]. If it is shown that supervised methods (with their interpretability) can be used after unsupervised methods (which process images well) then we can perhaps reduce the trade-off between interpretability and performance.

2 Related Work

In the paper, *A Neural Algorithm of Artistic Style*, Gatys et al. demonstrate that artistic style can be extracted from an image, offering a "path forward to an algorithmic understanding of how humans create and perceive artistic imagery" [1]. Gatys et al. extracted content from a 'content image' and style from a 'style image'. They then combined the two images to form a new image with the content of the 'content image' painted in the style of the 'style image', as shown in Figure 1. This result was accomplished by minimizing 'content loss' and 'style loss'.

Figure 1: Houses (content image) painted in the style of different painting (style images) [1]

The applications of finding underlying structure in paintings have also been explored, including using models to detect forgeries in paintings [4]. Polatkan, et. al show that some supervised techniques can also detect changes in style, while providing interpretability with the results such as the effect of "image clarity". [4] This work also shows that providing a quantitative model to characterize art has useful applications.

Past work shows that style can be extracted from a given painting and in this project we extend this to see if it can extract a genre's style across paintings.

3 Methods

3.1 The Data-Set

The data set *Best Artworks of All Time* was found on Kaggle [5]. It contains over 8400 paintings from 50 of the most famous artists. These paintings span a variety of genres and time periods. The four genres which had both a significant amount of similarities and differences between art styles and had the largest quantity of paintings were chosen for analysis. A total of 3560 paintings were used: 556 in Renaissance, 586 in Baroque, 1370 in Impressionism and 1048 in Post-Impressionism.

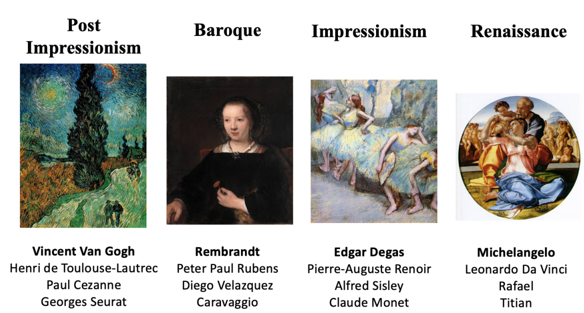

Figure 2: Chosen Genres and Artists

3.2 Preprocessing Techniques

The first preprocessing step performed was to extract features using two neural networks. Next, to reduce model complexity, we used Principle Component Analysis (PCA) which reduced the size of our preprocessed data. After these two steps, the data was ready to be classified and clustered.

3.2.1 Extracting a vector representation of the painting using Img2Vec

Img2Vec uses "pre-trained models in PyTorch to extract a dense vector embedding for any image" [6]. It does this through multiple iterations of convolution and pooling. In convolution, a filter (matrix) of a particular window size is 'slid' along the image and the dot product of this filter and the RGB pixel values in the window is taken. Through pooling, the dimensionality of the convolution is reduced, by finding the maximum or average value in every window of the image. This process initially extracts low-level features like lines and then as the number of layers increase it extracts higher level features [9].

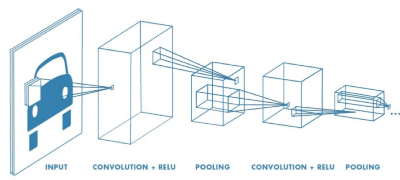

Figure 3: Convolutional Neural Network Visualization [7]

Img2Vec uses the pretrained RESNET-18 Neural Network and extracts a 512x1 feature vector for every painting that is a result of layers of convolution and pooling.

3.2.2 Extracting 'style' using Gram Matrices

After every layer in a convolutional neural network, the activation output F_{ij} is extracted, where i denotes the filter and j denotes the position. This vector is multiplied with its transpose for each filter to compute the gram matrix, as in the equation below. This produces a representation of style as the transpose contains only stylistic similarities and not content-wise similarities and so the multiplication produces a vector that captures style. [8]

$$GramMatrix = \Sigma_k F_{ik} F_{jk}$$

As can be seen in Figure 4, the gram matrices that were extracted from the different layers of Resnet18 have a clear physical interpretation: they are different layers of style in the image. The 1$^{\text{st}}$ layer represents the low level, horizontal brush/pencil strokes that are evident in the background of the original image. The 5$^{\text{th}}$ layer encapsulates finer brush strokes used for the curly hair and beard.

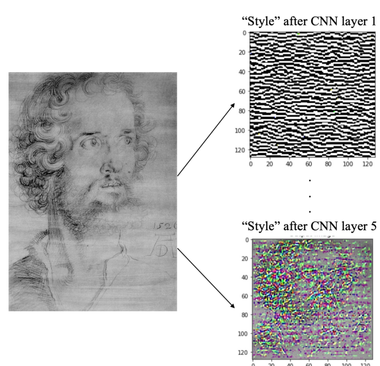

Figure 4: Visualizing Gram Matrices using Style Transfer as in Gatys et.al

In total, 5 style layers were extracted as gram matrices. Upon flattening each matrix into a 1 dimensional row, there were over 100,000 feature columns extracted per painting.

3.2.3 Dimensionality reduction using Principle Component Analysis

Given that the vectors extracted using Img2Vec and the gram matrices were very large, the feature space needed to be reduced in dimension to reduce model complexity, computational power required, and running times. Through eigenvalue decomposition, PCA reduced the feature set by linearly combining existing features to form new features. Our feature set was reduced from over a 100,000 features to 4,000 features.

3.3 One vs. All Classification

One vs. All Classification is a method for multi-class classification where the classification is reduced to a binary problem by comparing each class to all the others (as a group). This gave us a measure of how well a classifier predicted each genre individually.

1. Support Vector Machine (Linear Kernel): a model that attempts to find the line that maximizes the distance between any two, separately labeled groups of data points in n-dimensional space. The linear kernel assumes linearity in the data.

   Hyperparameters: Kernel which was chosen to be linear after cross validation to maximize AUC.

2. Naive Bayes Gaussian: Naive Bayes splits the training data into sections by label and fits a Gaussian curve to every separately labeled data section. Upon testing, it compares how well each new input fits each classes' Gaussian curve and classifies based on best fit. It assumes identically and independently distributed data.

3. Random Forest: Constructs decision trees that are each trained to classify a painting. The final classification is the mode of each trees' output classification. Unlike linear models, random forests can capture non-linear interaction between the features and the target.

   Hyperparameters: Number of estimators (number of trees) and Max Depth were tuned to be 500 and 10 using cross validation to maximize AUC.

3.3.1 Evaluation Metrics for One vs. All Classification

A Receiver Operating Characteristic curve was plotted for the classification for each genre, which plots the True Positive Rate versus the False Positive Rate. The Area under the Curve (AUC) estimates the probability with which a randomly selected sample with classification 1 will be classified correctly compared to a randomly selected sample with classification 0. An AUC of 1 indicates perfect classification, whereas an AUC of 0.5 indicates classification equivalent to a coin toss, which makes it a good estimator of model performance for binary classification.

3.4 Multiclass Classification

A more traditional approach to a classification task with multiple labels is to predict the label of every painting, as opposed to one vs all classification. The following models were used for multiclass classification:

1. Random Forest: Using decision trees as in one vs. all classification to capture non-linear relationships.

   Hyperparameters: number of estimators (number of trees) and Max Depth which was tuned which was tuned to be 500 and 30 using cross validation to maximize accuracy.

2. Logistic Regression: Finds a set of weights for the features that minimizes classification error on training data, using Stochastic Gradient Descent. It guards from over-fitting to the training data using a regularization penalty, λ. To predict classes it multiplies the weights with the features to output a probability that a painting belongs to each genre.

   Hyperparameters: C ($\frac{1}{\lambda}$) is tuned using LogisticRegressionCV, an inbuilt cross validator version of Logistic Regression in sklearn.

3. K-neighbors classifier: Finds the K-nearest points to a given point, p, and classifies p as the mode classification label of those nearest points.

   Hyperparameters: K (number of neighbours) - tuned using GridSearchCV cross validation in sklearn.

3.4.1 Evaluation Metrics for Multiclass Classification

1. Accuracy: The proportion of labels that were predicted correctly.
2. Confidence for each class: In our multiclass classification models, probabilities that a given sample belongs in each class is computed. Classification is performed by picking the label with the highest probability for every sample. The confidence measure is the mean probability assigned to all the samples classified as a particular genre. This measure acts a way to evaluate how sure our model is when it classified a painting as a particular genre.

3.5 Clustering

Clustering was run on the PCA-reduced featurized dataset (without the labels) to reveal any latent structure in the data. The models below were used to cluster in the data into four categories and then they were analyzed to see how they correlate to the existing categorization of genre.

1. K-Means Clustering: Finds k centroids (corresponding to each cluster) in the dataset that best maximizes distance between centroids. It classifies each data point as belonging to the closest centroid.

   Hyperparameter: Number of Clusters (k), chosen as 4 to mimic genre.

2. Latent Dirichlet Allocation: Unsupervised topic modeling that finds topics within unlabeled data. LDA finds the joint distribution of the hidden variables (latent structure) and observed variables (features). Using the Bayesian estimate, the log likelihood of the posterior probability is maximized on paintings never seen in order to cluster them.

   Hyperparameter: Number of Clusters (k), chosen as 4 to mimic genre. Prior of topic distribution (α), 11.25 chosen using cross validation to maximize silhouette score.

3.6 Evaluation Metrics for Clustering

1. Silhouette Score: This measures homogeneity within a cluster and heterogeneity between clusters. A positive silhouette score indicates that the points within a cluster are close together but are far from other clusters, while a negative score indicates overlap. The silhouette score is calculated for each data point using Equation 1, where a measures average distance within a cluster and b measures average distance to all points in any *other* clusters.

$$S = \frac{b - a}{max(a, b)} \tag{1}$$

4 Results

4.1 One vs All Classification

As seen in Figure 5, Support Vector Machine with a linear kernel had the best results. It had an average AUC greater than 0.93 for every genre. Naive Bayes Gaussian was used as a baseline comparison and had an average AUC of 0.638. Random forest performed well but not as well as SVM with an average AUC of 0.783.

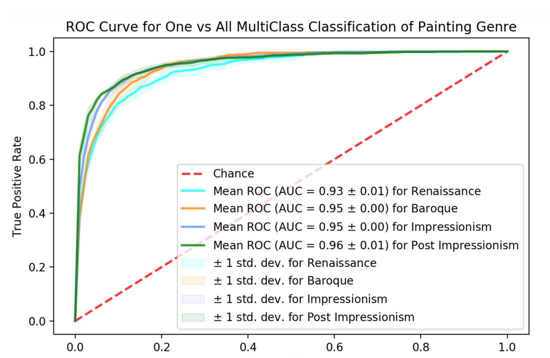

Figure 5: Multi-class Classification with SVM - Linear Kernel

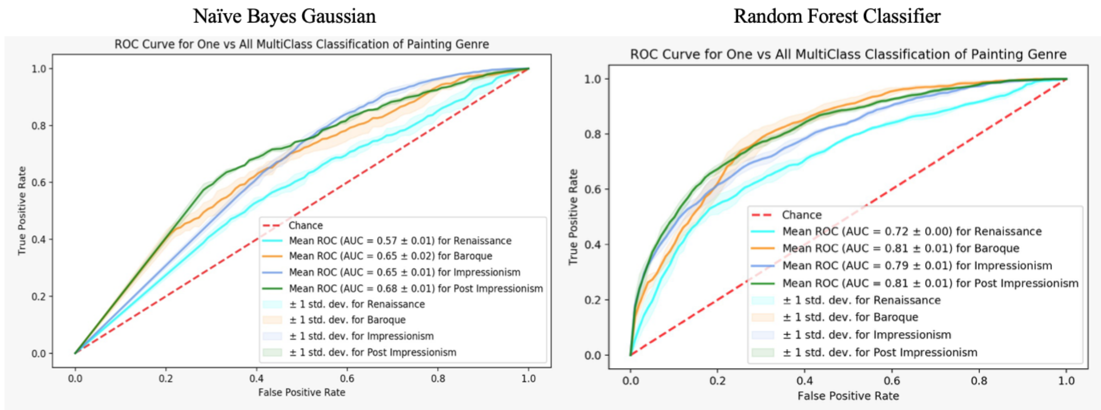

Figure 6

4.2 Clustering

Both clustering methods produced large clusters that were close together and so had low silhouette scores - K-Means had an average score of 0.08 and LDA produced a score of -0.01. This indicates the the boundaries between the latent clusters (differentiated by color) are blurry. Additionally, the shapes of the points show the genre they belong to, as indicated by the legend.

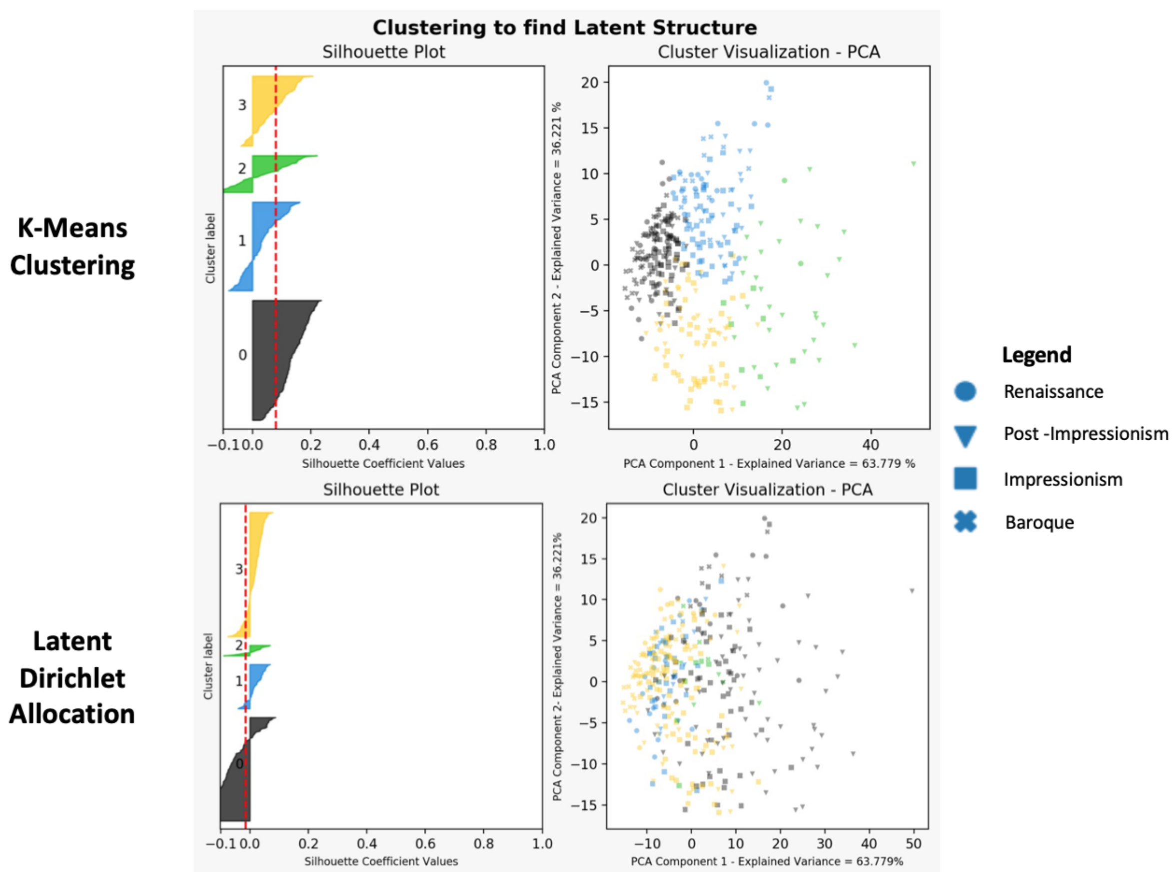

Figure 7

4.3 Multiclass Classification

With multiclass classification, Logistic Regression had the highest accuracy score of 91%. Interestingly, logistic regression had an average confidence of about 70% in its predictions for all of the 4 genre classifications. K-Nearest Neighbors had an accuracy of 77%. Its average confidence for Renaissance and Baroque was about 70% but its confidence in predicting impressionism and post impressionism was around 85%. Lastly, Random Forest Classifier had an accuracy of 50%. Interestingly, Random Forest *never* predicted that a painting was Renaissance. It predicted baroque with a confidence of 30% and both Impressionism and Post Impressionism with an accuracy of 40%.

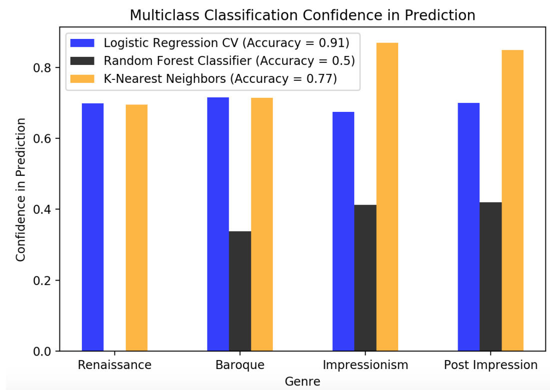

Figure 8

5 Discussion

5.1 Quantitative models can predict painting genre

Both the one vs all and multiclass classification produced an 'accuracy' of greater than 90% indicating that there are quantitative models that can be used to understand art. Additionally, they performed well across cross validation folds, indicating little overfitting.

5.2 Classification of art genres: a linear problem?

The classification (both one vs.all and multiclass) results demonstrate a very interesting trend: linear models were significantly more successful in classification tasks than non-linear models. Specifically, in One vs All Classification, an SVM with a linear kernel outperformed the robust Random Forest algorithm. With multi-class classification, logistic regression achieved an accuracy 41% greater than Random Forest. These results support the idea that there is a simple, linear relationship between features and genre. This is surprising given the complexity of the task, and it is somewhat incredible that the PCA-reduced neural network generated features result in a simple correlation with class. This result also testifies to the usefulness of running simple supervised models after extracting features from unsupervised methods, as it lends itself to more transparent interpretation, such as the linearity in the data.

5.3 Misclassified Paintings

The venn diagram seen in figure 9, represents the number misclassified paintings by genre in one vs all classification using SVM. Intersections between genres show the common paintings that were misclassified. As can be seen in figure 9, the intersection between the genres Impressionism and Post Impressionism as well as the intersection between the genres Renaissance and Baroque contain more paintings in common than any other intersections. This is interesting as it represents the similarity between impression/post-impression and baroque/renaissance.

Additionally, when delving deeper into particular misclassified paintings we find features of the painting that result in the misclassification, as shown in Figure 9. Further analysis of these features could point towards how to improve the extracted feature set.

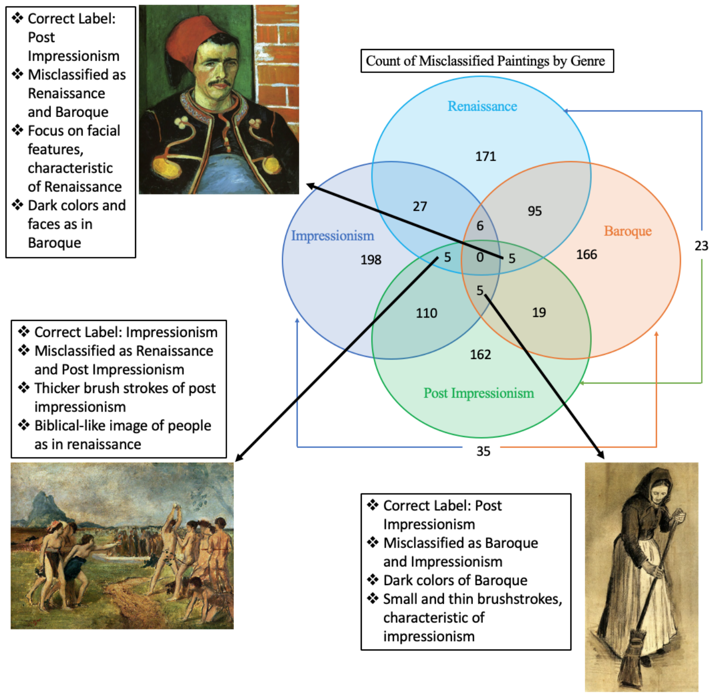

Figure 9

5.4 Latent Structure and its overlap with Genre

Examining the clustering models, we find that overall, the latent structure does not coincide with the division by genre. This suggests that there may be an underlying categorization that is different from genre which is represented by PCA components 1 and 2. Analyzing these PCA components gives us no physical intuition about the clusters as it combines entries across gram matrices and the img2vec representation. This lack of intuition about latent structure is a disadvantage of using unsupervised methods.

However, we do find that, on average, Post-Impressionist /Impressionist paintings (Set A) and Renaissance/Baroque (Set B) are close together. This indicates that the latent structure does recognize that the styles of Set A and Set B are different. The structure also shows that within the sets genre may be an ambiguous distinction, consistent with the majority of misclassified paintings, as discussed with regard to Figure 9.

6 Conclusion and Future Work

Through this project, we have successfully shown that extracting features from paintings using convolutional neural nets and then passing them into supervised classifiers can successfully classify paintings by genre. Interestingly, the superior performance of linear models show that features produced seem to linearly correlate to the class labels, indicating an underlying simple structure. The latent structure produced with clustering did not coincide with genre, which may indicate that there is a stronger categorization of paintings that transcends genres. This is especially true for Impressionism and Post Impressionism and Baroque and Rennaisance which were often confused both within clustering and classification.

Future work could look more closely at the misclassified paintings to then inform feature extraction techniques that produce features that are better predictors between similar genres. Additionally, further analysis on clustering could produce a different categorization of art independent of similar genres. The same framework outlined in this paper could also be used to compare individual artists. Finally, these results could be applied to make applications for teaching art and art genres in a more structured manner, making it more accessible.

Acknowledgments

Professor Barbara Engelhardt
Jonathan Lu
Diana Cai
The SciKit Learn library which was used for all the code for this assignment

References

[1] Leon A. Gatys and Alexander S. Ecker and Matthias Bethge, (2015),A Neural Algorithm of Artistic Style, `http://arxiv.org/abs/1508.06576`

[2] Dan C. Ciresan and Ueli Meier and Jürgen Schmidhuber, (2012), Multi-column Deep Neural Networks for Image Classification, `http://arxiv.org/abs/1202.2745`

[3] Girshick, Ross and Iandola, Forrest and Darrell, Trevor and Malik, Jitendra, (2015), Deformable Part Models are Convolutional Neural Networks `https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Girshick_Deformable_Part_Models_2015_CVPR_paper.html`

[4] G. Polatkan, S. Jafarpour, A. Brasoveanu, S. Hughes and I. Daubechies, ”Detection of forgery in paintings using supervised learning,” 2009 16th IEEE International Conference on Image Processing (ICIP), Cairo, 2009, pp. 2921-2924

[5] `https://www.kaggle.com/ikarus777/best-artworks-of-all-time`

[6] `https://github.com/christiansafka/img2vec`

[7] `https://becominghuman.ai/extract-a-feature-vector-for-any-image-with-pytorch-9717`

[8] `http://www.subsubroutine.com/sub-subroutine/2016/11/12/painting-like-van-gogh-with-convolutional-neural-networks`

[9] `http://www.subsubroutine.com/sub-subroutine/2016/9/30/cats-and-dogs-and-convolutional-neural-networks`