Identifying Painters from Color Profiles of Skin Patches in Painting Images

Ivo WIDJAJA

National University of Singapore
2003

Name: Ivo WIDJAJA
Degree: Master of Computing
Dept: Computer Science
Thesis Title: Identifying Painters from Color Profiles of Skin Patches in Painting Images.

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 themselves have pointed out the usefulness of specific quantifiable features in the paintings.

This thesis presents a method for identifying painters using color profiles of skin patches in painting images. A model of color profiles and its feature extraction method are proposed. Various color models for representing the color profiles are explored. Moreover, various implementations of multi-class Support Vector Machine classifiers are compared for their classification performance. Experimental results show that a weighted combination of several Directed Acyclic Graph SVMs with Gaussian kernels gives the best classification performance.

Keywords: Painting Classification
Color Profiles
Skin Patches
Multi-class Support Vector Machines Classifiers
SVM with Error Correcting Output Code
Directed Acyclic Graph Support Vector Machines

Identifying Painters from Color Profiles of Skin Patches in Painting Images

Ivo WIDJAJA
(B. Sc. (Hon.) in Computer Sciences, ITS)

A Thesis Submitted
for the Degree of Master of Computing
Department of Computer Science
School of Computing
National University of Singapore
2003

Acknowledgments

Thank God, it's finally over.

I would like to express my deepest gratitude to my supervisor, A/P Leow Wee Kheng, for providing advice and continuous guidance during my coursework and master's years.

I would also like to express my special thanks to Indri and Li Rui for advices and help in realizing my thesis. Intellectual and spiritual support from many friends make this thing possible.

Lastly, I will always be grateful to my family for their constant prayers when I needed them most.

Contents

Acknowledgments **i**

Table of Contents **ii**

List of Figures **v**

List of Tables **vi**

Summary **vii**

1 Introduction **1**

1.1 Motivation 1

1.2 Problem Definition 3

1.3 Organization of the Thesis 4

2 Artistic Aspects of Paintings **6**

2.1 Painting Classification 6

2.2 Color in Paintings 7

2.3 Lights and Shadows in Paintings 9

2.4 Painting Skin Tones 11

3 Related Work **13**

3.1 Classification of Paintings 13

3.2 Painting Image Retrieval 16

4 Painter Identification Method **19**

4.1 Feature Extraction 20

4.2 Color Models 21

4.2.1 RGB and sRGB 22

4.2.2 HSV Color Model 23

4.2.3 HLS Color Model 25

4.2.4 HSI Color Model 27

4.2.5 CIELAB Color Model 28

4.3 Profile Normalization 29

4.3.1 Cubic Spline Interpolation 30

4.3.2 Linear Interpolation 32

4.4 Multi-class Pattern Recognition 33

4.5 Support Vector Machine 34

4.5.1 Binary Classification using Support Vector Machine 34

4.5.2 Multi-class Support Vector Machine Classifiers 36

5 Experiments and Analysis of Results **43**

5.1 Data Set 43

5.2 Experimental Setup 47

5.2.1 Profile Normalization Method 49

5.2.2 Kernel and Regularizing Parameters 49

5.3 Test Results 53
5.3.1 Single-Channel Classification Results 54
5.3.2 Combined Single-Channel Classification Results 57
5.3.3 Combined-Channel Classification Results 62

6 Future Work **63**
6.1 Painting Samples 63
6.2 Contributing Factors in Features 64
6.3 Classification Method 64
6.4 Combining Classifiers 65

7 Conclusion **66**

Bibliography **68**

Appendix A: Painting Resources **73**

Appendix B: Painting Samples **75**

List of Figures

1.1 Skin patches in Michaelanglo's *The Creation of Adam*. 4

2.1 The *Chiaroscuro* technique. 10

4.1 Color profile extraction. 20
4.2 RGB color model. 22
4.3 Single-hexcone HSV color model. 24
4.4 Double-hexcone HLS color model. 25
4.5 HSI color model. 27
4.6 Cubic spline interpolation. 30
4.7 Linear interpolation. 33
4.8 Support Vector Machine's optimal separating hyperplane. 34
4.9 Decision functions of Directed Acyclic Graph Support Vector Machine. 40

5.1 The *Contrapposto* pose. 44
5.2 Painting samples. 45
5.2 Painting samples. 46
5.3 k-NN's performance of various color models with various k. 56
5.4 Confusion matrix of the best combined single-channel classifier. . . 61

List of Tables

4.1 SVM kernel functions. 36

5.1 Classification accuracy using cubic spline and linear interpolation. . 50

5.2 Best kernel and regularizing parameter values for each single-channel classifiers. 51

5.3 Comparison of classification results between different kernels. 52

5.4 Single-channel classification results. 54

5.5 Comparison with k-NN classification. 57

5.6 Classification performance of combining three single-channel classifiers. 58

5.7 Classification performance of combining four single-channel classifiers. 60

5.8 Comparison between combined single-channel and combined-channel classifiers. 62

Summary

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.

This thesis presents a method for identifying painters using color profiles of skin patches in painting images. A model of color profiles and its feature extraction method are proposed. Various color models for representing the color profiles are explored, namely RGB, HSV, HSI, HLS and CIELAB. Moreover, various implementations of multi-class Support Vector Machine classifiers are compared for their classification performance, namely Directed Acyclic Graph SVM (DAGSVM), one-vs-one SVM and SVM with Error Correcting Output Code (SVM ECOC). Experimental results show that a weighted combination of several Directed Acyclic Graph SVMs with Gaussian kernels gives the best classification

performance. Moreover, user-oriented color models such as HSV, HSI and HLS perform better than hardware-oriented RGB color model.

Chapter 1

Introduction

1.1 Motivation

Recognizing a painter from his paintings is difficult from both the viewpoints of art and science. The intricacy of painting classification lies in the complexity of the features in paintings, their painters' inconsistent styles and painters' many different approaches in their artworks [Gov01, p. 4]. These are some of the reasons why research on digital analysis of painting images has received very little attention. Moreover, 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 [SKZ98] 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 [vdHP00] 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 [Col83, 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* [Bax91, pp. 81–85].

In addition, Baxandall suggested using relief (*rilievo*) to categorize paintings

from the fifteenth century Italy [Bax91, 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 [vdHP00], 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.

1.2 Problem Definition

The input to the problem is a skin patch extracted from a source painting. Our algorithm is to identify the painter of the painting from which the skin patch is drawn. This translates into a pattern classification problem. The thesis should answer how skin patches are extracted and modelled, and which learning methods are used.

To tackle the classification problem for the whole domain of the painting images that exist in the art history will be extremely difficult. We will limit the experimental data set to a collection of paintings from four prominent classical

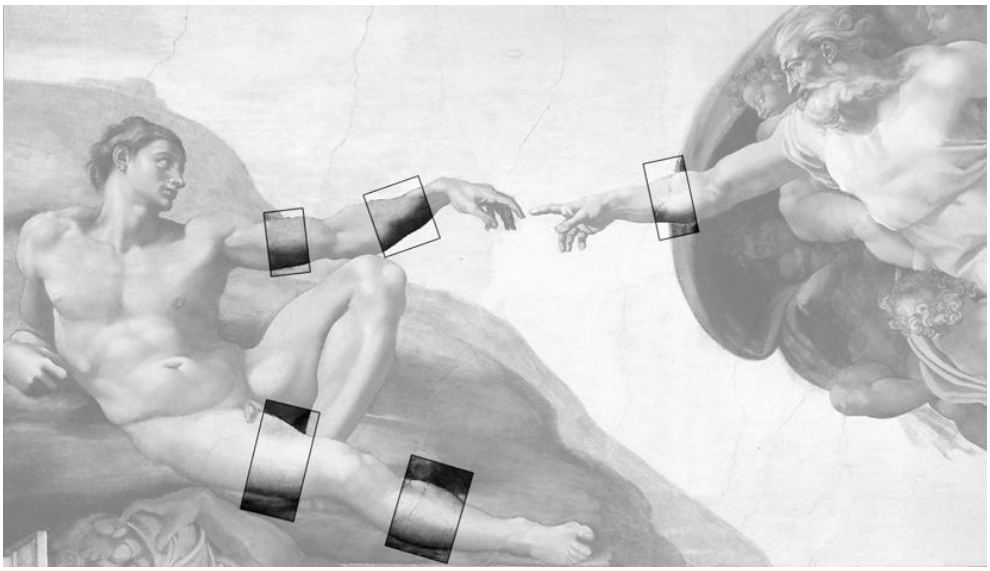


Figure 1.1: Skin patches in Michaelanglo's *The Creation of Adam*.

painters whose artworks contain the feature of interest described in the previous paragraph.

1.3 Organization of the Thesis

To perform painting classification well, it is important to understand paintings from the point of view of the art. This knowledge helps us to build a reliable model of color profile and to analyze the experimental results. In Chapter 2, art historians and art critics' perspectives on painting analysis and color analysis are discussed.

After looking from the art's viewpoint, science's viewpoint is presented in Chapter 3. It summarizes existing work in painting classification and painting image retrieval. Features, methodologies and results of existing work are outlined.

Chapter 4 describes our method of identifying painters from skin patches. It includes details on feature extraction, and the method of building color profiles from skin patches. Various color models such as RGB, HSV, HSI, HLS and CIELAB are covered in depth. This chapter also explains several SVM multi-class classification methods.

Next, experimental results and analysis are discussed in Chapter 5. Possible approaches for improvements and open research areas are discussed in Chapter 6. Finally, Chapter 7 concludes this thesis by summarizing the results and analysis of this research work.

Chapter 2

Artistic Aspects of Paintings

2.1 Painting Classification

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 [Tea97b].

There is parallel progression between style and time-line. Hence, most of the chronological period in painting history usually corresponds to certain style, which can be syntactic, semantic or a combination of both. Baroque, Classicism, Roman-

ticism, Surrealism, Realism and abstract art have different semantic focus on the subject matters [Gov01], while Renaissance, Impressionism, Fauvism and Cubism carry emphasis on some unique syntactic features. Impressionist paintings contain loose brush strokes, brilliant colors, bright ambience, and tinted shadows [Gov01, pp. 157]. Fauvists used saturated colors with intense color interaction or sometimes juxtaposed combination. Pointillist George Seurat from Neo-Impressionism used short dab of primaries to make optical mixture of color.

2.2 Color in Paintings

Form and color constitute the main plastic means by which a painter paints. What colors are used in a painting depend not only on the colors present in the subject matter, but also on the ability of the artist to realize the full potential of his material [Col83, p. 67].

Intrinsically, a painter can decide on what colors he wants to put in the paintings from those available on his palette. Consciously or unconsciously a painter may put his personal signature in the choice of colors. Extrinsically, a painter is bound by several factors in selecting colors: paint material and color symbolism.

Painters use color paints which come principally from mineral, vegetable extracts or synthetic material. The artist's palette is limited by the materials found in his era. Prehistoric color are red ochre and black. Charcoal, red ochre and yel-

low ochre were used in the Stone Age. Egyptian then added turquoise, ultramarine blue, white and green. Cobalt blue and chrome yellow which are commonly used by Turner were available only since 1802 and 1814 respectively [Tow96, p. 41].

Painters also choose colors according to color symbolism, which the art historian Baxandall calls *values of colors* [Bax91, pp. 81–85]. For example, Early Christians banned the use of green, because it had been used in the pagan ceremonies [Fei97, p. 122]. In the Renaissance and Baroque era blue, gold and vermillion were associated with the worship. Divine subject matters were attributed with pure and brilliant hue [Car93, p. 14]. Very often, color symbolism and material consideration are intertwined. The Virgin Mary, the most venerated divine figure, had her robe painted in ultramarine, the rarest and most expensive blue pigment around fifteenth century [Bax91, pp. 6–11].

The color model in the painter's mind is usually the hue-saturation-brightness model. Hue is the *type* of color. Saturation is the colorfullness or the opposite dullness of color. Brightness indicates the lightness and darkness of color. This model was first proposed by Philipp Otto Runge [Car93, p. 36] when he created Runge's sphere to represent the relation between hue, saturation and brightness. Munsell also used a very similar model. Johannes Itten, a *20*th century German painter and art teacher, created a version close to Runge's Sphere [Car93, p. 7].

Most paint materials comes in hue. New hue can be created by mixing two or more hues. Then artist controls the brightness by mixing with black and white

paint. If he wants to control the saturation of the color, then the original hue can be diluted by adding grey mixed or some amount of complementary color. In summary, painters build the whole gamut of color by selecting a hue and adjusting its saturation and brightness.

Art critics also analyze a particular painter based on the hue of their painting materials. Many famous painters used a unique combinations of certain hues. These colors are usually referred as the painter's palette. Rembrandt's palette, for instance, was rich in ochres, umbers and siennas and all earth hues [Car93, p. 31]. Monet's palette was dominated by primary hues, green and white [Car93, p. 41]. Turner tended to use primary colors [Tow96, Gag01, p. 35, p. 164]. These facts show that hue is an important feature in differentiating the paintings.

2.3 Lights and Shadows in Paintings

The first traditional method to model forms and volume using transition of light and dark called *chiaroscuro* was introduced by Leonardo da Vinci. It relies on systematic change in value resulting in highlight, light, shadow, core of shadow, reflected shadow and cast shadow (Figure 2.1) [Fei97, pp. 40-41]. Da Vinci had also suggested the use of complementary color in shadow.

Painters use tonal value to depict depth in their two-dimensional paintings. In the simplest form, a painter simply applies black and white paint to control the

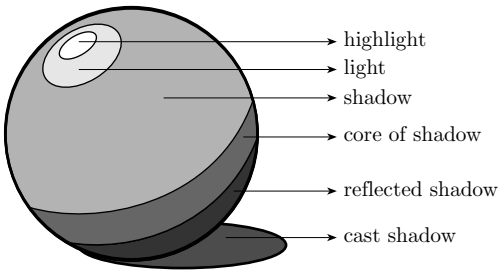


Figure 2.1: The *Chiaroscuro* technique.

brightness value of a color to create highlights and shadows. The conception of color in the Middle Ages before Renaissance was essentially brightness or value-based [Gag01, p. 68]. As an example, in thirteenth century, Giotto had produced shading using darker and lighter values of the same hue [Fei97, p. 134].

New techniques and styles emerged after color theories were developed by several artists and scientists. One of the most influential theory was given by Wolfgang Von Goethe. His book *Theory of Colors* presented several important points: the role of complementary colors in shadows, simultaneous contrast created by complementary colors, and intense color interaction induced by hues [Fei97, p. 14]. It introduced hue to add more dimension in selecting color and creating tone. Now, a common method to create shadow is to use a darker tone mixed with blue, the object's color and its complementary.

Many painters have their own unique way to produce lights and shadows. Impressionist Auguste Renoir used pink and yellow for highlights, blue and violet for

shadows [Gov01, pp. 157]. Fauvist painters like Henri Matisse used direct complementary color on shadows to give more dramatic quality [Tea97a, pp. 110-111].

2.4 Painting Skin Tones

The technique for painting skin tones is basically similar to the general method of creating lights and shadows of painting. The mixture for skin color in shadow are detailed in [Tea97a, pp. 130-131]. Red-based and yellow-based colors are the foundation of the skin color. White, blue and green can be added to accentuate the shadows and highlights.

In [Tea96, p. 82], the method to create highlights is presented. Highlights in painting rarely depict the maximal degree of lights. Highlights reflect the atmosphere colors, so their color tends toward white plus the color range of the chromatic surroundings within which the object lies. Hues of green, blue and earth color can be used as the part of the mixture. This concept is applicable to shadow, too.

Creating tonality does not mean just applying dark and light value or controlling one dimensional brightness. The whole gamut of hue, saturation and brightness affects the depiction of the three dimensional subject matter in paintings. Other than that, different painters have different ways of distributing the range of color from light to dark. Ingres' paintings, for instance, seems to have a

flatter distribution of tonal transition [Gag01, p. 199].

Chapter 3

Related Work

3.1 Classification of Paintings

Among all related works, the work presented by Sablatnig et al. [SKZ98] has the most similar objective with our work in terms of painting classification. Sablatnig et al. introduced a hierarchically structured classification scheme for portrait miniatures. It examined the *structural signature* of a painter based on brush strokes. It assumed that the painter applies an individual line system of strokes and colors to his strokes' basic pattern. Three levels of classification were used: color, shape of region, and structure of brush strokes.

In the first level, color classification was performed by computing the mean RGB value of the image. The image was then transformed to HLS model. The

L-channel of the image was used for face extraction. A schematic face model was then constructed from two ellipsoids fitted to the shape of the face. Several characteristic regions were segmented using this model.

In the second level, shape classification was performed by comparing similar regions between different paintings. This region-based matching reduced the complexity of comparison since the search space was reduced to specific regions of interest. Next, the stroke detector was applied in the regions of interest to retrieve brush strokes for stroke classification. Finally in the third level brush strokes were classified by comparing curvatures and orientations of neighboring stroke segments.

Sablatnig's approach took advantage of the fact that paintings from certain periods like Renaissance and Baroque and from many painters like Rembrandt contain large proportions of human figures, whereas Impressionist paintings contain larger proportions of landscapes. Unfortunately, only qualitative results were mentioned in the paper.

Herik and Postma [vdHP00] presented a comprehensive experiment on painting classification using neural network on many features. The data set contained 60 paintings by 6 impressionist and neo-impressionist painters: Claude Monet, Vincent van Gogh, Paul Cézanne, Alfred Sisley, Camille Pissarro and George Seurat. The painting images were processed by re-sampling and low-pass filtering using Gaussian filter to remove JPEG artifact. Then many features were extracted and selected for classification. These features included:

- global features:
  - fractal dimension,
  - Fourier spectra,
  - RGB color histogram,
  - Independent Components (using FastICA),
  - HSI color histogram,
  - oriented receptive-field histograms,
  - hue histogram
- region features: These features were measured in nine non-overlapping regions of the painting
  - mean intensity,
  - skewness,
  - Kurtosis,
  - Standard Deviation,
- combination of fractal dimension, Fourier spectra and mean intensity

The features of the training images were used to train the multi-layer neural network using the standard back-propagation algorithm. Generalization performance was measured by means of 10-fold cross-validation. Over-fitting was avoided using early-stopping criterion.

Classifications based on individual features yielded generalization performance above chance level ($\frac{1}{6}$). The best three features were fractal dimension, Fourier spectra and mean intensity. The first two features represent texture and brushwork. Combination of these three features yielded classification accuracy of 0.85 on the test set. Surprisingly, mean intensity measured over the nine regions turned out to outperform more complex color features such as RGB and HSI histograms. RGB histogram unexpectedly outperformed the HSI histogram.

3.2 Painting Image Retrieval

Most of the image retrieval systems are based on syntactic features such as color histogram, Fourier spectra and texture. Other features such as shape, principal color and composition are also used.

There is a growing number of systems that try to use semantic features. Most of them bridge the semantic and syntactic gap by labelling syntactic features with semantic meaning. One example is to use color red, orange and purple to suggest warm color, and blue and green to suggest cool color.

Hachimura [Hac95, Hac96] proposed a method that used *principal* color as a feature in color painting retrieval. First, an image was segmented using K-means clustering with Godlove's color distance measured in the Munsell color space. After the initial segmentation, smaller regions were merged. Principal color regions were

chosen from the resulting regions that have the following properties:

1. great *moment arm* value (based on Moon and Spencer color harmony theory) defined as $M = \sqrt{64(V-5)^2 + C^2}$, where V and C are Munsell's value and chroma of the region color, respectively.
2. large region area
3. large region compactness
4. located in the center of the painting

Hachimura examined 400 Japanese paintings but did not mentioned any quantitative result.

Corridoni et al. [CdBMV96, CdBP98] presented a system which translated Johannes Itten's formalism of color into a formal language that associated semantics with a combination of chromatic properties of painting images. Itten's formalism analyzes the use of color in art and its effects on user's psyche. For instance, certain colors can suggest warm or cool mood. In Corridoni's method, the semantics were induced by colored regions segmented using CIELUV and k-means. They were described in relation to *intra-region* and *inter-region* properties. Color, warmth, hue, luminance, saturation, position and size were intra-region properties. Inter-region properties included contrast and harmony of hue, saturation and luminance. The system was implemented to retrieve 1000 paintings dating from Renaissance to the present.

Isomoto et al. [IYYI99] used color, shape and impression keywords as features of paintings. Impression words such as *bright*, *happy* and *lovely* were used to express the atmosphere of the paintings. Each feature was described as a fuzzy set and fuzzy membership function was used for image retrieval. The sample set and test results were not described in their paper.

Chapter 4

Painter Identification Method

This chapter presents our method of identifying the painter based on skin patches in painting images. It is organized as follows:

1. Firstly, the feature extraction method and the model for representing skin patches are detailed.

2. Secondly, various color models for representing colors are explained.

3. Lastly, several Support Vector Machines for multi-class classification are presented.

4.1 Feature Extraction

The feature required is in the form of color profiles of skin patches. Our method of extracting color profile features consists of 4 main steps (Fig. 4.1):

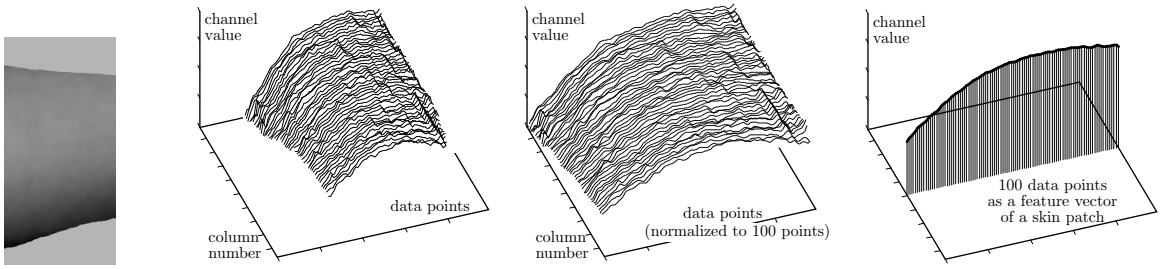


Figure 4.1: Color profile extraction. (a) Aligned image of a skin patch. (b) Extracted profile curves of a color channel, one curve for each column in the aligned image. (c) Normalized profile curves. (d) Averaged profile curve.

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. 4.1(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. 4.1(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. 4.1(c)) using cubic spline interpolation (Section 4.3.1). 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. 4.1(d)). This profile curve, which contains 100 data points, will be used as the feature vector for SVM classification.

4.2 Color Models

The following color models are considered:

- The sRGB model [SACM96]
- The HSV (hue, saturation, value) model [Smi78]

- The HSI (hue, saturation, intensity) model [GW93]
- The HLS (hue, lightness, saturation) model [Ost31]
- The CIELAB model [Ber00]

Since the RGB and hue-saturation-brightness-based color models are not perceptually uniform, perceptually uniform CIELAB color space is also tested in the experiment. This model is also widely use in some color-based classification algorithms.

4.2.1 RGB and sRGB

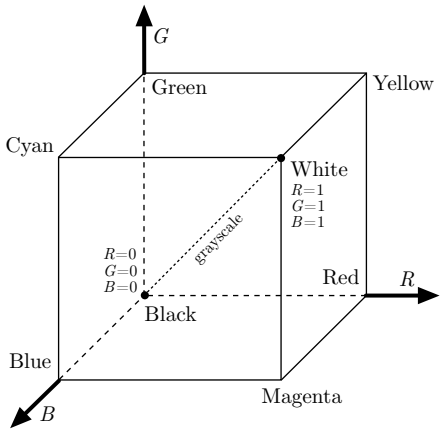


Figure 4.2: RGB color model.

RGB color model uses orthogonal Cartesian system to represent the Red, Green and Blue channels (or components). Its conceptual model forms a unit cube shown

in Fig. 4.2.

The sRGB model [SACM96] is a standard RGB model used for representing colors for internet use. In the sRGB, R, G and B values range from 0 to 255, instead of 0 to 1.

4.2.2 HSV Color Model

HSV (Hue, Saturation, Value) color model proposed by Smith [Smi78] is based on the intuitive appeal of the artist's tint, shade and tone [FvFH90]. It uses the cylindrical coordinate system and forms a six-sided pyramid. The hue H is measured by the angle around the center V axis starting with Red at 0^0. The value of S starts from 0 at the center V axis to 1 on the sides of the hexcone. The top of the cone corresponds to White with $V = 1$ and the apex at the bottom represents Black with $V = 0$.

The conversion from RGB to HSV is given as follows. Let

$$M = max(R, G, B)$$

$$m = min(R, G, B)$$

$$d = M - m$$

The value of V is calculated as follows:

$$V = M \tag{4.1}$$

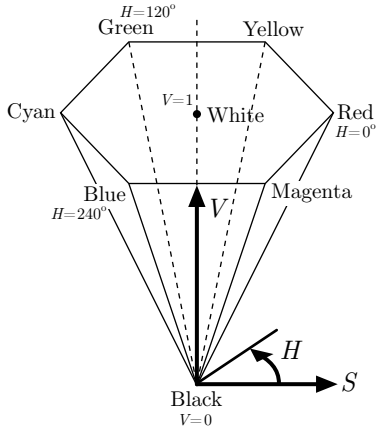


Figure 4.3: Single-hexcone HSV color model.

The value of S is given by:

$$S = \begin{cases} 0, & \text{if } M = 0 \\ d/M, & \text{otherwise} \end{cases} \tag{4.2}$$

If $S = 0$, then hue H is $undefined$. Otherwise,

$$H' = \begin{cases} (G - B)/d, & \text{if } M = R \\ 2 + (B - R)/d, & \text{if } M = G \\ 4 + (R - G)/d, & \text{if } M = B \end{cases} \tag{4.3}$$

The value of hue H is obtained from H' as follows:

$$H = \begin{cases} 60H' + 360, & \text{if } H' < 0 \\ 60H', & \text{otherwise} \end{cases} \tag{4.4}$$

4.2.3 HLS Color Model

Ostwald [Ost31] proposed the HLS (Hue, Luminance, Saturation) model. It uses the cylindrical coordinate system and forms a double hexcone (Fig. 4.4). It is produced from the HSV model by pulling HSV's White point upward to make an upper cone. The hue H is measured similarly as that in HSV. The strongest colors in the HLS model are achieved at $L = 0.5$, whereas they are achieved in the HSV model by setting $S = 1$ and $V = 1$.

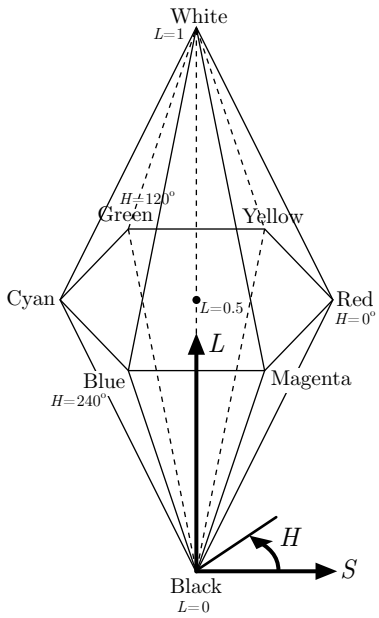


Figure 4.4: Double-hexcone HLS color model.

The conversion from RGB to HLS is given as follows. Let

$$M = M(R, G, B)$$

$$m = m(R, G, B)$$

$$d = M - m$$

$$D = M + m$$

The value of L is calculated as follows:

$$L = D/2 \tag{4.5}$$

The value of S is given by:

$$S = \begin{cases} 0, & \text{if } d = 0 \\ d/D, & \text{if } L \leq 0.5 \\ d/(2 - D), & \text{otherwise} \end{cases} \tag{4.6}$$

If $d = 0$, then hue H is $undefined$. Otherwise,

$$H' = \begin{cases} (G - B)/d, & \text{if } M = R \\ 2 + (B - R)/d, & \text{if } M = G \\ 4 + (R - G)/d, & \text{if } M = B \end{cases} \tag{4.7}$$

The value of hue H is obtained from H' as follows:

$$H = \begin{cases} 60H' + 360, & \text{if } H' < 0 \\ 60H', & \text{otherwise} \end{cases} \tag{4.8}$$

4.2.4 HSI Color Model

HSI (Hue, Saturation, Intensity) model was presented by Gonzales and Woods [GW93]. It also uses the cylindrical coordinate system and forms a double tricone (Fig. 4.5). The intensity I is decoupled from hue and saturation. It is suggested that this model is compatible with hardware despite its intuitiveness. As for HSV and HLS, the hue H is measured similarly ranging from 0^0 to 360^0.

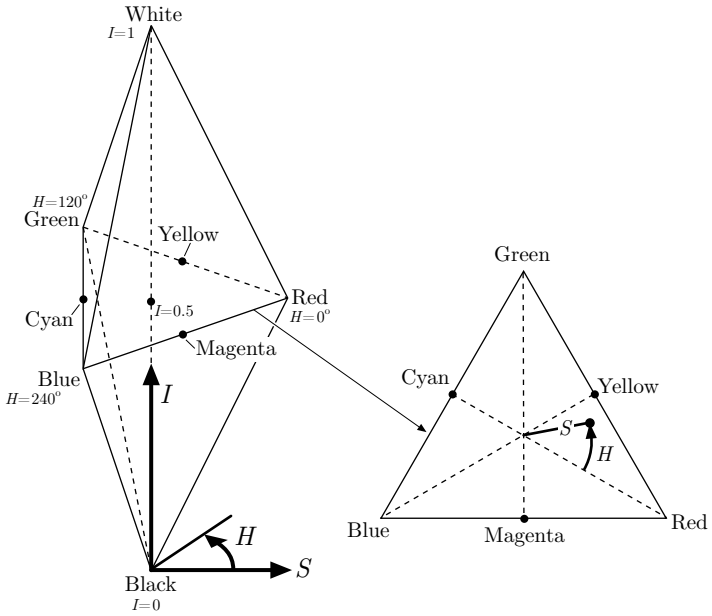


Figure 4.5: HSI color model.

The conversion from RGB ($[0, 1]$ range) to HSI according to Gonzales and Woods [GW93] is:

$$I = \frac{1}{3}(R + G + B) \tag{4.9}$$

$$S = 1 - \frac{3}{R + G + B} \min(R, G, B) \tag{4.10}$$

$$H = \cos^{-1}\left\{ \frac{(R - G) + (R - B)}{2\sqrt{(R - G)^2 + (R - B)(G - B)}} \right\} \tag{4.11}$$

4.2.5 CIELAB Color Model

RGB, HSV, HLS and HSI spaces are not perceptually uniform. This means in these spaces the same Euclidean distances at different locations or orientations are not perceptually identical. The CIE (International Commission on Illumination) developed CIELAB to overcome this problem. CIELAB has three components: L^*, a^* and b^*, where L^* describes luminance and a^* and b^* describe chromaticity.

The transformation from sRGB to CIELAB is done as follows [Ber00]:

$$R' = R/255, \qquad G' = G/255, \qquad B' = B/255 \tag{4.12}$$

where $0 \leq R, G, B \leq 255$.

Then, apply gamma correction:

$$R_s = \begin{cases} R'/12.92, & \text{if } R' = 0.03928 \\ \left[\dfrac{R' + 0.055}{1.055}\right]^{2.4}, & \text{otherwise} \end{cases} \tag{4.13}$$

G_s and B_s are obtained in the same manner. Next, transform $R_sG_sB_s$ to CIEXYZ.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R_s \\ G_s \\ B_s \end{bmatrix} \tag{4.14}$$

Finally, generate CIELAB values with the following equations:

$$L^* = 116\left[f\left(\frac{Y}{Y_n}\right) - \frac{16}{116}\right] \tag{4.15}$$

$$a^* = 500\left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right] \tag{4.16}$$

$$b^* = 200\left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right] \tag{4.17}$$

$$f(w) = \begin{cases} w^{1/3}, & \text{if } w > 0.008856 \\ 7.787w + \frac{16}{116}, & \text{otherwise} \end{cases}$$

where X_n, Y_n, Z_n are the tristimulus values of the reference white:

$$X_n = 0.950456, \qquad Y_n = 1, \qquad Z_n = 1.088754$$

4.3 Profile Normalization

Profile normalization is performed to normalize the length of the profile curves to 100 points. For the normalization, we tested two different types of interpolation:

cubic spline interpolation and linear interpolation.

4.3.1 Cubic Spline Interpolation

Cubic spline is used to interpolate data points between a sequence of $n+1$ points [BBB98]. It assumes that the equation formed by the interpolated points is a polynomial of third degree.

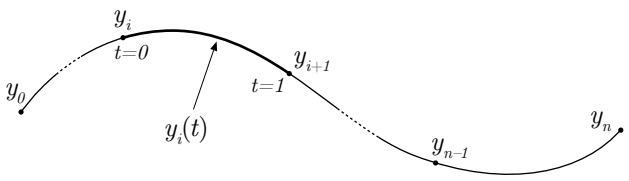


Figure 4.6: Cubic spline interpolation.

Consider a 1-dimensional spline of $n+1$ points $y_0, y_1, ..., y_n$. The ith piece of spline between y_i and y_{i+1} is represented by a cubic equation:

$$y_i(t) = a_i + b_i t + c_i t^2 + d_i t^3, \qquad t \in [0,1] \tag{4.18}$$

From $y_i(t)$ and its derivative $y_i'(t)$ we have

$$y_i(0) = y_i = a_i \tag{4.19}$$

$$y_i(1) = y_{i+1} = a_i + b_i + c_i + d_i \tag{4.20}$$

$$y_i'(0) = D_i = b_i \tag{4.21}$$

$$y_i'(1) = D_{i+1} = b_i + 2c_i + 3d_i \tag{4.22}$$

Solving Eq. 4.19–4.22 obtains:

$$a_i = y_i \tag{4.23}$$

$$b_i = D_i \tag{4.24}$$

$$c_i = 3(y_{i+1} - y_i) - 2D_i - D_{i+1} \tag{4.25}$$

$$d_i = 2(y_i - y_{i+1}) + D_i + D_{i+1} \tag{4.26}$$

The splines should be continuous at the connecting points, i.e.,

$$y_i(0) = y_i \tag{4.27}$$

$$y_i(1) = y_{i+1} \tag{4.28}$$

$$y_i'(1) = y_{i+1}'(0) \tag{4.29}$$

$$y_i''(1) = y_{i+1}''(0) \tag{4.30}$$

It is assumed that the second derivative at the end points are zero:

$$y_0''(0) = 0 \tag{4.31}$$

$$y_n''(1) = 0 \tag{4.32}$$

Rearranging all these equations leads to a symmetric tridiagonal system [BBB98, p. 12-13]:

$$\begin{bmatrix} 2 & 1 & & & & & & \\ 1 & 4 & 1 & & & & & \\ & 1 & 4 & 1 & & & & \\ \dots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & 1 & 4 & 1 & & \\ & & & & 1 & 4 & 1 & \\ & & & & & 1 & 2 & \end{bmatrix} \begin{bmatrix} D_0 \\ D_1 \\ D_2 \\ \vdots \\ D_{n-2} \\ D_{n-1} \\ D_n \end{bmatrix} = \begin{bmatrix} 3(y_1 - y_0) \\ 3(y_2 - y_0) \\ 3(y_3 - y_1) \\ \vdots \\ 3(y_{n-1} - y_{n-3}) \\ 3(y_n - y_{n-2}) \\ 3(y_n - y_{n-1}) \end{bmatrix} \tag{4.33}$$

Solving this equation we can obtain $y_i(t)$ for each spline segment using Eq. 4.23-4.26.

4.3.2 Linear Interpolation

Linear interpolation is basically taking the equation of a straight line between two points to interpolate the values in between.

The formula for 1-dimensional linear interpolation is:

$$y_i(t) = y_i + (y_{i+1} - y_i)t, \qquad t \in [0, 1] \tag{4.34}$$

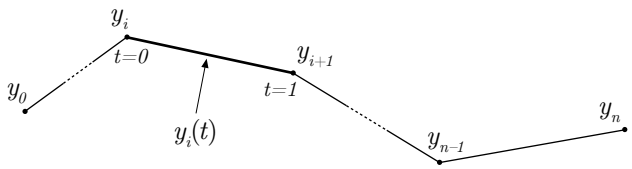


Figure 4.7: Linear interpolation.

4.4 Multi-class Pattern Recognition

Let $(\mathbf{x}_i, d_i), 1 \leq i \leq N$, be a set of independent sample points, where $\mathbf{x}_i \in \mathfrak{R}^n$ is a vector of n components and $d_i \in \{1, ..., k\}$ is the class label of $\mathbf{x}_i$. The k-class pattern recognition problem is to construct a decision function $f(\mathbf{x}, \alpha)$ that determines the class of a sample $\mathbf{x}$. α is a vector of parameters of the function. The problem is to find parameters α such that the total of the loss function Λ over all points is kept minimum. The loss function can be stated as:

$$\Lambda(d, f(\mathbf{x}, \alpha)) = \begin{cases} 0, & \text{if } d = f(\mathbf{x}, \alpha) \\ 1, & \text{otherwise} \end{cases} \tag{4.35}$$

where d is the actual class of the sample $\mathbf{x}$.

4.5 Support Vector Machine

4.5.1 Binary Classification using Support Vector Machine

Support Vector Machine (SVM) for the binary, i.e., 2-class, pattern classification problems has been well studied. It performs well compared to other learning methods such as neural networks [Vap95]. For example, Chapelle tested SVM for histogram-based image classification and obtained good results [CHV99].

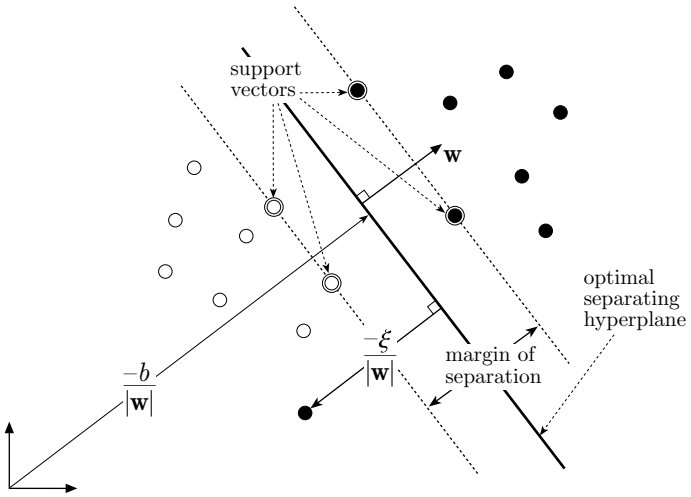


Figure 4.8: Support Vector Machine's optimal separating hyperplane.

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. SVM classification

is stated as a quadratic optimization problem:

$$L(\mathbf{w}, \xi) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i \tag{4.36}$$

with the following constraints: $\forall i, y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$. The optimal hyperplane is given by $\mathbf{w}$. C is the regularization parameter.

In real-world pattern recognition task patterns in two or more classes are often not well-separated by a linear hyperplane. A non-linear mapping is introduced to map the input vectors into the *feature space*. The optimal hyperplane is then constructed in this feature space.

The function $\phi(\mathbf{x})$ denotes the mapping of input vector $\mathbf{x} = (x_1, ..., x_n)$ to a higher dimensional feature space. SVM classification in higher dimensional feature space is solved by minimizing:

$$W(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{4.37}$$

where *kernel function* $K(\mathbf{x}_i, \mathbf{x}_j) = \phi^T(\mathbf{x}_i)\phi(\mathbf{x}_j)$ is the dot product of the mapping function $\phi(\mathbf{x})$ and α_i are Lagrange multipliers. The decision function of the SVM is:

$$f(\mathbf{x}) = sgn\big(\sum_{i=1}^{N}\alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b\big) \tag{4.38}$$

The most commonly used kernels are given in Table 4.1. Schölkopf [SSB+97] has compared SVM with Gaussian kernel to classical Radial Basis Function classifier (RBF) and concluded that SVM yields excellent results.

Table 4.1: SVM kernel functions.

Type of SVM	Kernel function $K(\mathbf{x}, \mathbf{x}_i)$	Comments
Polynomial	$(\mathbf{x}^T\mathbf{x}_i + 1)^p$	p is chosen a priori and is the degree of the polynomial kernel
Gaussian (or RBF)	$e^{-\|\mathbf{x}-\mathbf{x}_i\|^2/2\sigma^2}$	σ is the spread of the Gaussian kernel
Hyperbolic tangent	$\tanh(\beta_0\mathbf{x}^T\mathbf{x}_i + \beta_1)$	only for some values of β_0 and β_1

4.5.2 Multi-class Support Vector Machine Classifiers

SVM was initially developed to solve two-class or binary pattern recognition problems. But real world tasks are usually multi-class pattern recognition problems. There have been several approaches developed to tackle this problem. Hsu and Lin [HL02] presented a comprehensive comparison of many multi-class SVM methods. They are basically divided into two approaches:

- The first approach considers all classes at once and solving one optimization problem.
- The second approach combines several binary classifiers to form a multi-class classifier. There are several methods in this approach:
  - One-vs-rest SVM
  - One-vs-one SVM
  - SVM with Error Correcting Output Code (ECOC)

– Directed Acyclic Graph SVM (DAGSVM)

Weston and Watkins [WW98] proposed a method that considers 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. The primal objective used is similar to those in [BB99, Vap98]. Crammer and Singer [CS01] also used a similar approach and achieved a lower error rate than the one-vs-rest method.

Hsu and Lin [HL02] presented a test result showing that DAGSVM and one-vs-one SVM are more practical than other methods. Platt [PCST00] presented a result that suggests that DAGSVM performs slightly better or at least the same compared to one-vs-rest and one-vs-one.

The above methods of combining binary SVM classifiers to form a multi-class classifier are described in the following sections.

One-vs-rest SVM

This method, also called one-against-all, is the simplest implementation of multi-class SVM classifier. It basically constructs k binary SVMs. Let the training set consists of N samples $(\mathbf{x}_i, d_i)$, where $\mathbf{x}_i \in \mathfrak{R}^n$ and $d_i \in \{1, ..., k\}$. Thus, the uth classifier is trained with class u samples as the positive samples and the remainder as the negative samples. That is, the training samples for the uth classifier are

$(\mathbf{x}_i, y_i)$ where

$$y_i = \begin{cases} +1, & \text{if } d_i = u \\ -1, & \text{otherwise} \end{cases}$$

The distance to the optimal separating hyperplane of the uth classifier is:

$$h_u(\mathbf{x}) = \sum_i \alpha_{u,i} y_i K(\mathbf{x}, \mathbf{x}_i) + b_u, \qquad 1 \leq u \leq k \tag{4.39}$$

The class of $\mathbf{x}$ determined by the one-vs-rest classifier, denoted by $H(\mathbf{x})$, is given by the binary classifier with the largest $h_u(\mathbf{x})$ value:

$$H(\mathbf{x}) = \arg \max_{u \in \{1,\ldots,k\}} h_u(\mathbf{x}) \tag{4.40}$$

One-vs-one SVM

Knerr [KPD90] suggested another method by constructing all possible binary classifiers. For a k-class problem, there are ${}^kC_2 = k(k-1)/2$ classifiers. During training, the classifier for class u and v will not use all samples but only the samples in class u and v. Thus, the training samples for class u and v binary classifier are $(\mathbf{x}_i, y_i)$ where:

$$y_i = \begin{cases} +1, & \text{if } d_i = u \\ -1, & \text{if } d_i = v \end{cases}$$

There are several methods for combining the results of these binary classifiers. Knerr suggested using the AND operator [KPD90]. Friedman [Fri96] suggested the *Max Wins* strategy and Kreßel [Kre99] has applied it with excellent results. Hsu

et al. [HL02] compared *Max Wins* and several other multi-class SVM methods and concluded that *Max Wins* is more suitable for practical application. The *Max Wins* strategy uses the following voting scheme:

The decision function $h_{uv}(\mathbf{x})$ of the classifier for class u and v is:

$$h_{uv}(\mathbf{x}) = sgn\big(\sum_i \alpha_{uv} y_i K(\mathbf{x}, \mathbf{x}_i) + b_{uv}\big) \tag{4.41}$$

If $h_{uv}(\mathbf{x}) = +1$ then increase the vote for class u by one. Otherwise, increase the vote for class v by one. The predicted class of $\mathbf{x}$ is given by the class with the largest vote.

Directed Acyclic Graph SVM (DAGSVM)

The disadvantage of the one-versus-one and the one-versus-rest is that these methods do not have bounds on the generalization error. Platt et al. [PCST00] suggested a method called *Directed Acyclic Graph Support Vector Machine* which has a bound on the generalization error. It also has superior testing time compared to the one-versus-one method.

DAGSVM also uses $k(k-1)/2$ binary classifiers and the same training algorithm as the one-versus-one method. However, in testing, DAGSVM uses a rooted binary directed acyclic graph which has $k(k-1)/2$ internal nodes and k leaves. Each internal node is a binary classifier for class u and v. To classify an input $\mathbf{x}$, binary decision functions in the nodes are evaluated starting with the root node. Then,

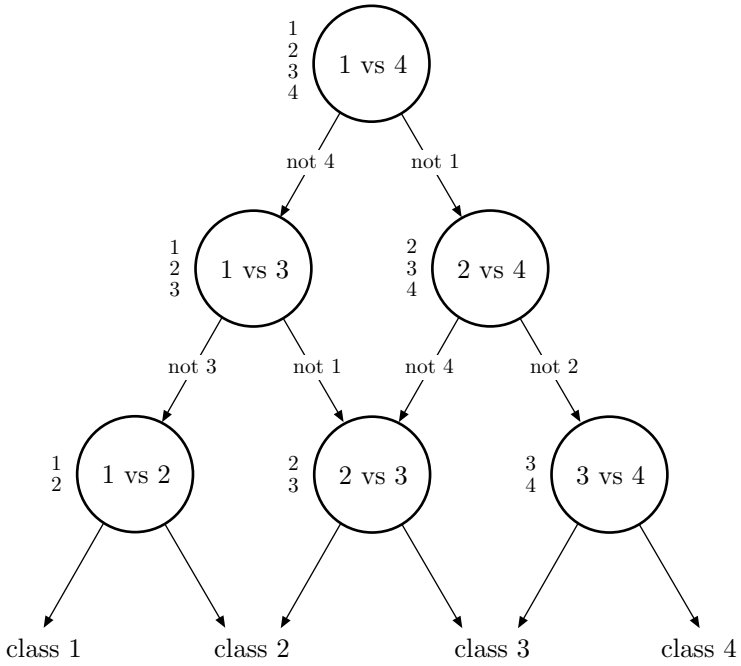


Figure 4.9: Decision functions of Directed Acyclic Graph Support Vector Machine. Each node represents a binary classifier and possible classes for each node are shown next to it.

based on the output of the nodes, the graph is traversed down to a leaf node which gives the class of $\mathbf{x}$ (Fig. 4.9).

SVM with Error-Correcting Output Code (ECOC)

Dietterich and Bakiri [DB91] presented Error-Correcting Output Code (ECOC) as a method to combine many binary classifiers to solve multi-class classification problems. Rennie and Rifkin [RR02] has applied SVM with ECOC for text classification and obtained classification performance better than that of Naive Bayes.

Let k be the number of classes, m be the number of binary classifiers used and $\mathbf{R}$ be a *code matrix* with k rows and m columns. This matrix represents the *data split* which defines the assignment of the training samples to the binary classifiers. Let $R_{uv} \in \{+1, -1\}$ be the matrix element in the uth row and vth column. $\mathbf{R}_u$ is the uth row vector and $\mathbf{R}_{.v}$ is the vth column vector of matrix $\mathbf{R}$. $\mathbf{R}_u$ defines the *codeword* for class u. $\mathbf{R}_{.v}$ defines the data split for building two superclasses for the vth binary classifier. Let the training samples for the vth binary classifier be $(\mathbf{x}_i, y_i)$. Samples $(\mathbf{x}_i, y_i)$ are derived from all original samples $(\mathbf{x}_i, d_i)$ and matrix $\mathbf{R}$. The vth binary classifier is trained with class u samples ($d_i = u$) as positive samples, if $R_{uv} = +1$ or as negative samples, if $R_{uv} = -1$:

$$y_i = \begin{cases} +1, & \text{if } d_i = u \text{ and } R_{uv} = +1 \\ -1, & \text{if } d_i = u \text{ and } R_{uv} = -1 \end{cases}$$

Intuitively, each row in $\mathbf{R}$ gives each class a code such that codes of different classes are very different (ideally orthogonal). Each column divides all samples from all classes into only 2 classes for each binary classifier. Different column vectors are made as different as possible (ideally orthogonal). The error correcting property is achieved by classifying a pattern to the class with the class code that is nearest to the one produced by ECOC SVM.

One of the challenges in solving k-class classification with ECOC is the design of good codewords. In principles the good codewords should satisfy two properties: (1) good separation in Hamming distance between each codeword and (2) each

column $\mathbf{R}_{.v}$ of matrix $\mathbf{R}$ should be well separated from other columns. It also means there is good separation in Hamming distance between each column and its complement.

Let the vth decision function be:

$$h_v(\mathbf{x}) = sgn\big(\sum_i \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b\big) \tag{4.42}$$

and the predicted vector denoted by $\mathbf{h}(\mathbf{x})$ be:

$$\mathbf{h}(\mathbf{x}) = \{h_1(\mathbf{x}), ..., h_m(\mathbf{x})\}$$

The class of $\mathbf{x}$ for ECOC classifier denoted by $H(\mathbf{x})$ is given as:

$$H(\mathbf{x}) = \arg \min_{u \in \{1,...,k\}} \Delta(\mathbf{h}(\mathbf{x}), \mathbf{R_u}) \tag{4.43}$$

where Δ is a distance function between the target row vector $\mathbf{R_u}$ and the predicted vector. The simplest example of Δ is the Hamming distance. $\mathbf{h}(\mathbf{x})$ can also be a real value probability or confidence level of membership. In SVM, it may be:

$$f_v(\mathbf{x}) = \sum_i \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b$$

Allwein et al. [ASS00] used loss function g for various model as an alternative approach to the distance function.

$$H(\mathbf{x}) = \arg \min_{u \in \{1,...,k\}} \sum_{v=1}^{m} g(f_v(\mathbf{x}), R_{uv}) \tag{4.44}$$

Chapter 5

Experiments and Analysis of Results

5.1 Data Set

The paintings were taken from *Artchive* (`http://www.artchive.com`) and *Web Gallery of Art* (`http://gallery.euroweb.hu/`) web sites. From both collections, we selected paintings which contain nudes with *contrapposto* pose [Tea97b] and/or paintings that expose arms and legs with transition from dark to light similar to contrapposto pose. *Contrapposto* is a classical pose of a standing figure, with all weight of the body resting on one straight leg, while the other leg is bent at the knee. *Birth of Venus* is an excellent example of *contrapposto*.

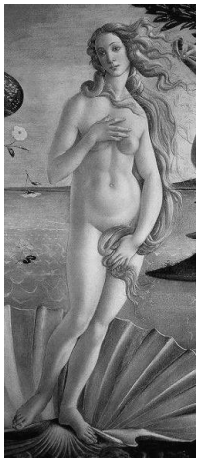


Figure 5.1: The *Contrapposto* pose.

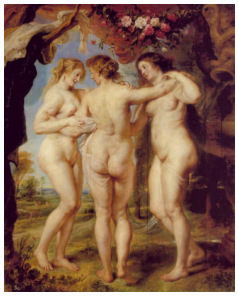
Since the features were extracted from skin patches, we focused on nude and semi-nude paintings mostly from the Classicism and Renaissance eras. Paintings of 4 painters were chosen. These painters were chosen because many of their paintings contain the features of interest. From each painter, 25 patches were collected from about 10 different paintings, giving a total of 100 samples. The four painters were:

1. *Peter Paul* ***Rubens*** (1577-1640) (11 paintings)

   From Flemish-Dutch Baroque period and famous for his colorist style, he introduced a new paradigm generously endowed in flesh and rhythms to the iconography of nude paintings [Tea97b].

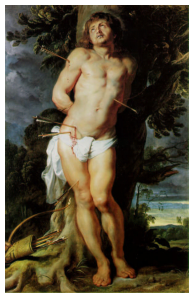
2. ***Michaelangelo*** *Buaonarroti* (1475-1564) (10 paintings)

   Carried the legacy of Italian Renaissance, he was not particularly interested

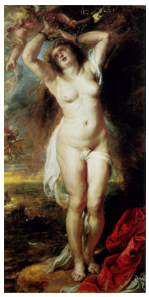


(a) The Three Graces

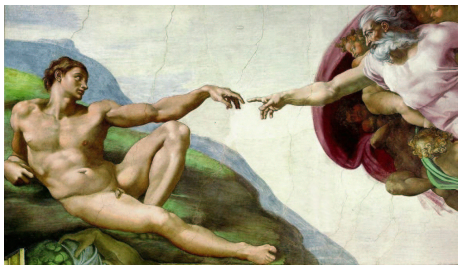
(b) The Little Fur



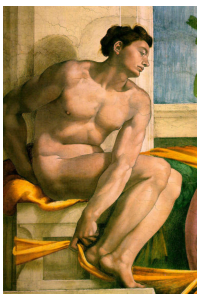
(c) St. Sebastian



(d) Andromeda

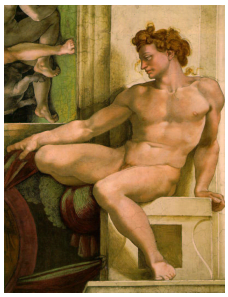


(e) The Creation of Adam



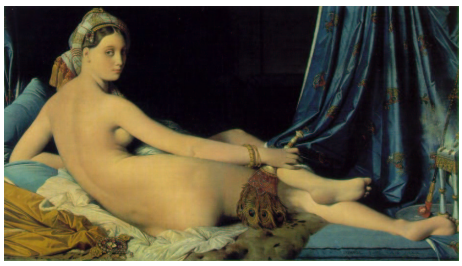
(f) Ignudi 2

(g) The Flood



(h) Ignudi 1

Figure 5.2: Painting samples of (a,b,c,d) Peter Paul **Rubens**, (e,f,g,h) **Michaelangelo** Buaonarroti.



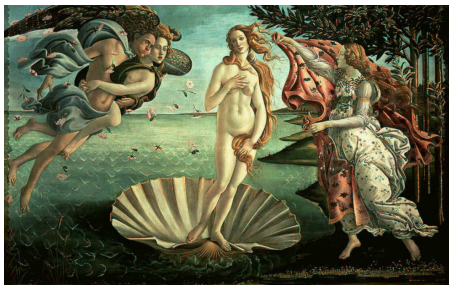
(i) La Grand Odalisque

(j) The Source

(k) Le Bain Turc

(l) Odalisque and Slave

(m) Venus and Mars



(n) The Birth of Venus

(o) Primavera

(p) Madonna of the Pomegranate

Figure 5.2: Painting samples of (i,j,k,l) Jean-August-Dominique **Ingres**, (m,n,o,p) Sandro **Botticelli**.

in color. His *The Creation of Adam* fresco in Sistine Chapel is hailed as one of the most beautiful nudes in art history. His impressive record in nude paintings can be seen in *Last Judgment* which contains about 400 full-bodied nudes.

3. *Jean-August-Dominique* ***Ingres*** (1780-1867) (15 paintings)

   As a prominent figure in French Neoclassicism, he showed distinguished style by the emphasis of drawing over color. Ingres' works show strong-sense classicism which is reflected by his female nude paintings [Tea97b].

4. *Sandro* ***Botticelli*** (1445-1510) (11 paintings)

   Worked in second half of Early Renaissance in Italy, he presented delicate and rhythmic composition in his poetic nudes, painted in sober colors. *Birth of Venus* is one of his classic nudes [Tea97b].

5.2 Experimental Setup

The following color models were considered: RGB, Smith's HSV (hue, saturation, value) [Smi78], Ostwald's HSI (hue, saturation, intensity) [Ost31], Gonzales and Woods' HLS (hue, lightness, saturation) [GW93] and CIELAB [Ber00]. Considering the results from Hsu and Lin [HL02] and Platt [PCST00], the following multi-class SVM methods were compared: one-vs-one with Max Wins, DAGSVM and ECOC SVM.

In our experiment we used $k = 4$ classes. The codewords for ECOC SVM were designed using the *exhaustive methods* (for $3 \leq k \leq 7$) presented by Dietterich and Bakiri [DB91]. The codeword length was $2^{k-1} - 1$. Row 1 contained all ones. Row 2 consisted of 2^{k-2} negative ones, followed by $2^{k-2} - 1$ ones. Row 3 consisted of 2^{k-3} negative ones, followed by 2^{k-3} ones, followed by 2^{k-3} negative ones, followed by $2^{k-3} - 1$ ones. Row i consisted of alternating 2^{k-i} minus ones and plus ones. With this method, the following matrix $\mathbf{R}$ for our 4 classes problem was used:

$$\begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ -1 & -1 & -1 & -1 & +1 & +1 & +1 \\ -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ -1 & +1 & -1 & +1 & -1 & +1 & -1 \end{bmatrix} \tag{5.1}$$

Since the codeword had a length of 7 bits, ECOC SVM used 7 binary classifiers. On the other hand, DAGSVM and one-vs-one SVM with Max Wins contained ${}^4C_2 = 6$ binary classifiers.

Three types of kernel functions, namely linear, polynomial of degree 2 and Gaussian, were tested with various kernel and regularization parameter values. Details are presented in Section 5.2.2.

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 is also tested for comparison.

All the tests for choosing profile normalization, kernel and regularizing param-

eters were performed with single-channel classifiers, which classify the skin patches based on only one color channel.

5.2.1 Profile Normalization Method

Two methods of normalizing the length of profile curves were compared: cubic spline and linear interpolation. The comparison was performed using the DAGSVM method and Gaussian kernel, a combination that gave the best classification result (See Section 5.2.2 for details). The Gaussian kernel parameter σ was set at 0.5 and regularizing parameter C was fixed at 100.

It turns out that there was no significant difference in classification performance between these two methods (Table 5.1). Nevertheless, since the cubic spline performed better or at least the same in 9 out of 15 channels, cubic spline was used as the profile normalization method in the subsequent experiments.

5.2.2 Kernel and Regularizing Parameters

The values of the Gaussian kernel parameter σ and regularizing parameter C affect the performance of the classifiers. The parameter values can be selected in two ways [Hay95]: (1) experimentally by measuring the performance on testing set for a set of the parameter values, and (2) analytically by estimating the bounds on generalization performance (see Chapelle et al. [CVBM02] and Keerthi [Kee02] for

Table 5.1: Classification accuracy using cubic spline and linear interpolation. Gaussian's $\sigma = 0.5$, regularizing parameter $C = 100$.

Channel	cubic spline	linear
R	**0.51**	0.49
G	0.63	**0.64**
B	**0.61**	0.60
H	**0.69**	0.69
S	0.57	**0.58**
V	**0.52**	0.51
H	**0.70**	0.70
S	0.59	**0.60**
I	**0.68**	0.67
H	**0.69**	0.69
S	**0.50**	0.50
L	0.66	**0.68**
L^*	0.55	**0.57**
a^*	0.69	**0.69**
b^*	**0.53**	0.51

details).

In comparing the generalization performance of different multi-class SVM methods, Hsu and Lin [HL02] used the first approach to find the best parameter values. In our experiment, the same approach was used to obtain these 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.

Table 5.2: Best kernel and regularizing parameter values for each single-channel classifiers.

	Linear		Poly 2		Gaussian		
Channel	C	success	C	success	σ	C	success
R	10^3	0.51	10^0	0.51	0.20	10^1	0.57
G	10^2	0.61	10^0	0.58	2.80	10^1	0.67
B	10^1	0.59	10^0	0.59	0.40	10^2	0.62
H	10^2	0.68	10^2	0.71	2.00	10^2	0.74
S	10^1	0.67	10^0	0.65	0.20	10^3	0.61
V	10^3	0.53	10^2	0.56	0.40	10^1	0.57
H	10^2	0.69	10^2	0.71	0.60	10^3	0.74
S	10^3	0.60	10^1	0.60	1.40	10^1	0.62
I	10^2	0.55	10^1	0.59	0.40	10^1	0.67
H	10^2	0.68	10^2	0.71	2.00	10^2	0.74
S	10^0	0.59	10^0	0.52	0.40	10^1	0.61
L	10^3	0.60	10^0	0.59	0.80	10^1	0.68
L^*	10^2	0.58	10^0	0.55	3.20	10^1	0.63
a^*	10^1	0.64	10^1	0.76	0.60	10^1	0.75
b^*	10^0	0.53	10^1	0.60	2.40	10^1	0.62

The test results are summarized in Table 5.2. It contains the parameter values for the most accurate single-channel classifier for each color channel and kernel

function. The multi-class classification method used was DAGSVM. For simplicity, all binary classifiers within the single-channel classifier were assumed to have the same kernel function.

Table 5.3: Comparison of classification results between different kernels. **Bold face** represents the best performance for each color channel. † represents the best kernel's performance given a particular method and a specific channel.

Method	DAGSVM			One-vs-one with Max Wins			SVM with ECOC		
Channel	Linear	Poly 2	Gaussian	Linear	Poly 2	Gaussian	Linear	Poly 2	Gaussian
R	0.51	0.51	**0.57**†	0.55	0.50	0.56†	0.54†	0.51	0.42
G	0.61	0.58	**0.67**†	0.60	0.55	0.64†	0.60†	0.57	0.49
B	0.59	0.59	**0.62**†	0.61†	0.60	0.56	0.61	**0.62**†	0.54
H	0.68	0.71	**0.74**†	0.62	0.65	0.70†	0.64	0.64	0.64†
S	**0.67**†	0.65	0.61	0.57	0.63†	0.61	0.58	0.60†	0.38
V	0.53	0.56	**0.57**†	0.56	0.45	0.56†	0.56†	0.51	0.43
H	0.69	0.71	**0.74**†	0.62	0.64	0.71†	0.65	0.65	0.68†
S	0.60	0.60	**0.62**†	0.59	0.60†	0.59	0.55	0.57	0.58†
I	0.55	0.59	**0.67**†	0.57	0.53	0.58†	0.58	0.52	0.60†
H	0.68	0.71	**0.74**†	0.62	0.65	0.70†	0.64	0.64	0.64†
S	0.59	0.52	**0.61**†	0.51	0.51	0.55†	0.54	0.56†	0.53
L	0.60	0.59	**0.68**†	0.46	0.52	0.60†	0.52	0.55	0.61†
L^*	0.58	0.55	**0.63**	0.61	**0.63**†	0.56	0.57	0.59†	0.41
a^*	0.64	**0.76**†	0.75	0.52	0.75†	0.73	0.64	0.58	0.75†
b^*	0.53	0.60	0.62†	0.47	0.50	**0.63**†	0.48	0.51†	0.46

Table 5.3 presents the complete results of generalization performance for the three different multi-class SVM methods and the three different kernels. Among the three kernels tested, the Gaussian kernel performs the best in most cases. This result is most obvious for DAGSVM. 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. It is less obvious for the one-vs-one SVM because the Gaussian performs the best in only 2/3 of the tests. For 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.

5.3 Test Results

Three types of painter identification 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. All the tests were performed using normalization with cubic spline interpolation, Gaussian kernels and their best kernel parameters (Table 5.2). The first test compared three different multi-class classification methods. Implied from the first test's result, the second and third tests used DAGSVM as the multi-class classifiers. The results are presented in the following sections.

Table 5.4: Single-channel classification results. **Bold face** represents the best performance for a channel among the 3 SVM methods.

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

5.3.1 Single-Channel Classification Results

Experimental results show that DAGSVM performs the best compared to other multi-class methods (Table 5.4). 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 (See Section 5.2 for a

discussion on the number of binary classifiers).

The individual classifiers use different kernel and regularization parameter values (Table 5.2) 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.

Comparison with k-NN

To assess the performance of SVM, we compared its classification performance with that of the k-Nearest-Neighbors (k-NN) algorithm. Figure 5.3 shows the classification accuracy of k-NN with various k using the leave-one-out method.

Table 5.5 summarizes the best performance for the single-channel classifiers and their comparison with SVM. The table shows that in most cases SVM outperforms k-NN.

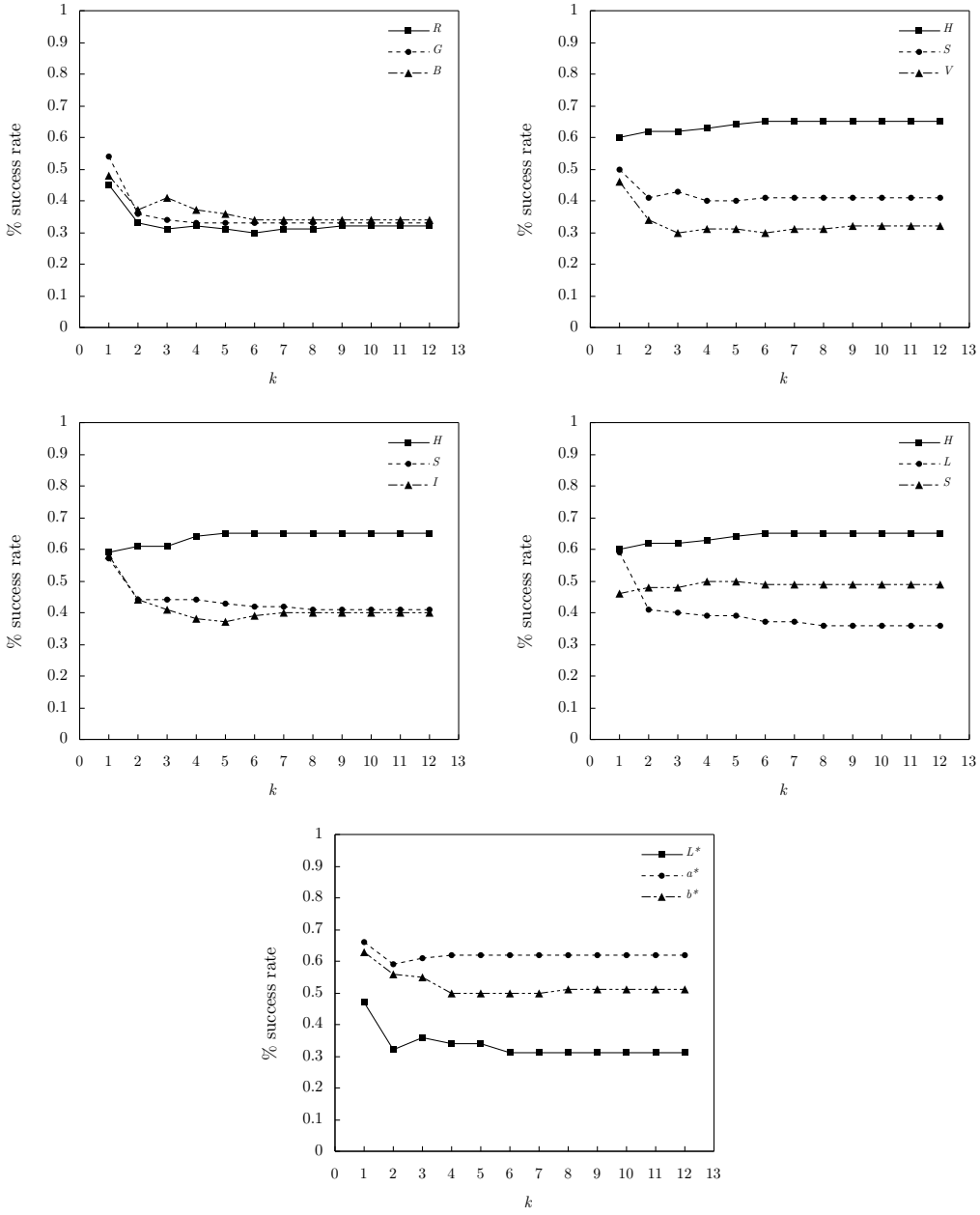


Figure 5.3: k-NN's performance of various color models with various k.

Table 5.5: Comparison with k-NN classification.

Channel	SVM	k-NN
R	**0.57**	0.45 (k=1)
G	**0.67**	0.54 (k=1)
B	**0.62**	0.48 (k=1)
H	**0.74**	0.65 (k=6)
S	**0.61**	0.50 (k=1)
V	**0.57**	0.46 (k=1)
H	**0.74**	0.65 (k=5)
S	**0.62**	0.57 (k=1)
I	**0.67**	0.58 (k=1)
H	**0.74**	0.65 (k=5)
S	**0.61**	0.50 (k=4)
L	**0.68**	0.59 (k=1)
L^*	**0.63**	0.47 (k=1)
a^*	**0.75**	0.66 (k=1)
b^*	0.62	**0.63** (k=1)

5.3.2 Combined Single-Channel Classification Results

Classification performance can be improved by combining several single-channel classifiers using weighted voting. The experiment was performed using DAGSVM

Table 5.6: Classification performance of combining three single-channel classifiers.

Color Model	weights	success
RGB	[3 2 2]	0.68
HSV	[3 2 2]	0.77
HSI	[2 2 3]	0.77
HLS	[2 3 2]	**0.82**
CIELAB	[2 3 2]	0.80

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 combined classification accuracies are presented in Table 5.6 and 5.7.

Combining Three Single-Channel Classifiers

The results in Table 5.6 show that combining RGB single-channel classifiers produces the poorest performance. This fact is consistent with the average performance of its individual channel which is the lowest amongst other color models' tristimulus combinations (Table 5.3).

Among the three-classifier combinations that use the hue-saturation-brightness models, HLS has the best performance. HLS has the two best individual channel performances: H_{HLS} for its hue channel and L_{HLS} for its brightness channel. HSV, which has the lowest performance in saturation-based channel S_{HSV} and brightness-based channel V_{HSV}, yields poorer combined single-channel classification accuracy.

Combining Four Single-Channel Classifiers

To increase the classification performance further, combinations of 4 single-channel classifiers were tested. Table 5.7 illustrates the various combinations. In combination (1) and (2), I_{HSI} and L_{HLS} were combined with HSV to compensate for the *weak* brightness channel V_{HSV}. Both combinations increased the performance of HSV from 0.77 to 0.81.

The next two combinations adopted similar approach by introducing the *good* channel to compensate for the *weak* channel of the same type. Combination (3) introduced the best saturation channel S_{HSI} to HLS. But, HLS' performance remains at 0.82. Combination (4) adds brightness channel L_{HLS} to CIELAB, since L_{HLS} performs the best compared to other brightness channels (I_{HSI}, V_{HSV} and L^*). CIELAB performance was increased from 0.80 to 0.82.

Finally we considered other combinations: (5) and (6) in Table 5.7. We com-

Table 5.7: Classification performance of combining four single-channel classifiers.

	H	S	V	L^*	a^*	b^*	H	S	I	H	L	S	weights	success
1	√	√	√						√				[5 3 3 1]	0.81
2	√	√	√								√		[5 3 1 3]	0.81
3								√		√	√	√	[1 3 5 3]	0.82
4				√	√	√					√		[1 5 3 3]	0.82
5							√	√	√		√		[5 1 3 3]	0.83
6					√		√	√			√		[3 3 1 5]	**0.85**

bined several best single-channel classifiers by taking into consideration the completeness of hue-saturation-brightness information. I_{HSI}, V_{HSV}, L_{HLS}, and L^* are the brightness channels. Channels a^* and b^* carry both hue and saturation information.

In combination (5) we introduced L_{HLS} to increase HSI's performance. This combination yielded a classification accuracy of 0.83.

The overall best classification performance was achieved by combination (6). It combined the three best hue-saturation-brightness channels and a^*, which was the best single-channel classifier. The combination of single-channel classifiers for H_{HSI}, S_{HSI}, L_{HLS} and a^* gave the highest classification accuracy of 0.85.

The confusion matrix of the best combination is given at Figure 5.4.

$$\begin{bmatrix} 20 & 1 & 2 & 2 \\ 2 & 19 & 2 & 2 \\ 2 & 0 & 21 & 2 \\ 2 & 2 & 6 & 15 \end{bmatrix}$$

a^* (CIELAB)

$$\begin{bmatrix} 19 & 2 & 1 & 3 \\ 2 & 21 & 0 & 2 \\ 2 & 0 & 21 & 2 \\ 6 & 1 & 5 & 13 \end{bmatrix}$$

H (HSI)

$$\begin{bmatrix} 16 & 6 & 3 & 0 \\ 6 & 12 & 4 & 3 \\ 3 & 2 & 19 & 1 \\ 2 & 4 & 4 & 15 \end{bmatrix}$$

S (HSI)

$$\begin{bmatrix} 16 & 3 & 3 & 3 \\ 6 & 14 & 3 & 2 \\ 4 & 2 & 19 & 0 \\ 3 & 2 & 1 & 19 \end{bmatrix}$$

L (HLS)

$$\begin{bmatrix} 21 & 1 & 0 & 3 \\ 3 & 20 & 1 & 1 \\ 1 & 0 & 24 & 0 \\ 3 & 2 & 0 & 20 \end{bmatrix}$$

Combined

Classification accuracy			Class assignment		
a^* (CIELAB)	:	0.75	class 1	:	Rubens
H (HSI)	:	0.74	class 2	:	Micheangelo
S (HSI)	:	0.62	class 3	:	Ingres
L (HLS)	:	0.68	class 4	:	Botticelli
Combined	:	0.85			

Figure 5.4: Confusion matrix of the best combined single-channel classifier.

5.3.3 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 two combinations of four channels using DAGSVM and Gaussian kernels.

Table 5.8 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 5.8: Comparison between combined single-channel and combined-channel classifiers.

Channels	combined-classfier	combined-channel
H_{HLS}, L_{HLS} and S_{HLS}	0.82	0.74 (C=10, σ=0.2)
H_{HSI}, S_{HSI}, I_{HSI} and L_{HLS}	0.83	0.81 (C=10, σ=0.2)
a^*_{Lab}, H_{HSI}, S_{HSI} and L_{HLS}	0.85	0.81 (C=100, σ=0.4)

Chapter 6

Future Work

6.1 Painting Samples

Due to limitation of resources, we have used a limited set of painting images. The quantity of samples can be increased by acquiring more painting images from paid painting database providers. Moreover, these providers supply color-managed and higher resolution images, which should improve the quality of the color profiles extracted from the images. Appendix A shows a list of painting sources and databases.

6.2 Contributing Factors in Features

Artist's intentional style [Col83] and unintentional characters from art materials used contribute to the final color in the paintings. The art materials include painting material (tempera or oil) and panting base (walls, woods or canvas). Rubens and Ingres used oil on canvas, Botticelli used tempera on canvas or wood, and Michaelangelo used fresco technique, i.e., painting on the wall. Our current tests do not separate these contributing factors. We can examine these factors as separate variables to understand how painting materials and their associated techniques affect the classification.

6.3 Classification Method

We have investigated the performance of different approaches in Multi-class Support Vector Machine that combine binary classifiers. Several possibilities to increase the classification accuracy exist such as tweaking the loss function in ECOC SVM and fine-tuning kernel and regularizing parameter selection.

In this work, experiment with ECOC SVM was performed only with hinge loss function. For further research, different loss function can be explored. The framework of combining binary classifiers with loss function was presented by Allwein et al. [ASS00].

In our experiment, we applied the same kernel and regularizing parameter values to a channel across different multi-class methods. The individual binary classifiers of the multi-class methods also used the same kernel and regularizing parameters. The effect of setting different parameter values for different binary classifiers has yet to be examined. Methods to find kernel parameters analytically [CVBM02, Kee02] may increase the classifier's performance.

6.4 Combining Classifiers

For simplicity we use simple weighted voting to combine the single-channel classifier. We can extend the idea of loss function to combine multiple classifiers as proposed by Allwein et al. [ASS00] to create more sophisticated final classifier. One way is to provide probabilistic or confidence value in the results of individual single-channel classifiers, in addition to the determined class. The confidence values can then be used for combining the classification results of the single-channel classifiers.

Chapter 7

Conclusion

In this work, we have demonstrated the potential of identifying the painters from color profiles of skin patches in painting images. Although the experiments have not covered the whole domain of painters and painting samples, this preliminary research has given insight to the characteristic, attributes and modelling of skin patches in painting images.

We have examined various color models for representing the features of the samples. User-oriented color models (hue-saturation-brightness models) such as HSV, HSI and HLS perform better than hardware-oriented model RGB. CIELAB also performs reasonably well. Combining the classifiers from various color channels can increase the performance of the classification accuracy, with the best performance achieved at 0.85.

The results also confirm the art's viewpoint of the importance of paintings' hues. In all color models except RGB, the classifier based on hue-oriented component always performs the best.

We also investigated the performance of various Multi-class Support Vector Machine classifiers in solving pattern recognition problem. Three different kernels have been tested and it turns out that the Gaussian kernel provides excellent results for this classification problem. Directed Acyclic Graph SVM (DAGSVM) proposed by Platt [PCST00] performs the best compared to other multi-class method such as one-vs-one with Max Wins and SVM ECOC.

Nevertheless, this work is a localized exploration inside the space of painter recognition problem and painting analysis since the skin patches provide only one small part of the large collection of painting features. Other local and global features, low-level and higher-level features are obviously needed to construct a practical and efficient classifier of paintings.

Bibliography

[ASS00] E. L. Allwein, R. E. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. In *Proc. 17th Int'l Conf. on Machine Learning*, pages 9–16, 2000.

[Bax91] M. Baxandall. *Painting and Experience in Fifteenth-Century Italy.* Oxford University Press, 1991.

[BB99] E. Bredensteiner and K. P. Bennet. Multicategory classification by Support Vector Machines. *Computational Optimizations and Applications*, pages 53–79, 1999.

[BBB98] 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.

[BBS99] B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors. *Advances in Kernel Methods - Support Vector Learning.* MIT Press - Cambridge, 1999.

[Ber99] A. Berger. Error-correcting output coding for text classification. In *IJCAI'99: Workshop on Machine Learning for Information Filtering*, 1999.

[Ber00] R. S. Berns. *Billmeyer and Saltzman's Principles of Color.* John Wiley and Sons, 2000.

[Bur98] C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[Car93] L. Caruso, editor. *Color.* Dorling Kindersley, London, 1993.

[Caw00] G. C. Cawley. *MATLAB Support Vector Machine Toolbox (v0.54β).* School of Information Systems,University of East Anglia, 2000. `http://theoval.sys.uea.ac.uk/~gcc/svm/toolbox`.

[CdBMV96] J. M. Corridoni, A. del Bimbo, S. De Magistris, and E. Vicario. A visual language for color-based painting retrieval. In *Proc. of 1996 IEEE Symposium on Visual Languages*, pages 68–75, 1996.

[CdBP98] J. M. Corridoni, A. del Bimbo, and P. Pala. Retrieval of paintings using effects induced by color features. In *Proc. of 1998 IEEE International Workshop on Content-Based Access of Image and Video Database*, pages 2–11, 1998.

[CHV99] O. Chapelle, P. Haffner, and V. Vapnik. Support Vector Machines for histogram-based image classification. *IEEE Trans. on Neural Networks*, 10:1055–1064, 1999.

[CL00] C.-C. Chang and C.-J. Lin. *LIBSVM – A Library for Support Vector Machines (v2.36)*, 2000.
`http://www.csie.ntu.edu.tw/~cjlin`.

[Col83] B. Cole. *The Renaissance Artist at Work: From Pisano to Titian.* Harper & Rows, New York, 1983.

[CS01] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2001.

[CST00] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines.* Cambridge University Press, 2000.

[CV95] C. Cortes and V. Vapnik. Support vector network. *Machine Learning*, 20:1–25, 1995.

[CVBM02] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for Support Vector Machines. *Machine Learning*, 46(1-3):131–159, 2002.

[DB91] 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*, pages 572–577, 1991.

[Fei97] E. A. Feisner. *How to Use Color in Art and Design.* Lawrence King Publishing, 1997.

[Fri96] J. Friedman. Another approach to polychotomous classification. Technical report, Department of Statistics, Stanford University, 1996.

[FvFH90] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: principles and practice.* Addison-Wesley, 2nd edition, 1990.

[Gag01] J. Gage. *Colour and Meaning — Art, Science and Symbolism.* Thames and Hudson, 2001.

[Gha00] R. Ghani. Using error-correcting codes for text classification. In Pat Langley, editor, *Proc. of ICML-00, 17th Int'l Conf. on Machine Learning*, pages 303–310. Morgan Kaufmann Publishers, 2000.

[Gov01] B. Govignon, editor. *The Beginner's Guide to Art.* Harry N. Abrams, Inc., New York, 2001.

[GW93] R. C. Gonzalez and R. E. Woods. *Digital Image Processing.* Addison-Wesley, 1993.

[Hac95] K. Hachimura. Extraction of principal colors from Japanese paintings. In *Proc. of 9th Scandinavian Conference on Image Analysis*, pages 457–464, 1995.

[Hac96] K. Hachimura. Retrieval of paintings using principal color information. In *Proc. of 1996 Int'l. Conf. on Pattern Recognition*, volume 3, pages 130–134, 1996.

[Hay95] S. Haykin. *Neural Network: Comprehensive Foundation.* Springer, 1995.

[HL02] C. W. Hsu and C. J. Lin. A comparison of methods for multi-class Support Vector Machines. *IEEE Trans. on Neural Networks*, 13:415–425, 2002.

[IYYI99] Y. Isomoto, K. Yoshine, H. Yamasaki, and N. Ishii. Color, shape and impression keywords as attributes of paintings for information retrieval. In *Proc. of 1999 IEEE Int'l. Conf. on Systems, Man, and Cybernetics*, volume 6, pages 257–262, 1999.

[Kee02] S. S. Keerthi. Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms. *IEEE Trans. on Neural Networks*, 13(5), 2002.

[KPD90] S. Knerr, L. Personnaz, and G. Dreyfus. Single-layer learning revisited: A stepwise procedure for building and training a neural network. In F. Fogelman-Soulié and J. Hérault, editors, *Neurocomputing: Algorithms, Architectures and Applications*, pages 41–50. Springer-Verlag, 1990.

[Kre99] U. Kreßel. Pairwise classification and Support Vector Machines. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods Support Vector Learning*, pages 255–268. MIT Press, Cambridge, 1999.

[LL03] R. Li and W. K. Leow. From region features to semantic labels: A probabilistic approach. In *Proc. Multimedia Modeling*, pages 402–420, 2003.

[MSC97] H. Maitre, F. J. M. Schmitt, and J. Crettez. High quality imaging in museums from theory to practice. In *Proc. SPIE, 3025*, pages 30–39, 1997.

[Ost31] W. Ostwald. *Colour Science*. Winsor and Winsor, 1931.

[PCST00] J. Platt, N. Cristianini, and J. Shawe-Taylor. Large margin DAGs for multiclass classification. In S. A. Solla, T. K. Leen, and K.-R. Muller, editors, *Advances in Neural Information Processing Systems*. MIT Press, Cambridge, 2000.

[RHC90] Y. Rui, T. S. Huang, and S.-F. Chang. Image retrieval: Current techniques, promising directions and open issues. *Journal of Visual Communications and Image Representation*, 10, 1990.

[Rif00] R. Rifkin. *SvmFu*, 2000. `http://five-percent-nation.mit.edu/SvmFu/`.

[RR02] J. D. M. Rennie and R. Rifkin. Improving multi-class text classification with the Support Vector Machine. Technical report, Massachusetts Institute of Technology, 2002. Published as AI Memo 2001-026.

[SACM96] M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta. A standard default color space for the internet — sRGB, 1996. `http://www.color.org/srgb.html`.

[SKZ98] R. Sablatnig, P. Kammerer, and E. Zolda. Hierarchical classification of paintings using face and brush stroke models. In *Proc. of 14th Int'l. Conf. on Pattern Recognition*, volume I, pages 172–174, 1998.

[Smi78] A. R. Smith. Color gamut transform pairs. In *Proc. of the 5th Annual Conf. on Computer Graphics and Interactive Techniques*, pages 12–19, 1978.

[SSB+97] B. Schölkopf, K.-K. Sung, C. J. C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik. Comparing Support Vector Machines with Gaussian kernels to Radial Basis Function classifiers. *IEEE Trans. on Signal Processing*, 45(11):2758–2765, 1997.

[SWS+00] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(12):1–32, 2000.

[Tea96] Parramón's Editorial Team. *Oils*. Barron's, New York, 1996.

[Tea97a] Parramón's Editorial Team. *All about Techniques in Color*. Barron's, New York, 1997.

[Tea97b] Parramón's Editorial Team. *How to Recognize Styles*. Barron's, New York, 1997.

[Tow96] J. Townsend. *Turner's Painting Techniques*. Tate Publishing, 1996.

[Vap95] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[Vap98] V. Vapnik. *Statistical Learning Theory*. John Wiley, New York, 1998.

[vdHP00] H. J. van den Herik and E. O. Postma. Discovering the visual signature of painters. In *Future Directions for Intelligent Systems and Information Sciences*, pages 129–147. Springer-Verlag, 2000.

[WMC+00] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for SVMs. In *Proc. NIPS*, pages 668–674, 2000.

[WW98] J. Weston and C. Watkins. Multi-class Support Vector Machines. Technical Report CSD-TR-98-04, Dept. of Computer Science, University of London, UK, 1998.

Appendix A: Painting Resources

The following resources contain free painting image collection online. Most of them are in JPEG format.

- **Web Gallery of Art**
  (`http://gallery.euroweb.hu/`)
  The Web Gallery of Art is a virtual museum and searchable database of European paintings of the Gothic, Renaissance and Baroque periods (1150-1800), currently containing over 11,000 reproductions.
- **Web Museum**
  (`http://www.ibiblio.org/wm/`)
- **Artchive**
  (`http://www.artchive.com/`)
- **CGFA**
  (`http://cgfa.sunsite.dk/`)
  Searchable fine art images collected by Carol Gerten (Jackson).
- **The Art Renewal Center**
  (`http://www.artrenewal.org/`)
- **ArtCyclopedia**
  (`http://www.artcyclopedia.com/`)
  It contains more than 7,500 artists with links to images online.

These remaining resorces are professional fine art painting image providers and charge professional licensing fees. Most of them provide the artwork in transparencies, some provide in digital format. These providers supply excellent image quality.

- **Art Resource Inc**
  (`http://www.artres.com/`)
  Art Resource is one of the world's largest fine art stock photo archives. It offers licensing to three million transparencies and black and white photographs of works of painting, sculpture, architecture and the minor arts.

- **Erich Lessing Culture and Fine Arts Archives**
  (`http://www.lessing-photo.com/`)
  It has over 30000 large-format colour transparencies including fine arts. Photographs which come from over 900 museums cover works by more than 2200 artists. The images are high resolution (120 MB) and are color controlled. Access via internet and electronic transfer in all desired sizes is available.

- **Scala Archives**
  (`http://www.scalarchives.it`)
  It provides around 80,000 large format transparencies of fine arts from most of Italy's museums and some museums outside Italy.

- **Bildarchiv Foto Marburg**
  (`http://www.fotomr.uni-marburg.de/`)
  Foto Marburg collection includes almost all the major museums in Germany.

- **The Bridgeman Art Library**
  (`http://www.bridgeman.co.uk/`)

- **Runion des Muses Nationaux (RMN)**
  (`http://www.photormn.fr`)
  The RMN of France provides access to works of art in most of the French national and regional museums.

Appendix B: Painting Samples

Peter Paul ***Rubens*** (1577-1640), (11 paintings)

1610-11	The Elevation of the Cross
1611-14	The Descent from the Cross
1616	Madonna in a Garland of Flowers
1618	The Rape of the Daughters of Leucippus
1618	The Union of Earth and Water
1618	St. Sebastian
1635-38	The Judgment of Paris
1638	Andromeda
1638	The Little Fur (Helen Fourment, the Second Wife to the Artist)
1638-40	Bacchus
1640	The Three Graces

Michaelangelo *Buaonarroti* (1475-1564), (10 paintings)

1503-05	The Holy Family with the infant St. John the Baptist (the Doni tondo)
	The rest of the paintings are part of Sistine Chapel fresco:
1508-12	The Creation of Adam
1508-12	The Flood
1508-12	Adam and Eve
1508-12	Ignudi 1
1508-12	Ignudi 2
1508-12	Ignudi 3
1508-12	The Libyan Sibyl
1508-12	The Cumaean Sibyl
1508-12	The Delphic Sibyl

Jean-August-Dominique ***Ingres*** (1780-1867), (15 paintings)

1806 Marie-Francoise Beauregard, Madame Riviere
1807 Madame Antonia Devaucay de Nittis
1808 Bather of Valpincon
1811 Cecile Bochet, Madame Panckoucke
1812 Alix-Genevieve de Seytres-Caumont, Comtesse de Tournon
1814 La Grand Odalisque
1839 Odalisque and Slave
1845 Louise de Broglie, Countesse d'Haussonville
1848 Betty de Rothschild, Baronne de Rothschild
1851 Marie-Clothilde-Ines de Foucauld, Madame Moitessier
1853 Josephine-Eleonore-Marie-Pauline de Galard de Brassac de Bearn, Princesse de Broglie
1856 Marie-Clothilde-Ines de Foucauld, Madame Moitessier
1856 The Source
1859 Delphine Ramel, Madame Ingres
1862 Le Bain Turc

Sandro ***Botticelli*** (1445-1510), (11 paintings)

1482 Primavera
1482 Pallas and the Centaur
1483 Madonna with the Child (Madonna with the Book)
1485 The Birth of Venus
1485 The Madonna of the Magnificat
1485 Venus and Mars
1487 Madonna of the Pomegranate
1489 The Cestello Annunciation
1495 Calumny, detail of Truth and Remorse
1495 Lamentation over the Dead Christ