IDENTIFYING PAINTERS FROM COLOR PROFILES OF SKIN PATCHES IN PAINTING IMAGES

Ivo Widjaja[1], *Wee Kheng Leow*[2], *Fang-Cheng Wu*[3]

[1,2]Dept. of Computer Science, National University of Singapore, Singapore
[3]Institute of Art Studies, National Central University, Taiwan, R.O.C
[1]iscp9104@nus.edu.sg,[2]leowwk@comp.nus.edu.sg, [3]chartres@cc.ncu.edu.tw

ABSTRACT

Research on digital analysis of painting images has received very little attention. The exact nature of scientific methods seems to be antithesis of art. Nevertheless, several papers have proposed methods to bridge this gap and have obtained interesting results. In fact, some art theorists have pointed out the usefulness of specific quantifiable features in the paintings. This paper presents a method for identifying painters using color profiles of skin patches in painting images. Various color models for representing the color profiles were explored. Various implementations of multi-class Support Vector Machine classifiers were compared. We found that a weighted combination of several Directed Acyclic Graph SVMs with Gaussian kernels gives the best classification performance.

1. INTRODUCTION

Art theorists often classify a painting based on a set of attributes called painting style. These attributes include the physical visual attributes in the paintings and the context of the painting — the artist, the creation date and the geographical location. In short, an artistic style is a combination of iconographic, technical and compositional features that give a work its character and allows it to be attributed to a particular school or period [1].

Research on digital analysis of painting images has received very little attention. The exact nature of scientific methods seems to be antithesis of art. Nevertheless, several papers have proposed methods to bridge this gap and have obtained interesting results. In fact, some art theorists themselves have pointed out the usefulness of specific quantifiable features in the paintings. Looking at the paintings systematically, two general types of features can be identified, namely syntactic and semantic features. Syntactic features include color, texture, composition and brush strokes. Semantic features include iconography, theme and subject matter.

Sablatnig [2] has proposed a painting classification method using brush strokes and face detection for paintings of portraits. Unfortunately, only qualitative results are mentioned in the paper. Herik and Postma [3] presented a comprehensive experiment on painting classification using neural network on many features. With a set of 60 paintings of 6 impressionist and neo-impressionist painters, they achieved an accuracy of 0.85 in identifying the painters.

From the point of view of painting art, form and color constitute the main plastic means by which a painter paints. Painters use colors which come principally from mineral and vegetable extracts and manufactured salts. Each color has its own characteristics and properties. What colors are present in a painting depend therefore not only on the colors present in the subject, but also partly on the ability of the painter to realize the full potential of his material [4, p. 67]. Moreover, a painter decides on what colors he wants to use from those available to him. Unconsciously, a painter may have preferences for certain colors; consciously, he may choose colors according to some conventional rules, which the art historian Baxandall calls *values of colors* [5, pp. 81–85].

In addition, Baxandall suggested using relief (*rilievo*) to categorize paintings from the fifteenth century Italy [5, p. 121]. *Rilievo* is described by Alberti as *"the appearance of a form modelled in the round, attained by the skilful and discreet treatment of the tones of its surface: '...light and shade make real things appear to us in relief'"*.

The above considerations motivate us to examine the possibility of identifying the painters based on color relief. Instead of looking at global features of paintings, such as those of Herik and Postma [3], our method examines local features, specifically the color relief of skin patches in painting images. In our method, color relief of skin patches is modelled as color profiles, which are the distributions of colors in the lateral profiles of the skin patches.

2. CLASSIFICATION OF SKIN PACTHES

2.1. Feature Extraction

Our method of extracting color profile features consists of 4 main steps (Fig. 1):

1. **Skin Patches Extraction**
   The relief of skin patches manifests as a transition from shadow to highlights. For consistency, we gather skin patches with incoming light falling on them at approximately 45–60 degrees incident angles. These skin patches are found in the limbs (arms and legs) of human figures in paintings. All patches are aligned by rotating them such that the light-to-dark transition is parallel to the vertical axis of the images Fig. 1(a). The non-skin portions are discarded. After alignment and cleaning, each column of the extracted image corresponds to a profile across the limb.
2. **Profile Curves Extraction**
   Each column of the aligned image is separated into three color channels. All columns of the same color channel are gathered to form the profile curves, one curve for each column (Fig. 1(b)). All values in each channel are normalized to the range of $[0, 1]$.
3. **Length-Normalization of Profile Curves**
   Each profile curve that comes in a different length is normalized to 100 points long (Fig. 1(c)) using cubic spline

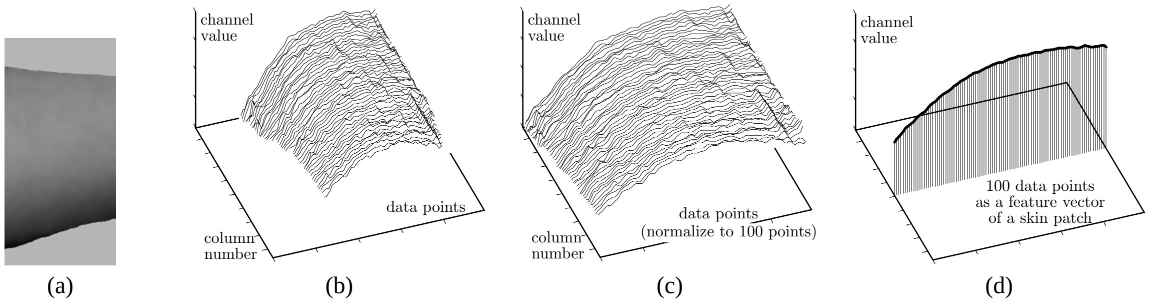

Fig. 1. Color profiles extraction. (a) Aligned image of a skin patch. (b) Extracted profile curves of a color channel, one curve for each column. (c) Normalized profile curves. (d) Averaged profile curve.

interpolation [6]. This is an expansion of the original data points, hence information loss is avoided.

4. **Profile Curves Averaging**
Finally, the normalized profile curves are averaged to form a single averaged profile curve for each color channel (Fig. 1(d)). This profile curve, which contains 100 data points, will be used as the feature vector for SVM classification.

2.2. Multi-Class Support Vector Machine Classifiers

Support Vector Machine (SVM) for binary, i.e., 2-class, pattern classification problems has been well studied. It performs well compared to other learning methods [7]. Chapelle tested SVM for histogram-based image classification and obtained good results [8].

Classification using SVM is achieved by constructing an optimal separating hyperplane between two classes of samples. It maximizes the margin of separation $\|2/\mathbf{w}\|$ and minimizes the upper bound of classification error. In applying SVM to pattern recognition problems, a non-linear mapping is chosen a priori to map input vectors into high-dimensional *feature space* where the optimal hyperplane is constructed. SVM is stated as quadratic optimization problem:

$$L(\mathbf{w}, \xi) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i \qquad (1)$$

with following constraints: $y_i(\mathbf{w}^T\phi(\mathbf{x_i})+b) \geq 1-\xi_i$ and $\xi_i \geq 0$. The optimal hyperplane is given by $\mathbf{w}$. C is the regularization parameter and ϕ is the mapping function whose dot product forms the kernel function.

In real world pattern recognition problems, the number of classes is usually more than two. Several approaches have been developed to tackle multi-class pattern classification problems using SVM. Hsu and Lin [9] presented a comprehensive comparison of many multi-class SVM methods. They can be divided into two main approaches:

- The first approach considers all classes directly at once and solving one optimization problem [10, 11].
- The second approach combines several binary classifiers to form a multi-class classifier. There are several methods in this approach: (i) one-vs-rest SVM, (ii) one-vs-one SVM [12, 13], (iii) SVM with Error Correcting Output Code [14] and (iv) Directed Acyclic Graph SVM [15].

One-vs-rest SVM is the simplest k-class SVM method. It uses k binary SVMs and each SVM is trained with class-i samples as the positive samples and the remainder as the negative samples. The final assigned class is given by the binary SVM with the highest output.

One-vs-one SVM [12, 13] constructs a set of binary SVMs for all possible pairs of classes. Each SVM will be trained with samples from the two corresponding classes. To determine the final assigned class, Knerr [12] suggested combining the SVMs results using an AND gate. Friedman [13] suggested a Max Wins algorithm. Each classifier contributes one vote to its preferred class, and the final class is the one with the highest vote. Kreßel [16] presented excellent results on the Max Wins strategy.

Error Correcting Output Code (ECOC) [14] was proposed as a general method to expand a multi-class classifier into many binary classifiers. Each class is assigned a “codeword” that consists of binary digits. The length of the codeword corresponds to the number of binary classifiers. The codewords from all classes form a matrix that is used to assign the training samples to the classifiers. The final assigned class is the one with the closest codeword to the codeword formed by concatenating the binary classifiers’ outputs.

Platt et al. [15] suggested a method called *Directed Acyclic Graph Support Vector Machine* (DAGSVM) which has a bound on generalization error, a property that other multi-class methods do not have. DAGSVM uses the same training algorithm as the one-versus-one method. However, in the classification part, DAGSVM uses a rooted binary directed acyclic graph which has $k(k-1)/2$ internal nodes and k leaves. To classifiy a sample, the binary decision function in each node is evaluated, starting with the root node and traversing down to a leaf node, which gives the final class.

Weston and Watkins [10] proposed a method that consider all classes at once. They concluded that this method does not outperform one-vs-rest and one-vs-one although there is a slight improvement. Crammer and Singer [11] also used a similar approach and achieved a lower error rate than the one-vs-rest method.

Hsu and Lin [9] presented a test result showing that DAGSVM and one-vs-one SVM are more practical than other methods. Platt [15] presented a result that suggests that DAGSVM performs slightly better or at least the same compared to one-vs-rest and one-vs-one.

3. EXPERIMENTS AND RESULTS

3.1. Painting Data Set

The painting images are taken from *Artchive* (`http://www.artchive.com`) and *Web Gallery of Art*

Table 1. Single-channel classification results. **Bold face** represents the best performance for a channel among the 3 SVM methods.

Channel	R	G	B	H	S	V	H	S	I	H	S	L	L^*	a^*	b^*
DAG	**0.57**	**0.67**	**0.62**	**0.74**	**0.61**	**0.57**	**0.74**	**0.62**	**0.67**	**0.74**	**0.61**	**0.68**	**0.63**	**0.75**	0.62
1-vs-1	0.56	0.64	0.56	0.70	0.61	0.56	0.71	0.59	0.58	0.70	0.55	0.60	0.56	0.73	**0.63**
ECOC	0.42	0.49	0.54	0.64	0.38	0.43	0.68	0.58	0.60	0.64	0.53	0.61	0.41	0.75	0.46

(`http://gallery.euroweb.hu`) web sites. These images are well-prepared and the quality is quite consistent and reasonable compared to other art web sites. By using these images, we can minimize the distortion of colors in the digitization process.

Since the features come from skin patches, we focus on nude and semi-nude paintings mostly from the Classicism and Renaissance eras. Paintings of 4 painters were chosen: *Peter Paul* ***Rubens*** (Flemish-Dutch Baroque), ***Michaelangelo*** *Buaonarroti* (Italian Renaissance), *Jean-August-Dominique* ***Ingres*** (French Neoclassicism), *Sandro* ***Botticelli*** (Italian Early Renaissance). These painters were chosen because many of their paintings contain the features of interest. From each painter, 25 patches were collected from about 10 different paintings, giving a total of 100 samples.

3.2. Experimental Setup

The following color models were considered: RGB, Smith's HSV (hue, saturation, value) [17], Ostwald's HSI (hue, saturation, intensity) [18], Gonzales and Woods' HLS (hue, lightness, saturation) [19] and CIELAB [20].

Considering the results from Hsu and Lin [9] and Platt [15], the following multi-class SVM methods were compared: one-vs-one with Max Wins, DAGSVM and ECOC SVM. For ECOC SVM we designed the codewords using the *exhaustive method* (for number of classes $3 \leq k \leq 7$) presented by Dietterich and Bakiri [14]. Three types of kernel functions, namely linear, polynomial of degree 2 and Gaussian, were tested with various kernel and regularization parameter values. For Gaussian kernels, the kernel parameter values σ of $0.2, 0.4, \ldots, 5$, were tested. For all three kernels, the regularization parameter values C of $10^0, 10^1, 10^2, 10^3$ were tested. To obtain these parameters using analytical methods, please refer to Chapelle et al. [21]

The leave-one-out method was used in all classification tests since the number of samples is quite small (only 100). The k-nearest-neighbor classifier had also been tested. In almost all cases, its performance was poorer than that of SVM. So, its classification performance is omitted in this paper.

Three types of tests were performed: classification based on (1) single color channel, (2) combining several single-channel classifiers, and (3) combining several color channels into a single input vector.

3.3. Single-Channel Classification Results

Among the three kernels tested, the Gaussian kernel performs the best in most cases. This result is most obvious for DAGSVM. It is less obvious for one-vs-one SVM where Gaussian performs the best in only 2/3 of the tests. In ECOC SVM, Gaussian kernel's advantage is less significant. Degree-2 polynomial kernel performs better or at least the same as linear kernel in many tests.

Experimental results shows that DAGSVM performs the best compared to other multi-class methods (Table 1). With Gaussian kernel, DAGSVM performs the best for almost all color channels. With other kernels, DAGSVM performs the best in about 2/3 of the test cases. Surprisingly, ECOC SVM does not perform very well even though it has a larger number of 7 binary classifiers compared to DAGSVM and one-vs-one SVM, which have 6 classifiers.

The individual classifiers use different kernel and regularization parameters to achieve their optimal performance. The most accurate single-channel classifier is based on a^* of CIELAB. Among HLS, HSI and HSV, the performance of the classifiers using H is the same since their formulations of H are the same. Classification performance based on H is almost as good as that using a^*. The best saturation component is the S of HSI and the best brightness component is the L of HLS.

3.4. Combined Single-Channel Classification Results

Classification performance can be improved by combining several single-channel classifiers using weighted voting. The combined classification accuracies are presented in Table 2. Since DAGSVM had the best performance, the experiment was performed using DAGSVM with Gaussian kernel. The voting weights were chosen based on the assumption that one channel would be dominant and two or more channels together could compensate for the misclassification of the dominant channel. For three-classifier combinations, the possible weights were permutations of 2, 2 and 3. For four-classifier combinations, the possible weights were permutations of 1, 3, 3 and 5 or permutations of 2, 2, 2 and 5. Exhaustive tests were performed to determine the best choice of weights.

The results in Table 2 show that combining RGB single-channel classifiers produces the poorest performance. Among the three-classifier combinations that use the hue-saturation-brightness models, HLS has the best performance. The overall best classification performance is achieved by combining the four best single-channel classifiers, taking into consideration the completeness of hue-saturation-brightness information. That is, the combination of single-channel classifiers for H_{HLS}, S_{HSI}, L_{HLS} and a^* gives the highest classification accuracy of 0.85.

3.5. Combined-Channel Classification Results

Instead of combining single-channel classifiers, a more commonly used approach is to concatenate the profiles of multiple channels into a single, long input vector. We tested this approach on the best color model HLS and the best combination of four channels using DAGSVM and Gaussian kernels. Table 3 shows that combined-channel classifiers have lower classification accuracies than those of combined single-channel classifiers. This comparison result is expected since each single-channel classifier of a combined classifer has its own optimal kernel and regularization parameters, whereas a combined-channel classifier uses only one set of kernel and regularization parameters for the combined input vector.

Table 2. Classification performance of combining single-channel classifiers.

Combination	RGB	HSV	HSI	HLS	CIELAB	$H_{HLS}, S_{HSI}, L_{HLS}, a^*$
Weights	[3 2 2]	[3 2 2]	[2 3 2]	[2 2 3]	[2 3 2]	[3 1 5 3]
Accurarcy	0.68	0.77	0.77	0.82	0.80	0.85

Table 3. Comparison between combined single-channel classifiers and combined-channel classifiers.

Combination	HLS		$H_{HLS}, S_{HSI}, L_{HLS}, a^*$	
	combined-channel	combined-classifier	combined-channel	combined-classifier
No of Classifiers	1	3	1	4
Accurarcy	0.74	0.82	0.81	0.85

4. CONCLUSION

We have presented a method of identifying the painters based on the color profiles of skin patches in painting images. Various color models such as RGB, HSV, HSI, HLS, and CIELAB were tested on their ability to represent color information accurately. In addition, the classification performance of three multi-class SVM classifiers, namely one-vs-one SVM, ECOC SCM and DAGSVM were compared. Test results show that the overall best classification accuracy of 0.85 is achieved by using a weighted voting of four single-channel DAGSVM classifiers. The four color channels are H and L of HLS, S of HSI and a^* of CIELAB.

The test results also confirm that hue is an important feature for expressing color relief in paintings, consistent with the point of view of painting art. Single-channel classifiers based on hue and a^* achieve the best single-channel classification accuracies.

Acknowledgment

This project is supported by This research is supported by NUS ARF R-252-000-072-112 and NSTB UPG/98/015.

5. REFERENCES

[1] Parramón's Editorial Team, *How to Recognize Styles*, Barron, New York, 1997.

[2] R. Sablatnig, P. Kammerer, and E. Zolda, "Hierarchical classification of paintings using face and brush stroke models," in *Proc. of 14th Int. Conf. on Pattern Recognition*, 1998, vol. I, pp. 172–174.

[3] H. J. van den Herik and E. O. Postma, "Discovering the visual signature of painters," in *Future Directions for Intelligent Systems and Information Sciences*, pp. 129–147. Springer-Verlag, 2000.

[4] B. Cole, *The Renaissance Artist at Work: From Pisano to Titian*, Harper & Rows, New York, 1983.

[5] M. Baxandall, *Painting and Experience in Fifteenth-Century Italy*, Oxford University Press, 1991.

[6] R. H. Bartels, J. C. Beatty, and B. A. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*, Morgan Kaufmann, 1998.

[7] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1995.

[8] O. Chapelle, P. Haffner, and V. Vapnik, "Support vector machines for histogram-based image classification," *IEEE Trans. on Neural Networks*, vol. 10, pp. 1055–1064, 1999.

[9] C. W. Hsu and C. J. Lin, "A comparison of methods for multi-class support vector machines," *IEEE Trans. on Neural Networks*, vol. 13, pp. 415–425, 2002.

[10] J. Weston and C. Watkins, "Multi-class support vector machines," Tech. Rep. CSD-TR-98-04, Dept. of Computer Science, University of London, UK, 1998.

[11] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of Machine Learning Research*, 2001.

[12] S. Knerr, L. Personnaz, and G. Dreyfus, "Single-layer learning revisited: A stepwise procedure for building and training a neural network," in *Neurocomputing: Algorithms, Architectures and Applications*, F. Fogelman-Soulié and J. Hérault, Eds., pp. 41–50. Springer-Verlag, 1990.

[13] J. Friedman, "Another approach to polychotomous classification," Tech. Rep., Department of Statistics, Stanford University, 1996.

[14] T. G. Dietterich and G. Bakiri, "Error-correcting output codes: A general method for improving multiclass inductive learning programs," in *Proc. of the 9th AAAI National Conference on Artificial Intelligence*, 1991, pp. 572–577.

[15] J. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin DAGs for multiclass classification," in *Advances in Neural Information Processing Systems*, S. A. Solla, T. K. Leen, and K.-R. Muller, Eds. MIT Press, Cambridge, 2000.

[16] U. Kreßel, "Pairwise classification and support vector machines," in *Advances in Kernel Methods Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds., pp. 255–268. MIT Press, Cambridge, 1999.

[17] A. R. Smith, "Color gamut transform pairs," in *Proc. of the 5th Annual Conf. on Computer Graphics and Interactive Techniques*, 1978, pp. 12–19.

[18] W. Ostwald, *Colour Science*, Winsor and Winsor, 1931.

[19] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley, 1993.

[20] R. S. Berns, *Billmeyer and Saltzman's Principles of Color*, John Wiley and Sons, 2000.

[21] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 131–159, 2002.