Machine Learning and Feature Engineering for Artist Classification

Aris Faria Chavez
Carnegie Mellon University
afariach@andrew.cmu.edu

Filip Nowicki
Carnegie Mellon University
fnowicki@cmu.edu

Shukan Parekh
Carnegie Mellon University
shukan@cmu.edu

Sagar Sachdev
Carnegie Mellon University
sagarsac@andrew.cmu.edu

Abstract

This paper describes an approach to automatically classify digital pictures of paintings by artist using shallow machine learning techniques. Traditionally, the task of classifying paintings has been entrusted to experts. The major drawback to this approach is that it usually takes years of training to understand the intricacies of an artist's painting style. We believe that Machine Learning can be utilized to automate this process and increase both its speed and accuracy. The approach outlined in this paper uses images of paintings from, arguably, fifty of the most influential artists of all time and presents a comparison between different feature extraction techniques and classification methods.

1. Introduction

With the advancement of computer vision, specifically, advancements in algorithms to detect edges, corners, shapes, and patterns, there is a significant amount of research being conducted in the scientific community to harness the power the of this science. The biggest type of problem that computer vision is being used to solve is one classification and naturally there is a huge potential to use Machine Learning techniques to help solve this problem. In this paper, we have investigated the application of computer vision augmented with shallow machine learning techniques in the task of classification.

The specific problem we would like to investigate is the classification of paintings by artist. Recently, much research has been conducted in this field. Specifically in applications such as classifying paintings by styles. However, we hope to investigate the applications in a field that is more specific [1,2]. This is a field that is traditionally the realm of human experts who have spent their lives learning about a particular style of painting, a particular period in in history or a particular artist. We believe that using Machine Learning techniques, we can develop a framework to classify artists that will be agnostic of painting style, period, or artist [3].

The following sections will highlight how we collected our data, detail the feature extraction methods we investigated and analyze the characteristics of classification methods we considered. Finally, we will conclude with a discussion about the performance of each of these methods.

2. Methods

In this section, we will delve into the details of how we collected our data, specifics of and reasoning behind our feature engineering techniques, and implementations of classification algorithms. The focus of our efforts focused, first, researching, distilling relevant feature extraction techniques followed by implementing classification methods. We also present a comparison between the performance of shallow machine learning techniques and that of a convolutional neural network.

2.1. Data Collection

The dataset utilized in this analysis was a collection of digital images of 50 influential artists. In total the data set contained 8446 images. In addition to these high-resolution images, the dataset also contained resized pictures of the painting to facilitate faster processing. The images were collected from Wikipedia and compiled in a folder structure. This dataset was effective because it contained works from different periods in history, 31 different countries of origin and 44 different genres. In theory, such distinctiveness would allow our model to be trained on different painting styles, color palettes, and types of subjects [4].

2.2. Feature Engineering

In order to extract meaningful for information from the digital images, it was crucial to account for all the factors that differentiate the paintings of one artist from another's. In this section we propose feature extraction methods that, we believe, work best in the task of quantifying painting characteristics. There were three major classes of methods that we looked at. The first class of methods contained those extracted from grayscale images. These included Local Binary Patterns, Hu Moments, and Haralick Textures. The second class of extraction methods investigated were those obtained using color images. These included HSV Histogram, RGB Histogram, and Most Representative Colors. The final class of extraction methods considered were gradient-based. These included SIFT Keypoints, HOG Descriptors, and Canny Edge Sum.

Figure 1: The digital image Fall of the Rhine at Schaffhausen by William Turner. This image will be used to illustrate the feature extraction methods.

2.2.1. Local Binary Patterns

Local Binary Patterns is a visual descriptor used for classification in computer vision. Popularized in 2002, it is implemented by picking a reference pixel in the image and using it as a threshold for surrounding pixels. An LBP value is then calculated by the reference pixel and stored. This process is repeated for each pixel in the image. The LBP values are then imported into a histogram resulting the final feature vector. The biggest advantage of this method is that it allows for extraction of fine details in an image. However, it can also be deemed This also serves as a disadvantage as since it does not allow for varying the scales [5, 6].

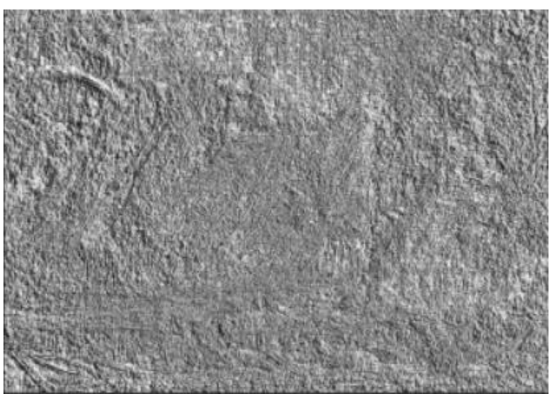

Figure 2: Local Binary Patterns applied to Figure 1.

2.2.2 Hu Moments

Moment invariants, first introduced by Hu, consist of 6 orthogonal invariants and one skew orthogonal invariant. This invariant are independent of size, position, orientation, and parallel projection and have been extensively been applied in pattern recognition [7].

	Hu Moments
1	2.877772438
2	6.69444447
3	10.76477311
4	10.71688617
5	-22.074117784
6	14.12863889
7	-21.47080317

Table 1: Hu Moments obtained from the image in Figure 1

2.2.3. Haralick Textures

Texture analysis was developed in the 1970s as a method of featurizing images by extracting the spatial distribution of densities in images. Haralick Textures describes 14 descriptive features for an image obtained from a Gray Scale Co-occurrence Matrix. Rotational invariance was obtained in this matrices by four co-occurrence matrices in four different directions of adjacency [8,9].

	Haralick Textures
1	1.91389639E-04
2	1.88252457E02
3	9.64853109E-01
4	2.67786772E03
5	1.67961488E01
6	2.34110562E02
7	1.05232184E04
8	8.65074276
9	1.30494749E01
10	2.3182-780E-04
11	4.54523255
12	-2.97902648E-01
13	9.94396219E-01

Table 2: Haralick Textures obtained from Figure 1

2.2.4. HSV Histogram

The HSV (Hue, Saturation, Value) color space, closely corresponds to the human visual perception of color. The HSV color space can be represented as a three-dimensional hexacone [10]. Hue is defined as an angle from 0 to 360 degrees and each value represents a different color. Saturation is the depth or purity of color and is measured as a radial distance from the central axis to the outer surface. Finally, Value is the brightness of a particular color [11].

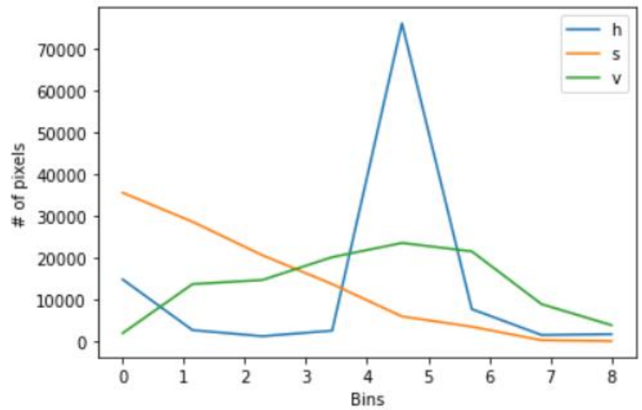

Figure 3: HSV Histogram obtained from Figure 1.

2.2.5. RGB Histogram

The color histogram is a simple set of 3 histograms, representing the distribution of color appearances in an image. The histogram has no representation of the spatial distribution of the colors, only their appearance [10].

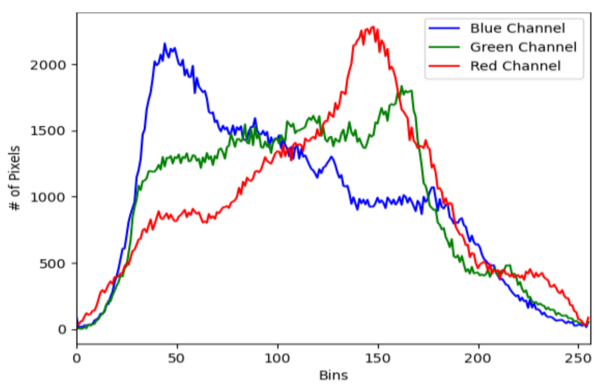

Figure 4: RGB Histogram obtained from Figure 1.

2.2.6. SIFT Keypoints

The Scale-Invariant Feature Transform converts an image to a large collection of feature vectors which are all rotationally and translationally invariant. The process involves obtaining a list of keypoints in the image and localizing them to nearby data points to obtain accurate fit of location, scale, and ratio of curvature. Then the edge and low contrast features are discarded to obtain the final list of keypoints [12].

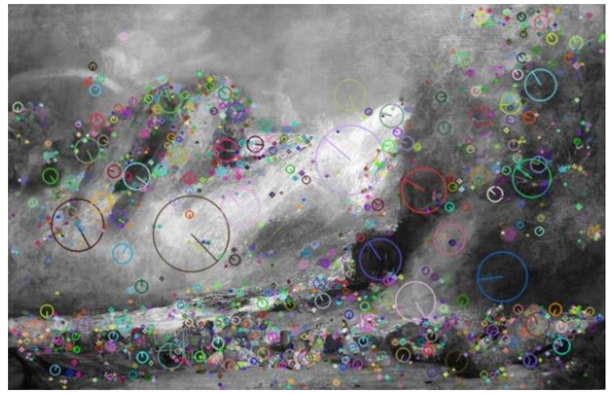

Figure 5: SIFT Keypoints of Figure 1.

2.2.6. HOG Descriptors

Histogram of Gradients (HOG) is a descriptor that focuses on the overall structure of an image. It contains information about the edges of an image and their associated directions. HOG descriptors for a given pixel are calculated by determining the change in intensity of the pixel in all directions. This process is repeated for all pixels and input into a histogram to obtain a feature vector that describes the entire image [13].

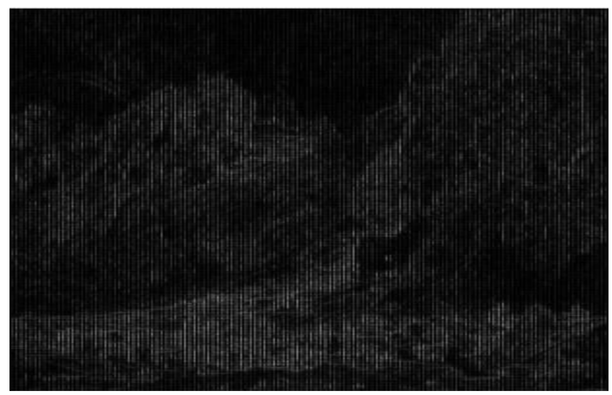

Figure 6: Gradients associated with each pixel in Figure 1.

2.2.7. Canny Edge Detection

Canny Edge Detection is a an edge detection method that locates edges of grayscale images by determining the change in intensity values of a pixel in the vertical and horizontal directions. Weak intensity changes and noise in the images are removed through a combination of Gaussian Blurring, Double Thresholding, and Non-Maximum Suppression [14].

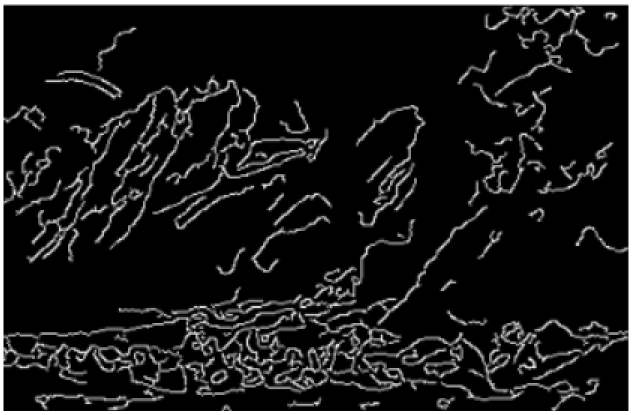

Figure 7: Detected edges of Figure 1 obtained using Canny Edge Detection.

2.4. Classification Methods

A major goal of this project was to compare and contrast the performance of shallow machine learning methods in a real-world classification problem. The methods considered were Softmax Regression, Random Forest Classifier, Gaussian Naïve-Bayes Classifier, K-Nearest Neighbors, Support Vector Machine, and AdaBoost. We optimized the performance of each classifier by tuning the hyperparameters and testing them with various combinations of feature extraction techniques. To measure the model's accuracy we performed 5-fold cross validation. Each of these classification algorithms was implemented using the scikit-learn package.

2.5. Convolutional Neural Network

There has been a huge push in the scientific community to apply convolutional neural networks in classification tasks. CNN's consist of layers that pool and filter data to extract the most useful information. The features obtained from a CNN are rotationally and translationally invariant making it extremely effective in image classification [15].

3. Experiments

3.1. Shallow Machine Learning Methods

This section highlights the experiments we conducted to obtain the optimized combination of extraction techniques and classification method. To do this, a set of artists was picked from the larger data set. During our experimentation, we picked the works of Paul Rembrandt, Vincent van Gogh, and Paul Cezanne. Doing so, allowed us to perform our experimentation with the smaller subset of 1276 images and optimize the model.

The first experiment we undertook was to find the most suitable classification method for this type of dataset.

Naturally, we tried to accomplish this by combining feature vectors obtained from every extraction method and applying every classification algorithm to this combined vector. The highest accuracy obtained for three artists using this method was 90.28% yielded by the Random Forest Classifier with 21 trees.

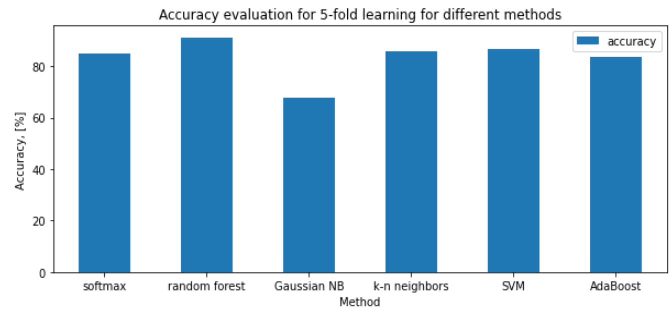

Figure 8: Bar chart depicting the accuracy for every classification algorithm using a combination of every feature extraction method.

It was also crucial to compare the performance of each feature extraction methods and understand the variation in accuracy with different classification methods. Therefore, the next step in this process was to run all classification methods with each extraction technique. Through this exercise we were able to get a sense of which features were effective with this data set. In this case Haralick Features and HSV Histogram were the most effective.

	Softmax	**Random Forest**	**Gaussian NB**	**KNN**	**SVM**	**AdaBoost**
RGB Histogram	82.37	85.89	64.58	85.03	86.52	64.42
HSV Histogram	82.21	85.03	58.62	86.52	86.68	74.84
Hu Moments	80.17	76.49	78.29	75.00	74.92	78.68
Haralick Textures	83.54	85.35	82.37	75.55	72.10	75.25
HOG	71.24	63.56	70.77	64.57	70.53	67.16
Canny Edges	68.73	66.30	66.14	68.97	74.61	73.59
SIFT	68.81	69.90	73.43	74.06	80.25	79.78
LBP	71.47	80.80	75.32	82.60	73.74	84.17

Table 3: Accuracy obtained for every combination of feature extraction method and classification method for 3 artists

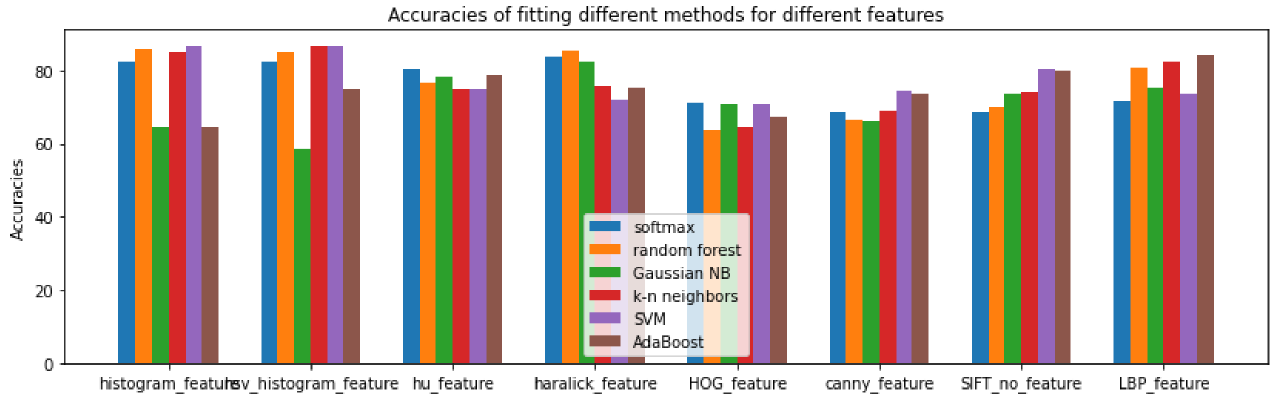

Figure 9: Bar chart depicting the accuracy for every feature extraction method and every classification method.

Through this process, we were able to determine that the HSV Histogram, HOG Descriptors, Haralick Features, and Canny Edge Detection were the most promising extraction techniques. We tested various combination of the aforementioned methods with all the classifications techniques under consideration. This way, we were hoping and to gauge how the accuracy would vary with across combinations and classification methods.

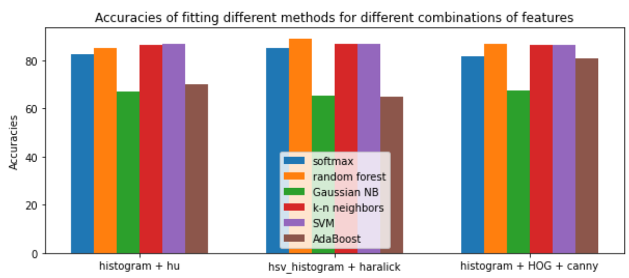

Figure 10: Bar chart depicting the accuracy of different combinations of extraction methods and every classification method.

While performing this analysis, it becomes clear that such an ad-hoc approach to optimization would not be robust. We then proceeded to test each classification method with every combination of feature extraction techniques to ensure that we would pick the most accurate one. This approach required us to compare and contrast between 255 combinations. After completing those calculations, we were able to determine many combinations of extraction methods that gave us higher accuracy than previously obtained. A plot of the accuracy of each combination of extraction methods vs. model accuracy is presented below.

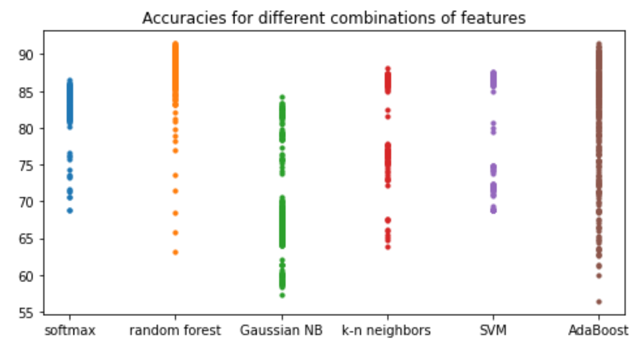

Figure 11: Scatter plot depicting the accuracy of all 255 combinations extraction methods for all classification methods

After obtaining a list of the best performing combinations, we proceeded to investigate the variation in performance with an increasing number of artists. Out of all the top combinations tested, we found that the combination of RGB Histogram, Hu Moments, Haralick Features, and Canny Edge Sum yielded the best accuracy with a growing number of randomly selected artists.

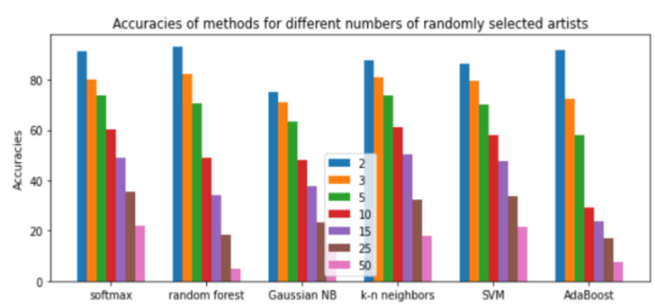

Figure 12: Bar chart depicting the accuracy of the best combination of extraction methods for varying numbers of artists

From the above results we were able to determine the best classification model for each value of randomly selected artists.

	No. of Artists	Best Method	Accuracy (%)
0	2	Random forest	93.22
1	3	Random forest	82.15
2	5	Softmax	73.85
3	10	K-N Neighbors	60.92
4	15	K-N Neighbors	50.16
5	25	Softmax	35.54
6	50	Softmax	22.10

Table 4: Best classification method for each value of artists considered.

Artist No.	Shallow ML Accuracy (%)	CNN Accuracy (%)
2	93.22	92.08
3	82.15	90.57
5	73.85	80.13
10	60.92	70.83
15	50.16	61.19
25	35.54	48.65
50	22.1	40.28

Table 5: Comparison of the accuracy between the optimal shallow classification method and CNN

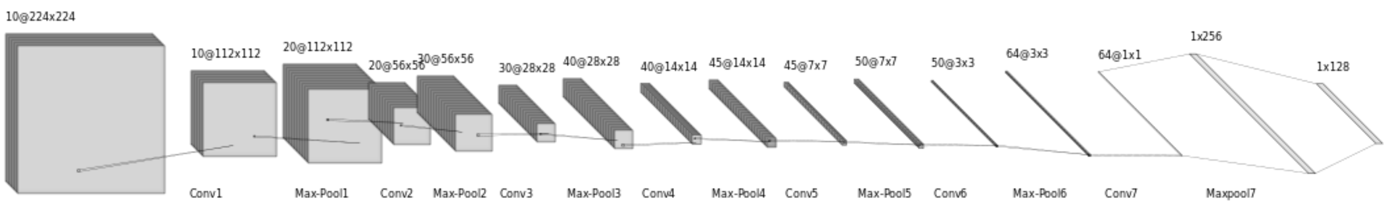

Figure 14: Depiction of the architecture of the CNN that was implemented.

3.2 Convolutional Neural Network

Another primary aim of this project was to compare the performance of shallow machine learning techniques with a Convolutional Neural Network. We experimented with different architectures of the CNN and settled on a network with 7 fully connected layers with a ReLu activation function augmented by Max Pooling and Dropout layers. We then compared the performance of the shallow machine learning methods with that of the CNN for an increasing number of randomly selected artists.

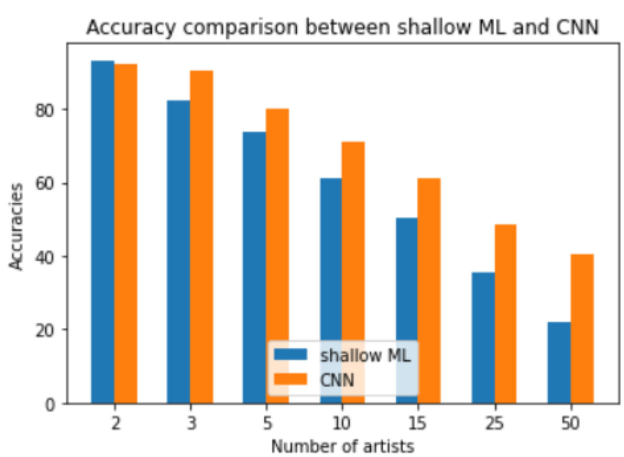

Figure 13: Bar chart depicting the accuracy of the optimal shallow classification method and CNN

4. Conclusion

The primary goal of this project was to apply shallow machine learning algorithms in the real-world classification problem of painter identification. Specifically, we wanted investigate the changes in performance of different shallow machine learning techniques with various feature engineering techniques. We considered three classes of feature engineering techniques and seven different classification methods. In order to reduce computational cost while optimizing the model, we planned to test the model with a smaller subset of 3 artists. The next step was be to investigate the trends in performance when increasing the number of artists. Finally, we wanted to compare the performance of the shallow machine learning techniques with that of a convolutional neural network.

We discovered that the highest accuracy obtained for 3 artists is 90.28%. We also determined the best combination of features for an increasing number of artists was a feature vector containing RGB Histogram, Hu Moments, Haralick Features, and Canny Edge Sum. In this case the accuracy started at about 93% and proceeded to drop to 22%. Here, Random Forest Classifier worked best at first but scaled poorly compared to KNN, SVM and Softmax Regression.

Finally, we contrasted the performance shallow machine learning techniques with that of a CNN. Upon comparison, we were able to ascertain that the performance of the CNN was comparable to that of the shallow machine learning techniques for a lower number of artists but scaled better when the number of artists increased. For 25 artists, using CNN resulted in an accuracy increase of 13%. For all 50 artists, the CNN resulted 40% accuracy, outperforming shallow ML by 18%.

References

[1] D. Keren, "Painter identification using local features and naive bayes," in Pattern Recognition, 2002. Proceedings. 16th International Conference on, vol. 2" pp. 474-477,2002.

[2] E. Cetinic, S. Grgic "Automated painter recognition based on image feature extraction", Proceedings of the 55th International Symposium ELMAR, pp. 19-22. IEEE, 2013.

[3] N. van Noord, E. Hendriks, and E. Postma. 'Toward Discovery of the Artist's Style: Learning to recognize artists by their artworks." Signal Processing Magazine, IEEE 32, no. 4, pp. 46-54., 2015.

[4] Kaggle.com. 2021. *Best Artworks of All Time*. [online] Available at: <https://www.kaggle.com/ikarus777/best-artworks-of-all-time> [Accessed 13 December 2021].

[5] T. Ojala, M. Pietikainen and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987, July 2002, doi: 10.1109/TPAMI.2002.1017623.

[6] T. Ahonen, A. Hadid and M. Pietikainen. "Face recognition with local binary patterns", in Proc. Eighth European Conf. Computer Vision, Prague, Czech Republic, May 11-14, 2004, pp. 469-481, 2004. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.214.6851 DOI:10.1007/978-3-540-24670-1_36

[7] Zhihu Huang and Jinsong Leng, "Analysis of Hu's moment invariants on image scaling and rotation," 2010 2nd International Conference on Computer Engineering and Technology, 2010, pp. V7-476-V7-480, doi: 10.1109/ICCET.2010.5485542.

[8] https://en.wikipedia.org/wiki/Haar-like_feature

[9] R. M. Haralick, K. Shanmugam and I. Dinstein, "Textural Features for Image Classification," in IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-3, no. 6, pp. 610-621, Nov. 1973, doi: 10.1109/TSMC.1973.4309314.

[10] George Stockman and Linda G. Shapiro. 2001. Computer Vision (1st. ed.). Prentice Hall PTR, USA.

[11] Sural, Shamik & Ayyasamy, Vadivel & Majumdar, Arun. (2005). Histogram Generation from the HSV Color Space.

[12] Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60, 91–110 (2004). https://doi.org/10.1023/B:VISI.0000029664.99615.94

[13] A. Singh, "Feature descriptor: Hog descriptor tutorial," Analytics Vidhya, 10-May-2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/. [Accessed: 13-Dec-2021].

[14] J. Liang, "Justin Liang," Canny Edge Detector. [Online] .Available:https://justin-liang.com/tutorials/canny/. [Accessed: 13-Dec-2021].

[15] Zhao W, Zhou D, Qiu X, Jiang W (2021) Compare the performance of the models in art classification. PLoS ONE 16(3): e0248414. https://doi.org/10.1371/journal.pone.0248414