CS 639 Final Project Report

An Automatic Method of Identifying Vincent Van Gogh's Paintings

Anze Xie (Andy)

December 8, 2020

Problem Overview and Motivation

The authorship of a piece of fine-art painting is very important since it can determine the art value, market value and historic value of the painting[2]. The common method employed by art specialists to identify the author of a painting is to scan the painting with UV light or X-rays[3]. However, these methods are invasive to the paintings and may cause some potential negative influence to the paintings. With the latest computer vision technologies and machine learning algorithms, it is possible to develop an automatic and non-invasive method to identify the authorship of paintings. Also, such method may help to improve the efficiency of data management for online databases. With the rapid development of information and communication technology, a significant number of fine-art paintings are upload to online datasets. An automatic method that can classify these paintings into different genres and authorships can make the process of managing these databases easier.

The main focus of this project is to implement an automatic method to identify Vincent van Gogh's paintings that was proposed by Folego et al.[1].

Current State-of-the-Art

The researches in fine-art classification have been lasting for years. Many researches have been working on methods for classifying the authorship and genre of fine-art paintings. For identifying the author of fine-art paintings, many machine learning algorithms are applied. Liao et al. proposed a cluster multiple kernel learning algorithm based on the oil paintings from the aspects of color, texture, and spatial layout[6]. Sandoval et al. presented a two-stage learning approach to achieve the goal of classifying fine-art paintings7]. They applied the six different pre-trained CNNs (AlexNet, VGG-16, VGG-19, GoogLeNet, ResNet-50, and Inceptionv3) as the first stage of the classification[7]. After this, a shallow neural network is used to rectify the mistakes made in the first stage and do additional classifications[7]. For the specific purpose of classifying van Gogh's paintings, Folego et al. used the pretrained VGG-19 to extract features from raw sub-patches of the paintings directly[1]. Then fitted a support vector machine to get the final response[1]. There are also some researches focused on specific features of Vincent Van Gogh's paintings, such as brush strokes. Johnson et al. employed wavelets and hidden Markov models to extract the individual brush stroke information for authorship identification[4]. Li et al. proposed a novel method of extracting the brush strokes in Vincent Van Gogh's paintings by integrating edge detection and image segmentation[5]. They then did statistical analysis on the individual and interactive features of the brush strokes to identify Vincent Van Gogh's paintings of different periods [5]. Many researchers also proposed methods that combines the general machine learning approaches and specific feature extraction techniques. Zhong et al. presented a two-channel method which combines the RGB channel and the brush stroke information channel[10]. They used the gray-level co-occurrence matrix to represent brush strokes which is a novel way of encoding the brush stroke information for deep learning [10].

My approach and implementation

This project mainly implemented an automatic method to identify Vincent van Gogh's paintings that was proposed by Folego et al.[1]. In addition to Folego et al.'s method, I also added some changes to the method [1]. The following list is a pipeline for the method used in this project. The image transformation is a part added by me. I also added some attributes when fitting the support vector classifier model. Although most of the method I used in the project is based on Folego et al.'s method, I implemented all the codes from scratch myself [1].

1. Patch extraction
2. Image transformation
3. Feature extraction
4. Fit support vector classifier model
5. Compute score
6. Furse score and output final response

Since training a CNN from scratch needs a large amount of data and is very time consuming, I followed Folego et al.'s method by using a pre-trained CNN[1]. According to Folego et al., the pre-trained CNN this method used is VGG-19 [1]. Since this model is trained with millions of images, it can extract complex visual patterns, as mentioned by Folego et al. [1]. The input size of VGG-19 is 224*224*3 [9]. Thus, each patch input needs to have the same size. I first cropped the paintings into a size that is a multiply of the input size of VGG-19. In the process of cropping the paintings, the peripheral part of the paintings are discarded. Then I break the cropped painting into patches.

The second step of the method implemented for this project is image transformation. When searching for paintings of other artists to build a data set to test if this method also works for identifying other artists' paintings, I found that many downloadable paintings' sizes are very small and also the number of available paintings is limited. Therefore, I decided to flip the patches extracted in step 1 horizontally to increase the size and variety of the training set. I also added some random noise to the patches to prevent the final model become overfitted.

As Folego et al. did in their method, I put the patches into VGG-19 and extracted the features from the third-to-last layer as well [1]. There are in total 4096 features for each patch [10]. Since I am a beginner to machine learning, I think extracting the features from the third-to-last layer, fitting a SVM model and trying different fuse method makes more sense to me. Also, SVM is an efficient and effective model for high-dimensional data.

After extracting the features from VGG-19 for each patch, I fitted a support vector classifier model using these features. This is also a step mentioned in Folego et al.'s paper [1]. Different from what they mentioned in their paper, I also balanced the classes when fitting the support vector classifier model since the number of positive and negative class samples are not balanced in the training set. As mentioned in the document of Sklearn, the weight of the two classes is adjusted inversely proportional to class frequencies in the input data. As Folego et al. did in their method, I compute the score for each patch in the test set as the signed distance of the patch to the separating hyperplane

[1]. The sign of the score indicates what class is this patch classified to. The magnitude of the score can be interpreted as the amount confidence of the classification result.

Five fusing methods are used in the project. They are also mentioned by Folego et al.[1]. These five fusing methods used in this method is listed below.

- Max number of votes(number of patches with positive/negative distance)

  The max number of votes method counts the number of patches classified as positive and negative respectively. Then determine the final response using the class with higher votes.

- Mean patch distance to separating hyperplane

  The mean patch distance is the mean of the signed distance of all the patches. The final response is determined by the sign of the mean patch distance. This method considers the influence of the numbers of patches in the 2 classes.

- Total patch distance to separating hyperplane

  The total patch distance is the sum of the signed distance of all the patches. The final response is determined by the sign of the total patch distance. This method ignores the effect of the numbers of patches in the 2 classes. This method may be affected by outliers.

- The maximum patch distance to separating hyperplane

  The maximum patch distance is the patch distance with the greatest magnitude. The final response is determined by the sign of the farthest patch distance. The intuition of this method is to determine the final response based on the class of the patch with the most confidence. However, this method may be vulnerable to outliers.

- The median of patch distance to separating hyperplane

  The median patch distance is the median of all the patch distances. The final response is determined by the sing of the median patch distance. This method is more resistant to the outliers.

Dataset description

There are two datasets used in this project. One dataset is used to training the classifier model for identifying van Gogh's paintings and another dataset is used for Pablo Picasso's paintings. I used the same dataset used by Folego et al. [1]. when training the model for identifying van Gogh's paintings. According to Folego et al., the data set contains 333 RGB paintings with similar density [1]. By the definition mentioned by Folego et al., density here means the number of pixels in per inch [1]. In total there are 124 images of van Gogh's paintings and 207 images for non-van Gogh's paintings [1]. The non-van Gogh paintings are collected from categories such as Impressionism, Post-Impressionism, Neo-Impressionism, and Expressionism so that they would match the context for this method, according to Folego et al.[1]. Due to the limited computing power I have, I made a subset of Folego et al.'s dataset as my dataset for this project [1]. In total, there are 87 images in my dataset. After image transforming and patch extraction, I obtained a dataset showed in table 1 below.

Class	Training paintings	Training patches	Test paintings	Test patches
van Gogh	23	78754	9	2648
non-van Gogh	42	17222	11	3114
Total	67	25076	20	5762

Table 1.

I collected the dataset for training the model identifying Pablo Picasso's paintings myself from various sources from the Internet. There are 147 non-Picasso paintings and 76 Picasso's paintings. The dataset status after image transformation and patch extraction is shown in table 2 below.

Class	Training paintings	Training patches	Test paintings	Test patches
Pablo Picasso	52	1683	24	263
non-Pablo Picasso	154	4779	43	343
Total	156	6462	67	606

Table 2.

Since the image size I found for this dataset is smaller, the patch extracted in this dataset is less. This is also a reason that motivates me to add an image transformation step in the pipeline mentioned before. As Folego et al. chose the non-van Gogh paintings from the categories that would match the context of this method, I also searched the non-Pablo Picasso from a bunch of artists who belong to these categories [1].

Results

After applying my method to the dataset of van Gogh's paintings, I got the following result.

	Correct prediction	False positive	False negative	Correctness rate	F1-scroe
Max vote	16	4	0	80%	0.889
Mean distance	17	3	0	85%	0.919
Sum distance	17	3	0	85%	0.919
Max distance	16	4	0	80%	0.889
Median distance	16	4	0	80%	0.889

Table 3.

Here the F1-score is computed using the following formula:

$$F_1 = \frac{TP}{TP + 0.5\,(FP + FN)}$$

TP = number of true positive

FP = number of false positive

FN = number of false negative

As shown in table 3 above, the method of mean distance and sum distance give the best output. According to Folego et al. the method of sum distance gives the best output with a

F1- score of 0.923 in their implementation [1]. The output of my implementation basically matches the output of theirs [1].

To further confirm the correctness of using mean distance and sum distance as the fusing method, I did a two-sample t-test for the mean of the patch distances of van Gogh's paintings and non-van Gogh paintings.

Null hypothesis:

$$Mean_{van\ Gogh} = Mean_{non-van\ Gogh}$$

Alternative hypothesis:

$$Mean_{van\ Gogh} \neq Mean_{non-van\ Gogh}$$

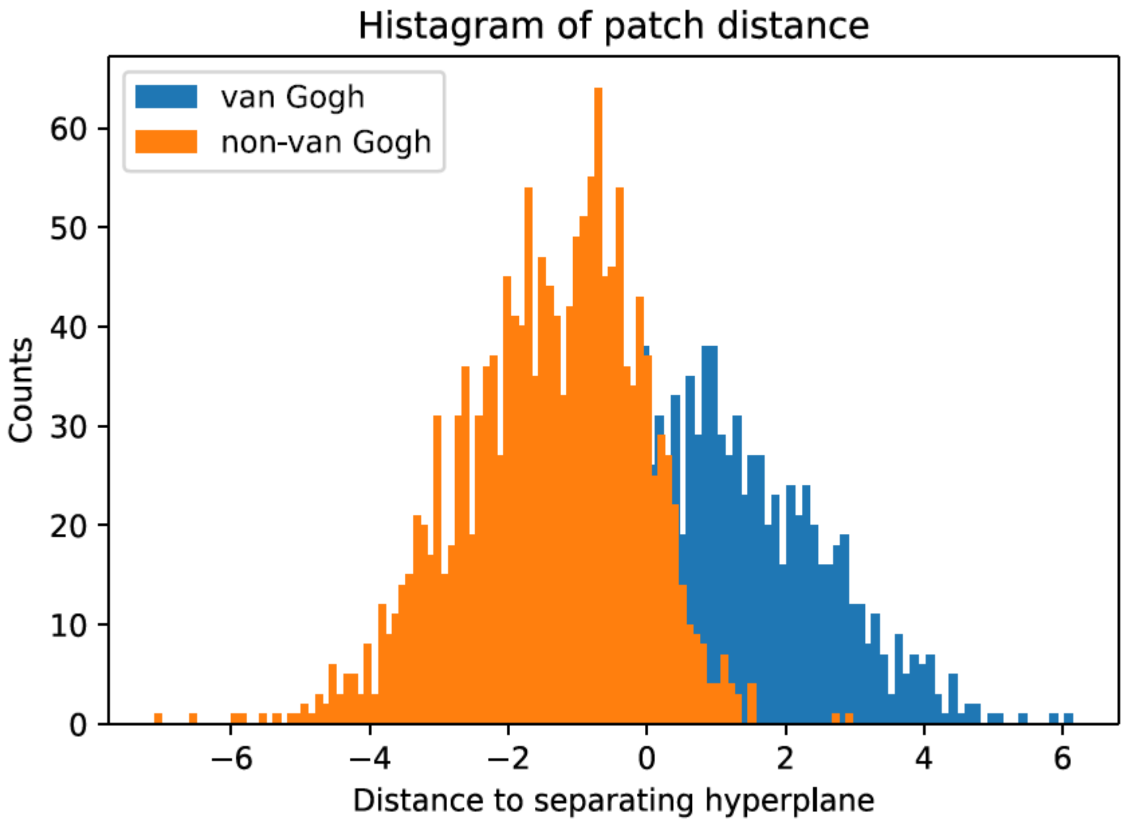

Figure 1

The mean and variance of van Gogh patch distance is 0.4418 and 2.9938 respectively. The mean and variance of non-van Gogh patch distance is -1.4115 and 1.6251 respectively. The t-statistic I obtained from the two-sample t-test is 32.238 and the p-value is almost 0. There is sufficient evidence for us to drop the null hypothesis. In this case, there is significant evidence showing that the distribution of the patch distance of van Gogh's paintings and non-van Gogh paintings are having very different mean values. This also shows that using the mean patch distance as the fusing method is reasonable.

The same method is applied to the dataset of Pablo Picasso's paintings. Table 4 below shows the results.

	Correct prediction	False positive	False negative	Correctness rate	F1-scroe
Max vote	49	10	8	73.1%	0.845
Mean distance	49	9	9	73.1%	0.845
Sum distance	49	9	9	73.1%	0.845
Max distance	43	13	11	64.2%	0.782
Median distance	49	9	9	73.1%	0.845

Table 4.

The correctness rate and F1-score of this method on identifying Picasso's paintings is lower than the its performance on the dataset for van Gogh's paintings. One possible reason could be the insufficiency of samples. Although there are more Picasso's paintings than van Gogh's paintings, the sizes of Picasso's paintings are smaller than van

Gogh’s paintings. Therefore, the number of patches extracted from Picasso’s paintings are a lot less than this number I get from van Gogh’s dataset.

Null hypothesis:

$$Mean_{Picasso} = Mean_{non-Picasso}$$

Alternative hypothesis:

$$Mean_{Picasso} \neq Mean_{non-Picasso}$$

A similar two-sample t-test is conducted on the patch distances here. The mean and variance of Picasso patch distance is 1.1164 and 10.8704 respectively. The mean and variance of non-Picasso patch distance is -2.3365 and 12.7982 respectively. The t-statistic and p-value for this test is 12.3125 and 0 respectively. There is sufficient evidence for us to drop the null hypothesis. In this case, there is significant difference between the mean of the 2 patch distance distributions. This test result matches the correctness rate and F1-score of mean distance in table 4 above.

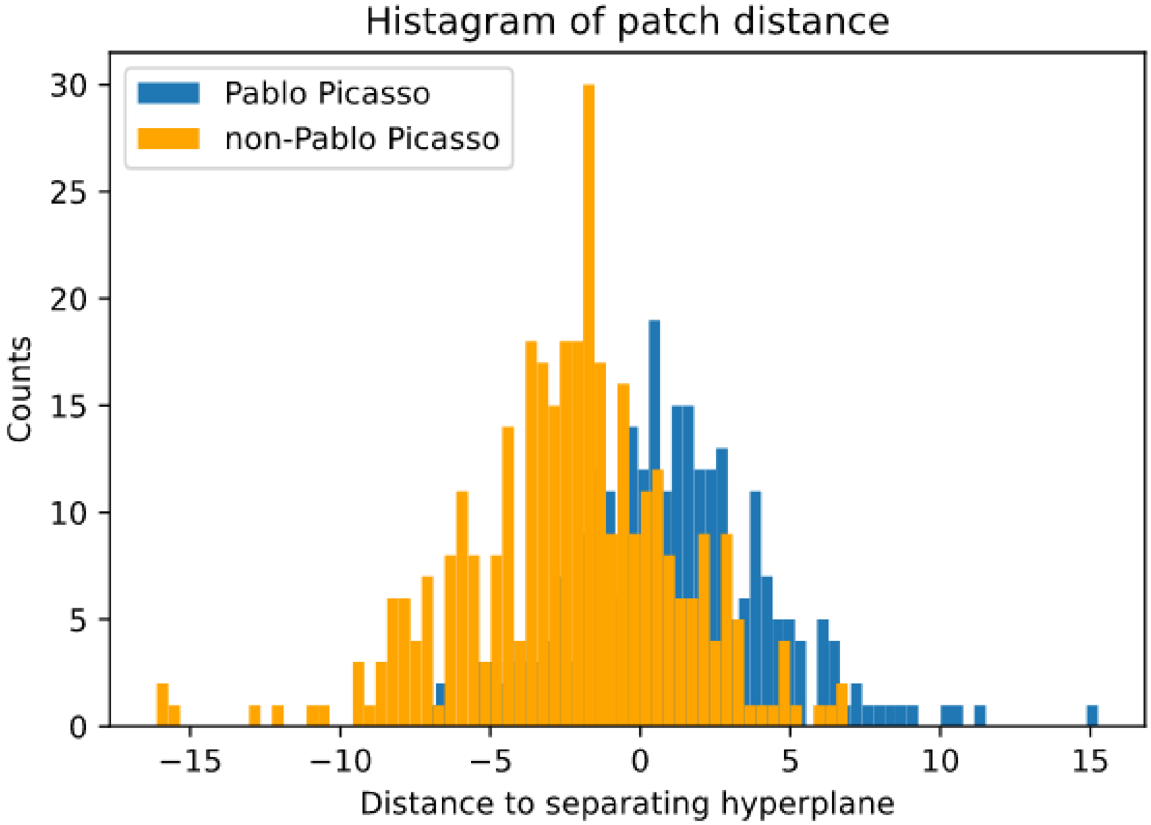

Figure 2

Future work

In the future, other classifiers could be implement to classify the features extracted from CNN. What's more, multi-class classification methods could be developed in similar ways to work on more complicated datasets. Other features of the fine-art paintings, such as brush strokes and complementary colors, could be combined with the automatically extracted features to obtain more precise output. Alternatively, more researches can be conducted on developing special CNN structures for fine-art classification.

References

[1]. Folego, G., Gomes, O., & Rocha, A. (2016). From impressionism to expressionism: automatically identifying van Gogh's paintings. *2016 IEEE International Conference on Image Processing (ICIP).* doi.10.1109/ICIP.2016.7532335

[2]. G. E. Newman and P. Bloom, Art and authenticity: the importance of originals in judgments of value. *Journal of Experimental Psychology,* vol. 141, no. 3, pp. 558, 2012.

[3]. J. Ragai, The scientific detection of forgery in paintings. *Proceedings of the American Philosophical Society,* vol. 157, no. 2, pp. 164-175, 2013.

[4]. Johnson, C. R., et al. Image processing for artist identification—computerized analysis of Vincent Van Gogh's painting brushstrokes. *IEEE Signal Process Mag Special Issue Visual Cult Heritage,* 25, 37-48.

[5]. Li, J., L, Y., Hendriks, E., & Wang, J. Z. (2011). Rhythmic brushstrokes distinguish van Gogh from his contemporaries: findings via automated brushstroke extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 34*(6), 1159-1176. Doi:10.1109/TRAMI.2011.203

[6]. Liao, Z., Gao., Zhou, T., Fan, X., Zhang, Y., & Wu, J. (2019). An oil painters recognition method based on cluster multiple kernel learning algorithm. *IEEE Access, 7,* 26842-26854. doi:10.1109/ACCESS.2019.2899389

[7]. Sandoval, C., Rirogova, E., & Lech, M. (2019). Two-stage deep learning approach to the classification of fine-art paintings. *IEEE Access, 7,* 2169-3536. doi:10.1109/ACCESS.2019.2907986

[8]. Sandoval, C., Rirogova, E., & Lech, M. (2019). Two-stage deep learning approach to the classification of fine-art paintings. *IEEE Access, 7,* 2169-3536. doi:10.1109/ACCESS.2019.2907986

[9]. Simonyan, K., Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556.*

[10].Zhong, S., Huang, X., & Xiao, Z. (2019). Fine-art painting classification via two-channel dual path networks. *International Journal of Machine Learning and Cybernetics, 11*(1), 137-152. doi.org/10.1007/s13042-019-00963-0