ARTIST IDENTIFICATION

ECE 6254 PROJECT REPORT

Connor Frost* **Kaitlin Burke*** **Kennedy Lee*** **Sowmya Venkatachari***
Venkata Subramanian Srinivasan*

*denotes equal contribution

1 PROJECT SUMMARY

The objective of this project is to develop an algorithm that can identify the artist of a given painting or drawing. Such an algorithm could be used to help attribute disputed works or detect forgeries. Confirming the origin of a painting can be painstaking work requiring the opinion of multiple experts. For example, a painting believed to be created by Van Gogh surfaced in the 90s and was repeatedly dismissed by experts. It took a two year investigation by the Van Gogh Museum starting in 2011 to confirm its authenticity [1]. Similarly, an art dealer purchased a painting for $185,000 in 2016 that he claims is a Rembrandt, but it has yet to be widely accepted as legitimate even four years later [2]. The FBI Art Crime Team, established in 2004, has recovered more than $800 million worth of forged paintings since its creation [3; 4]. Paintings by such highly renowned artists can sell for millions of dollars, and investigations take years.

The technical analysis involved in detecting an art forgery is elaborate. The painting under suspicion is compared with a genuine artwork by the same artist or in the same period. The materials, pigments, brushstrokes, style, the age deterioration in the paintings, etc. are compared by a team of art experts. UV rays and X-rays are commonly used to examine the structure of the paintings or additions to it. Chemical analyses of the paintings often require pieces of the painting to be removed [5]. An accurate machine learning algorithm could be a useful tool to help speed such investigations and to help avoid undesirable damage to the paintings.

We implemented a variety of techniques for feature extraction, dimensionality reduction, and classification of paintings. Our report compares the accuracy of the results using different combinations of these techniques. We used a dataset of attributed artworks available on Kaggle to train our algorithms.

2 PROJECT DESCRIPTION

2.1 INTRODUCTION

The motivation of this project is to develop a model to identify the artist given an image of a painting. We performed an extensive analysis on a broad range of statistical machine learning algorithms in identifying the creator of a work of art. An algorithm with a high success rate could be used to look for fraudulent artwork, to confirm the artist of long-lost works of art, or to attribute disputed artworks.

2.2 BACKGROUND

2.2.1 DATASETS

We used the Kaggle dataset "Best Artworks of All Time" as our source for images [6]. The resized images in this dataset are approximately 1K in resolution. Working with higher resolutions would require more computing power and time than we had available for this project. The dataset includes paintings from 50 of history's best artists, but we focused on the four Baroque artists in the dataset, including Rembrandt. We reserved 20% of each artist's works from the original dataset to be used as testing images, while the rest were used for training.

2.2.2 FEATURE EXTRACTION

For more basic classification techniques, such as Linear Regression or SVM, in addition to using the image vector, various image features can be identified and extracted from the original image vector which can help in improving the efficiency or accuracy. The methods we selected were based on techniques used in pre-existing artist identification publications [7] [8] [9]. All extraction methods used the OpenCV python package, with the exception of Local Binary Patterns, which used skimage.

- **Flattened Image vector** - Obtained by resizing the image matrix (MxN) to (1xMN)
- **Scale-Invariant Feature Transform (SIFT)** - SIFT extracts specific points from an image to construct a descriptor. To keep a consistent feature vector length across images, SIFTs were limited to 50 keypoints per image.

- **Histogram of Oriented Gradients (HOG)** - HOGs are vectors made of bins of intensity gradients corresponding to the gradient direction. The vector is defined for various blocks in the image, and those blocks are concatenated together to form the final vector. For HOG, images were resized to 200x200 before extraction and a 10 pixel cell size was used to keep vector lengths low.
- **Color Histograms** - Color Histograms are simple vectors that describe how much each color in the RGB color space is represented at various intensities.
- **Hu Moments** - Hu moments are calculations based an image's image moments that are invariant to various image transformations.
- **Local Binary Patterns** - LBPs are measures of the relationship of each pixel's intensity to its neighbors. In order to limit and standardize the size of these features, images were re-scaled before patterns were calculated.
- For convolutional neural networks, feature extraction is done by the network itself. This means the experimenter no longer has to manually select features.

2.2.3 DIMENSIONALITY REDUCTION

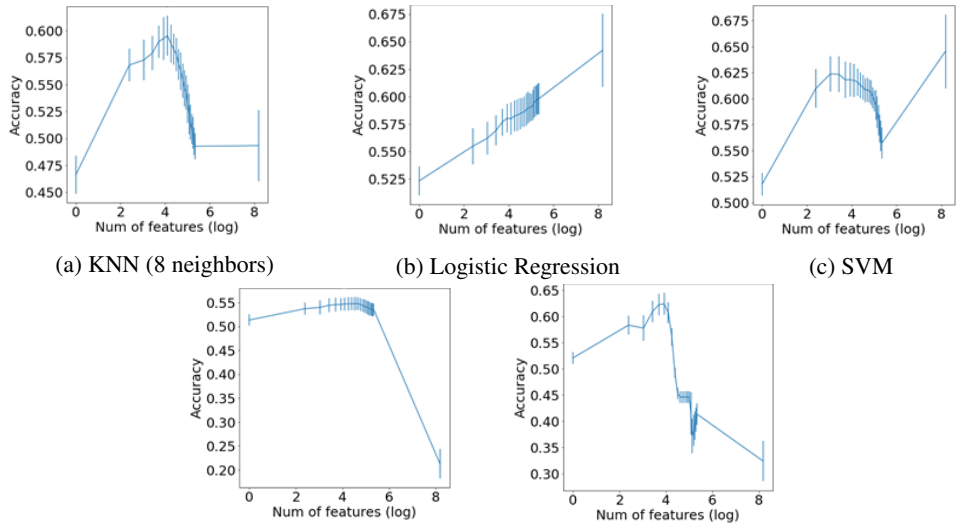
A machine learning model can potentially benefit from a large amount of training data and features. However, when the number of features is very high compared to the number of data samples, different combinations of features might not be well represented in the dataset and a machine learning model might overfit to the training dataset (curse of dimensionality). In our project, we explored three different algorithms for dimensionality reduction which are listed below:

- **k-PCA** - Principal Component Analysis (PCA) is a dimensionality reduction technique that projects the input data to a lower dimensional space by finding principal components of the data that are orthogonal to each other. Kernel-PCA (k-PCA) is the extension of PCA to datasets that are not linearly separable.
- **LDA** - For labelled datasets, Linear Discriminant Analysis (LDA) is a technique commonly used to find a projection of the dataset to a lower-dimensional feature space where the separation between different classes is maximized.
- **t-SNE** - t-distributed Stochastic Neighborhood Embedding (t-SNE) is a probabilistic dimensionality reduction technique that preserves the local structure of the data while also segregating it into well-defined clusters. t-SNE computes an embedding in a lower dimensional space such that the difference between the probabilistic similarity measures in the two different spaces is as small as possible [10].

2.2.4 CLASSIFICATION TECHNIQUES

The objective of the classification algorithm is to assign an input image one label from a given set of categories. We implemented the following classification techniques on our extracted features:

- **Nearest Neighbor (Non-linear)** - Has a small number of hyperparameters to tune (one). Requires balanced and scaled data to reduce biasing. Outliers and high dimensionality in the dataset can cause biasing.
- **Random Forest (Non-linear)** - Uses ensemble learning (multiple trees) to reduce overfitting. Does not require feature scaling and is relatively robust to outliers, noise and small changes in the dataset.
- **Logistic Regression (Non-linear)** - Performs best with data that is linearly separable. Assumes linearity between the features and classes, limiting classification capability. Lacks resistance to overfitting with high dimensional datasets.
- **Naïve Bayes (Linear)** - Only requires one pass of the dataset to calculate the posterior class probabilities. Has low prediction accuracy due to assumption of independent predictors. A class must be in observed in the dataset for a probability to be assigned to it.
- **SVM (Linear)** - Uses hyperplanes to separate classes. High-dimensional spaces are beneficial, allowing for better separation of the classes. Suffers in the case of overlapping classes where data is not linearly separable.



(a) KNN (8 neighbors) (b) Logistic Regression (c) SVM

(d) Gaussian Process (e) QDA

Figure 1: Cross Validation Accuracy vs Number of Features after performing kPCA

- **Neural Network [Multi-layer Perceptron] (Non-linear)** - Requires a large dataset for proper fitting of the model. Does not have great feature extraction capability compared to the CNN.
- **Gaussian Process (Non-linear)** - Generates a distribution for the prediction value. Prior model shape specification can be added through the kernel function choice. Has good results without cross validation through maximizing the marginal likelihood.
- **Decision Tree (Non-linear)** - Does not require data to be normalized or scaled but is sensitive to small changes in the dataset. A single decision tree is relatively weak for learning (best applied with ensemble learning).
- **QDA (Non-linear)** - Assumes that the observations come from a multivariate distribution. Allows for classes to have different covariance matrices, unlike LDA. A small group sample size may cause the estimation of the covariance matrix for the group to not fully represent the group's population covariance, leading to classification inaccuracies.
- **AdaBoost (Non-linear)** - It is an ensemble method like Random Forest. Relatively resistant to overfitting through using boosting by assigning more weight to observations that were classified incorrectly.
- **CNN (Non-linear)** - Requires a large dataset for proper fitting of the model. Has great feature extraction capability, hence it can learn from images. Requires extensive parameter tuning and is computationally expensive and slow to train.

2.2.5 Parameter Tuning

We used randomized search cross validation to select different parameters for the classifiers that affect the trade-off between bias and variance and determine if the model underfits or overfits the dataset. The parameters that we tuned include the number of nearest neighbors for kNN, the maximum depth for decision tree and random forest, the C and gamma values for SVM, the learning rate for AdaBoost, and the number of layers and neurons for MLP.

3 Results and Discussion

3.1 Dimensionality Reduction

The dimensions of extracted features such as LBPs (dimension: 5600) and HoGs (dimension: 3600) were much higher compared to the number of images in the training dataset. We observed the ef-

Table 1: Model comparison based on accuracy, weighted precision, weighted recall and weighted F1 score

Classifier	**Features Used**	**Accuracy**	**Precision** (Weighted)	**Recall** (Weighted)	**F1 Score** (Weighted)
Nearest Neighbor	HoG	0.45106383	0.6188	0.4511	0.5079
Random Forest	Hu Moments	0.45957447	0.7998	0.566	0.6627
Logistic Regression	Hu Moments	0.47234043	0.7740	0.5574	0.6452
Naive Bayes	Color Histogram	0.53191489	0.6554	**0.6596**	0.6567
SVM	Local Binary Patterns	0.53617021	**0.9602**	0.459	0.6102
Neural Network	Hu Moments	0.55744681	0.5980	0.4468	0.4972
Gaussian Process	Color Histogram	0.57021277	0.7122	0.6553	**0.6735**
Decision Tree	Hu Moments	0.58297872	0.5435	0.4596	0.4895
QDA	Color Histogram	0.65531915	0.5858	0.583	0.5828
AdaBoost	HoG	0.65957447	0.8019	0.5362	0.6133
CNN	-	**0.7009**	-	-	-

fects of dimensionality reduction (kernel PCA) on the classification accuracy (obtained using cross validation) for different feature representations and classifiers. The results for HoGs are shown here [Fig.1]. We observed that for certain classifiers such as KNN, Gaussian Process and QDA, dimensionality reduction significantly improved performance for most feature representations, whereas for certain other classifiers such as SVMs and logistic regression, dimensionality reduction reduced performance for most feature representations.

3.2 Analysis based on Accuracy

It is evident from the low dimensional tSNE representation of the image vectors [Fig.6a] that the classification problem requires carefully selected feature-algorithm combination. On running the grid search for the best classification algorithm-features combination, it is interesting to note that there is no popular choice of features representation among the classifiers [Table.1].

(a) Image Vector (tSNE)

(b) Training time comparison

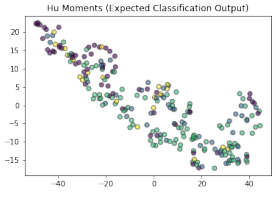
Figure 2: 2D tSNE projection of the resized image vectors (30x30) (left); Comparison of model training time (right)

3.2.1 Hu moments based classifiers

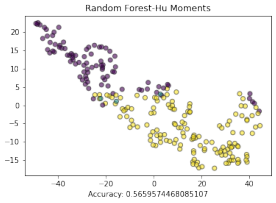
The algorithms - random forest, logistic regression, neural network and decision tree obtains maximum accuracy on the hu moments feature representation [Fig.3]. It is interesting to note that even in the presence of high dimensional features like SIFT (dimension: 2560) or LBP (dimension: 5600), reasonably complex algorithms like neural network and random forest relies on hu moments.

3.2.2 Color Histogram based classifiers

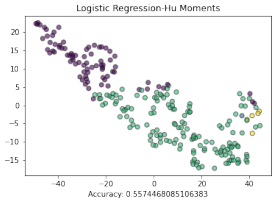
The tSNE plots for the histogram based classifier [Fig.4] shows that it provides a reasonable feature representation for the images as there exist a noticeable separation in the low dimension representation of the features. This is reflected among the classifiers since the accuracy of the histogram based classifiers appears to be consistent with relatively unconstrained hyperparameter choices. The algorithms - naive bayes, gaussian process, QDA obtains maximum accuracy for the features based on the color histogram.



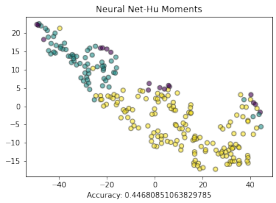
(a) Hu Moments (tSNE)



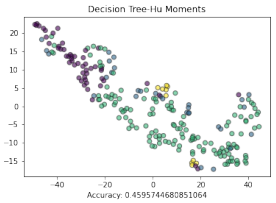
(b) Random Forest



(c) Logistic Regression

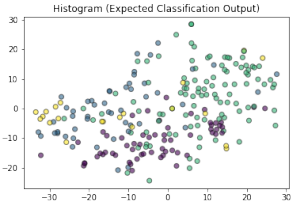


(d) Neural Network

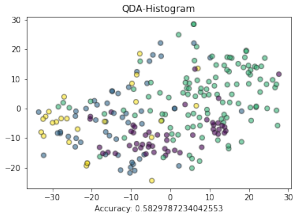


(e) Decision Tree

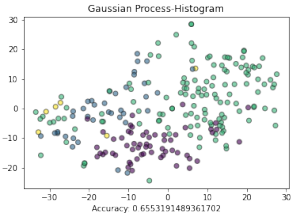
Figure 3: Hu Moments based classification



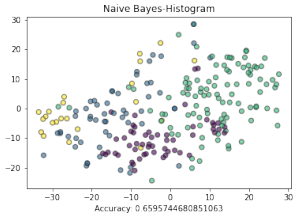
(a) Color Histogram (tSNE)



(b) QDA



(c) Gaussian Process



(d) Naive Bayes

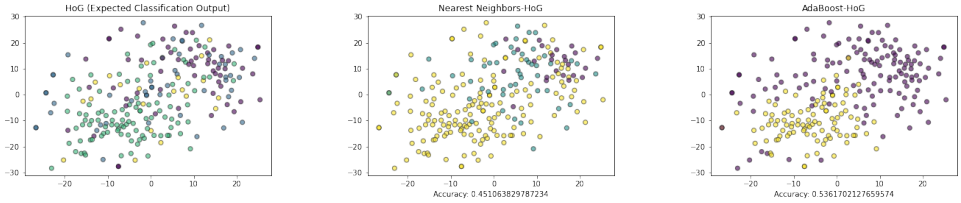
Figure 4: Color Histogram based classification

3.2.3 OTHER CLASSIFIERS

None of the classifiers except CNN utilizes the original image vector or SIFT as the input features. But the hyperparameter choice suffers from the methodology of selection based on grid search based on accuracy. Since it is noticeable from the tSNE representation [Fig.5 Fig.6] that SVM achieves high accuracy by predicting the most likely class, hence poorly performs in other metrics such as recall and f1 score.

3.3 ANALYSIS BASED ON F1 SCORE

It is noteworthy to observe that most classifier achieves high accuracy by ignoring one or more classes as probable output for the images [Table.2]. Only 4 of the 10 classifiers predicts across all the artists and three of them uses color histogram as their features.



(a) HoG (tSNE) (b) Nearest Neighbors (c) AdaBoost

Figure 5: HoG based classification

(a) Local Binary Pattern (tSNE) (b) SVM

Figure 6: Local Binary Pattern based classification

Table 2: F1 score per class comparison

Classifier	Peter	Diego	Rembrandt	Caravaggio
SVM	0.11428571	0.	0.63221884	0.
Random Forest	0.57553957	0.	0.67460317	0.
Neural Net	0.56451613	0.	0.68382353	0.
AdaBoost	0.52791878	0.	0.73267327	0.
Nearest Neighbors	0.26190476	0.22	0.63396226	0.
Logistic Regression	0.62857143	0.03921569	0.6798419	0.
Gaussian Process	0.63865546	0.63636364	0.72033898	0.22222222
Decision Tree	0.46551724	0.15789474	0.60728745	0.25806452
Naive Bayes	0.69565217	0.54716981	0.75728155	0.37209302
QDA	0.53211009	0.47058824	0.72300469	0.30434783

4 Conclusion and Future Work

On careful comparison of multiple classification algorithms, we observe that there is not any specific algorithm achieving the best performance across all the metrics.

We can see that histograms as the most effective feature representation as the classifiers based on them tends to have a non-zero most probable prediction across all 4 classes of artists. On ranking the algorithms based on the metrics, we find that SVM, naive bayes and gaussian process tends to be the best classifier in terms of precision, recall and F1 score respectively.

Since the hyperparameter choice was completely based on grid search, moving forward we can perform a grid/randomized search based on different metrics to observe if the limitation is in the model or the algorithm. Currently there isn't any correlation between the complexity of the model to the classification although the CNN model (53,325,508 trainable prarmeters) outperforms all the other algorithms. One constraint would be optimizing the algorithms to work on a limited set of training data.

We expect the future research would be towards an ensemble classifier with enhanced accuracy across all the classes. Also, it would be interesting to analyze the performance with pretrained CNN feature extractors such as VGGnet[11] and Resnet[12] and larger CNN models with batchnorm and dropout layers.

REFERENCES

[1] *Sunset at Montmajour, 1888 - by Vincent van Gogh*, 2009. [Online]. Available: https://www.vincentvangogh.org/sunset-at-montmajour.jsp. [Accessed: 03-Oct-2020].

[2] B. Katz, "A Dutch Art Dealer Says He Discovered a New Rembrandt," *Smithsonian.com*, 17-May-2018. [Online]. Available: https://www.smithsonianmag.com/smart-news/dutch-art-dealer-says-he-discovered-new-rembrandt-180969117/. [Accessed: 03-Oct-2020].

[3] "Art Theft," *FBI*, 03-May-2016. [Online]. Available: https://www.fbi.gov/investigate/violent-crime/art-theft. [Accessed: 03-Oct-2020].

[4] E. Zachos, "How More Than Half the Art in This French Museum Was Forged," *National Geographic*, 01-May-2018. [Online]. Available: https://www.nationalgeographic.com/news/2018/05/fake-art-france-culture-spd/close. [Accessed: 03-Oct-2020].

[5] Bonner, Gerald, and Joseph Veach Noble. "Detection of Forgeries in the Visual Arts." Encyclopædia Britannica, Encyclopædia Britannica, Inc., 22 May 2020, www.britannica.com/art/forgery-art/Detection-of-forgeries-in-the-visual-arts.

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

[7] N. Viswanathan, "Artist Identification with Convolutional Neural Networks," 2017.

[8] J. Chen and A. Deng, "Comparison of Machine Learning Techniques for Artist Identification," 2018.

[9] A. Blessing and K. Wen, "Using Machine Learning for Identification of Art Paintings," 2010.

[10] Hinton, Geoffrey E., and Sam T. Roweis. "Stochastic neighbor embedding." Advances in neural information processing systems. 2003.

[11] Karen Simonyan, Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2015

[12] K. He, X. Zhang, S. Ren, J. Sun,"Deep Residual Learning for Image Recognition," 2015

[13] https://www.kaggle.com/c/painter-by-numbers/data

[14] E. Cetinic, T. Linic, and S. Grgic, "Fine-tuning Convolutional Neural Networks for Fine Art Classification," *Expert Systems with Applications*, vol. 114, pp. 107–188, Dec. 2018.

[15] D. Wynen, C. Schmid and J. Mairal, "Unsupervised Learning of Artistic Styles withArchetypal Style Analysis," 2018.

[16] E. Goukassian, "How Computers Can Help Art Historians Identify Disputed Rembrandts," *Artsy*, 04-Apr-2019. [Online]. Available: https://www.artsy.net/article/artsy-editorial-computer-authenticate-disputed-artworks. [Accessed: 02-Oct-2020].