Semi-supervised Annotation of Brushwork in Paintings Domain using Serial Combinations of Multiple Experts

Marchenko Yelizaveta
National University, Singapore
marchenk@comp.nus.edu.sg

Chua Tat-Seng
National University, Singapore
chuats@comp.nus.edu.sg

Jain Ramesh
UC Irvine, USA
jain@ics.uci.edu

ABSTRACT

Many recent studies perform annotation of paintings based on brushwork. They model the brushwork indirectly as part of annotation of high-level artistic concepts such as artist name using low-level texture features and supervised inference methods. In this paper, we develop a framework for explicit annotation of paintings with brushwork classes. Brushwork classes serve as meta-level semantic concepts for artist names, paintings styles and periods of art and facilitate the incorporation of domain-specific ontologies. In particular, we employ the serial multi-expert framework with semi-supervised clustering methods to perform the annotation of brushwork patterns. Serial combination of multiple experts facilitates step-wise refinement of decisions based on the preferences of individual experts. Each individual expert performs focused subtasks using relevant feature set, which decreases the 'curse of dimensionality' and noise in the feature space. Each expert focuses on the annotation of the currently available samples from its unlabeled pool using semi-supervised agglomerative clustering. This approach is more appropriate as compared to the traditional classification methods since each brushwork class includes a variety of patterns and cannot be represented as a single distribution in the feature space. The experts exploit the distribution of unlabelled patterns and further minimize the annotation error. The multi-expert semi-supervised framework out-performs the conventional methods in annotation of patterns with brushwork classes. This framework will further be adopted to facilitate ontology-based annotation with higher-level semantic concepts such as the artist names, painting styles and periods of art.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis; I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems

General Terms

Algorithms, Design, Performance, Experimentation.

Keywords

Multiple experts, brushwork, painting annotation

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MM'06, October 23-27, 2006, Santa Barbara, California, USA.

Copyright 2006 ACM 1-59593-447-2/06/0010…$5.00.

1. INTRODUCTION

Visual characteristics of paintings such as color, brushwork, and composition concepts constitute a large body of expert analysis in the paintings domain [1]. The artists and art historicans formulated various heuristics and theories defining these concepts and utilized them for description of paintings. The color, brushwork and composition tightly relate to high-level semantic information of painting such as artist name, painting styles and fine art periods. Due to this, artistic concepts in the domain of paintings have been used for painting analysis to support applications such as brush-stroke detection, image annotation and retrieval and anti-fakery analysis etc [8, 12, 15, 17]. In particular, brushwork serves as an important cue to the identification of artist. Table 1 demonstrates the domain-specific knowledge on various brushwork classes. The brushwork classes in Table 1 refer to the techniques of depiction used by famous artists from Medieval to Modern periods of art. From Table 1, we can observe that brushwork classes vary with respect to paintings style and period of art.

Brushwork is important for the annotation of artistic concepts in the paintings domain. Several studies performed automatic brushwork analysis and utilized it for the annotation of paintings with artist names [8, 12]. Herik et al. [8] extracted color and texture features and performed classification of impressionistic paintings using neural network. The feature set used included color histograms in HSI and RGB color spaces, coefficients of the Fourier spectra, image intensity statistics and the Hurst coefficient. They reported classification accuracy of 80% with respect to artist names. Li et al [12] proposed the use of 2-D MHMMs to model brushwork in Chinese paintings for the purpose of annotating with artist names. They utilized the three-level pyramid of Daubechies coefficients as features and profiled each artist as a mixture of 2-D MHMMs. They reported an accuracy of between 58% to 80% depending on the number of mixture components in the model and on the subset of paintings used for the analysis. Both methods directly model the artist profile based on low-level features. Such approach has several drawbacks. First, it does not incorporate domain-specific knowledge for disambiguation of results. An example of an external domain-specific ontology is AAT [2] that offer wide range of artistic concepts for annotation. Second, since brushwork is not represented explicitly in such a framework, the introduction of other high-level concepts in arts domain will require additional training.

Most studies utilize a single classifier approach to assign labels in image annotation task. This approach is shown to be fruitful in many applications [5, 8]. However, it suffers from the 'curse of dimensionality' problem, which inevitably leads to the loss of accuracy [6]. To alleviate this problem, multi-expert frameworks have been proposed. Early work on expert combination mostly focused around 'multiple experts vs multiple levels' [7, 21]. Recent studies have shown that the use of multi-expert approaches could lead to higher accuracy as compared to the single classifier approach [10, 19]. Pudil et al. [19] proposed a serial (sequential) approach to expert combination. Generic approach to serial combinations of multiple classifiers has been discussed in [20]. Another possible way to increase the accuracy of inference methods is the use of semi-supervised techniques. These techniques exploit both labeled and unlabelled data within inference process. Such techniques are beneficial because in many circumstances the data density can provide good clues regarding what data points belong to what classes. Unlike the traditional classification methods, clustering techniques may incorporate both labeled and unlabelled instances within inference process and, thus, could facilitate semi-supervised inference. Existing methods for semi-supervised clustering fall into two general categories: constraint-based and distance-based. Constraint-based methods rely on the user-provided labels or relational constraints to guide the algorithm towards more appropriate data partitioning [22]. In the distance-based approaches, an existing clustering algorithm that uses a particular clustering distortion measure is employed, but the measure is trained to satisfy the labels or constraints in the given supervised data [11].

In this paper, we focus on the annotation of brushwork patches with respect to the brushwork classes. Explicit annotation of paintings with brushwork classes facilitates further identification of the artists (for example, "impasto by van Gogh", "impasto by Cezanne" etc.) and it serves as the meta-level knowledge for the domain-specific ontology. To our knowledge, this is first attempt to explicitly model artistic brushwork concepts for the purpose of further ontology-based annotation in the paintings domain. Due to the large variety of brushwork patterns, we utilize a number of statistical and signal processing features for the representation of brushwork contents. This leads to the problem of high-dimensional feature space. To perform annotation in such a space, we adopt a serial multi-level expert framework, where experts at each level utilize only a subset of features determined by the artistic domain knowledge. Within this framework, the decision process drives unlabelled patterns through the intermediate nodes (decisions) of the decision hierarchy to achieve the final annotation. We further extend this framework to perform semi-supervised annotation. Semi-supervised annotation is achieved at the level of individual experts associated with the decision tree nodes. The decision process of each individual expert represents constraint-based semi-supervised clustering. The expert performs clustering of the feature space and maximizes the number of partitions that satisfy the constraints. Based on such partitions, the expert refines the set of labels assigned to the unlabelled patterns and passes it to subsequent expert at the next level. The decision process continues until it reaches all terminal nodes of the decision tree.

The rest of this paper is organized as follows. Section 2 provides a brief discussion of the ontology-based annotation of paintings using artistic concepts. Section 3 describes features used to model brushwork patterns. Section 4 provides the background of serial combinations of multi-level experts. Section 5 describes the framework for serial combination of multiple experts for semi-supervised annotation of brushwork patches with brushwork classes. Finally, Section 6 presents the experimental results, followed by conclusions in Section 7.

2. ANNOTATION OF PAINTINGS

In our previous work [16] we introduced a generic framework for the annotation of paintings with artist names and painting styles. Within this framework, each painting is represented as a set of visual concepts referring to the artistic color, brushwork and composition concepts. These concepts represent visual level (atomic) concepts within our framework and serve as the basis for annotating higher-level concepts such as the artist names, periods of art and painting styles. These visual and high-level concepts are organized into a three-level ontology as demonstrated in Figure 1, which shows the organization of artistic concepts in visual, abstract and application-specific levels. Such approach has several advantages. First, the explicit assignment of visual-level concepts offers more flexibility to paintings annotation and retrieval. Second, it facilitates the introduction of domain-specific ontologies within the proposed framework, thus, enabling the use of ontology-based learning for concept disambiguation and propagation.

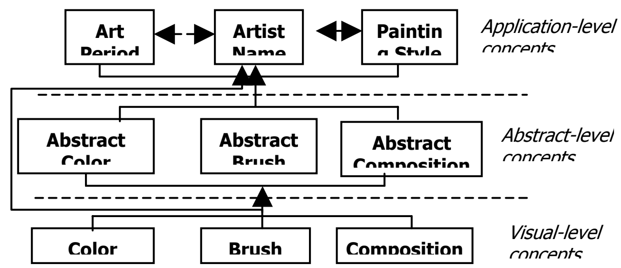
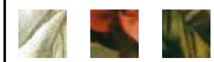
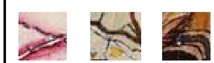
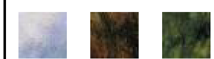
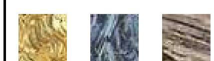
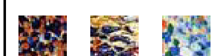


Figure 1. Three-level ontology of artistic concepts

Concepts at visual level (atomic concepts) include color, brushwork and composition concepts. They provide cues for the identification of abstract and application-level concepts such as color palette, brushwork technique etc. Abstract-level concepts include those concepts that are provided by artistic theories and mostly used by the art experts. This level includes concepts like "expressive", "harmony", "rational", "gestural" etc. In artistic theories, "expressive" means the usage of complimentary contrasting colors, while "rational" means brushwork techniques with very careful application of brush-strokes such as "divisionism", "pointillism" and "shading" [1]. Application-level concepts include artist name, painting style and period of art concepts. Double-edged arrows show that these concepts are inter-connected. They are designed for the novice users.

Within our framework, brushwork is of importance for artist annotation. Indeed, with brushwork patches classified as, for example, "impasto" class, we can further subdivide them into "impasto by Van Gogh", "impasto by Sezanne" etc. These sub-categories readily serve as the basis for annotating with artist name. Similarly, the use of specific brushwork classes implies the painting style and the period of art. For example, "glazing" and "shading" brushwork classes are widely used in Medieval Art, while "pointillism" class is never used in this period. Table 1 lists the brushwork classes that are used in our framework together with brief arts background.

Table 1. Brushwork classes used for annotation

Class	Background	Characteristics	Low level features	Examples
Shading	Depiction of foldings in Medieval Period	Edges and gradients, often directional, intensity contrast, weakly or non-homogeneous	Multiscale Gabor texture features, Zernike moments, Chiaroscuro (intensity) color contrast, Multiscale Fractal Dimension, Lacunarity	
Glazing	Depiction of nudity/face in Medieval Period	Subset of hues (yellow, red, orange), intensity contrast, gradients, non-homogeneous, may contain edges	Top major colors with account for the perceptual similarity, Chiaroscuro (intensity) color contrast, Daubichies Wavelet Transform, Zernike moments, Multiscale Fractal Dimension, Lacunarity	
Mezzapasta	Widely used technique in paintings. The color palette used varies with respect to the art period.	Homogeneous, low intensity contrast and small gradients	Mean and Deviation of image magnitudes, Directional Histogram of Gradient Magnitudes, Major colors with account for perceptual similarity	
Grattage	Depiction of objects and patterns in Fauvism and Expressionism painting styles of Modern Art period	Edges, high gradients, intensity contrast, inhomogeneous	Number of Edge Pixels, Mean and Deviation of Directional Edge Histogram, Chiaroscuro (intensity) color contrast, Daubichies Wavelet Transform, Multiscale Fractal Dimension, Lacunarity	
Scumbling	Depiction of sky, clouds, greenery and atmosphere in Fauvism, Impressionism, Post-impressionism and Pointillism painting styles of Modern Art	Soft gradients, low intensity and hue contrast, low directionality, weakly homogeneous	Daubichies Wavelet Transform, Zernike moments, Chiaroscuro (intensity) and Complimentary (hue) color contrast, Multiscale Fractal Dimension, Lacunarity	
Impasto	Widely used in Impressionism, Post-impressionism, Pointillism	Edges, high gradients, often directional, low hue contrast, high intensity contrast	Number of Edge Pixels, Directional Histogram of Gradient Magnitudes, Chiaroscuro (intensity), Complimentary (hue) color contrast, Daubichies Wavelet Transform, Multiscale Gabor texture features	
Pointillism	Often used for depiction of atmosphere/air in Pointillism painting style	Medium intensity contrast, medium roughness, no directionality, homogeneous	Mean and Deviation of Magnitude, Chiaroscuro (intensity) color contrast, Daubichies Wavelet Transform, Zernike Moments	
Divisionism	Widely used in Pointillism, demonstrates the Color Mixing Principle	High gradients, high roughness, high intensity and hue contrast, no directionality, weakly homogeneous	Mean and Deviation of Magnitude, Daubechies Wavelet Transform , Chiaroscuro (intensity) and Complimentary (hue) color contrast, Multiscale Fractal Dimension, Lacunarity	

3. BRUSHWORK ANALYSIS

There are several approaches to analyze brushwork. Meltzer et al. [18] have developed methods for the explicit detection of brush-strokes for further identification of artists. Their method made a number of assumptions like the high intensity around brush-stroke, and therefore is not suitable in our case. Moreover, explicit brush-stroke detection requires a controlled high-resolution collection.

Other research [8,12] adopted texture-based representation and analysis of brushwork patches. These approaches are not so computationally expensive and can be used for the collections downloaded from the Web like in our case.

Table 1 demonstrates that our collection includes a vast number of patterns. The patterns are mostly stochastic. They exhibit a variety of properties such as directional (for example, “impasto”), non-directional (“pointillism”), contrasting (“divisionism”) and smooth (“mezzapasta”). In terms of the spatial homogeneity we can roughly group brushwork patterns as homogeneous (“mezzapasta” and “pointillism”), weakly homogeneous (“divisionism”) and inhomogeneous (“scumbling”, “shading” and “glazing”).

Various comparative studies showed that no single texture features representation approach performs best for all kinds of textures. Hence, to capture the variety of patterns in our dataset, we need to utilize various signal-based and statistical texture feature representations. As Table 1 demonstrates we utilize color and texture features for pattern representation. To calculate color features, we utilize CIE L*u*v color space. From color histogram, we extract major colors with account for their perceptual similarity [4]. We calculate complimentary and chiaroscuro color contrasts based on our previously developed method [17].

In order to model the variety of brushwork patterns, we use several texture features. First, we make use of the edge-based features to capture linear components of a pattern. We apply Canny edge detector [3] with fixed threshold to the whole collection and calculate directional histogram:

$$EdgeHist = \frac{P_i}{\sum_i P_i} \tag{1}$$

where P_i denotes the number of edge pixels in the *i-th* direction. Next we extract the gradient-based features. These are statistics of image gradients (mean and deviation) and their directional histogram. We calculate the directional gradient histogram using the formulae above. For both histograms, we employ eight directions.

For representing the directional characteristics, we utilize multi-scale Gabor Transform proposed for the image retrieval by Majunath et al. [15]. A Gabor filter bank is a pseudo-wavelet filter bank where each filter generates a near-independent estimate of the local frequency content. Gabor filter acts as a local band-pass filter with certain optimal joint localization properties in the spatial domain and spatial frequency domain. To extract the Gabor features, the input image *I(x, y)* is convolved with a set of Gabor filters of different orientations and spatial frequencies that cover appropriately the spatial frequency domain. In our experiments, we utilize 8 orientations and 4 scales. The general functional *g(x;y)* of the two-dimensional Gabor filter family can be represented as a Gaussian function modulated by an oriented complex sinusoidal signal:

$$g(x,y)=\frac{1}{2\pi\sigma_x\sigma_y}\exp[-\frac{1}{2}(\frac{\tilde{x}^2}{\sigma_x{}^2}+\frac{\tilde{y}^2}{\sigma_y{}^2})+2\pi jW_{\tilde{x}}] \tag{2}$$

where σ denotes the scaling parameters of the filter with respect to x and y, W is the center frequency, and θ determines the orientation of the filter.

Another important texture feature is the Dyadic Wavelet Transform (DWT). DWT is most useful for multi-resolution image analysis and captures a variety of texture properties [13]. Dyadic wavelet decomposition is carried out using 2 channel filter banks composed of a low-pass (G) and a high-pass (H) filter and each filter bank is sampled at a half rate (1/2 down sampling) of the previous frequency. We employ Daubechies filter banks for our study. This filter bank has the important qualities of orthogonality and compact support.

To extract texture features from Gabor and Daubechies filter response, we calculate the mean and deviation of energy distribution of the transform coefficients for each sub-band at each decomposition level. Let the image sub-band of size NxN be $I_i(x, y)$ with i denoting the specific sub-band, then the resulting feature vector obtained from the filter response is $f\{\mu_i,\sigma_i\}$ with,

$$\begin{aligned}\mu_i&=\frac{1}{N^2}\sum_{x=1}^{N}\sum_{y=1}^{N}\left|I_i(x,y)\right|\\ \sigma_i&=\frac{1}{N^2}\sum_{x=1}^{N}\sum_{y=1}^{N}\left|I_i(x,y)-\mu_i\right|\end{aligned} \tag{3}$$

The major drawback of energy-based features above is the implicit assumption of texture homogeneity. Such assumption does not hold for several classes of brushwork in our dataset that include non-regular textures (for example, “scumbling” and “shading”).

To represent non-regular textures, Mandelbrot [14] popularized the self-similar fractional Brownian motion (fBm) model, which is characterized by a single parameter known as the Hurst parameter. The Hurst parameter controls the visual roughness of the process at all scales. In our study, we utilize the extended self-similar (ESS) model [9] that measures the Hurst Parameter at various scales and, thus, encodes more detailed textural information. First, ESS model calculates the directed increments (in x and y orientation) of dyadic scales for an image *I(x,y)*:

$$\begin{aligned}\Delta_s^{Xaxis}(x,y)&=I(x+2^s,y)-I(x,y)\\ \Delta_s^{Yaxis}(x,y)&=I(x,y+2^s)-I(x,y)\end{aligned} \tag{4}$$

The structure function is defined as the average of the incremental power over all available pixels:

$$f_s^{\theta}=\frac{1}{N(N-2^s)}\sum_x\sum_y|\Delta_s^{\theta}(x,y)|^2 \tag{5}$$

for $\theta = \{X_{axis}, Y_{axis}\}$. The multi-scale Hurst parameters are computed for scale s to obtain the isotropic and directed features as follows:

$$\begin{aligned}H_s(x,y)&=\frac{1}{2}\log_2(\frac{f_{s+1}^{Xaxis}+f_{s+1}^{Yaxis}}{f_s^{Xaxis}+f_s^{Yaxis}})\\ H_s^{\theta}(x,y)&=\frac{1}{2}\log_2(\frac{f_{s+1}^{\theta}}{f_s^{\theta}})\end{aligned} \tag{6}$$

Finally, we utilize statistical moment descriptors to extract shape information from the brushwork patches. We employ these features to represenr “glazing”, “shading” and “scumbling” classes. We further utilize region-based approach for shape description that makes use of all the pixel information across the patch and does not require the shape boundary. Teague [23] first introduced the use of Zernike moments to overcome the shortcomings of information redundancy present in the popular geometric moments. Zernike moments have property of orthogonality and have been shown effective in terms of the image representation. Zhang et al. [25] demonstrated that Zernike moments out-perform geometrical moments in shape retrieval task. Another important property of Zernike moments is that they are rotation invariant and can be easily constructed to an arbitrary order. The Zernike polynomials are a set of complex, orthogonal polynomials defined over the interior of a unit circle $x^2+y^2=1$ as:

$$\begin{aligned}V_{nm}(x,y)&=V_{nm}(r,\theta)=R_{nm}(r)e^{jn\theta}\\ R_{nm}(r)&=\sum_{s=0}^{\frac{n-|m|}{2}}(-1)^s\frac{(n-s)!}{s!\left(\frac{n+|m|}{2}-s\right)!\left(\frac{n-|m|}{2}-s\right)}r^{n-2s}\end{aligned} \tag{7}$$

where n is non-negative integer, m is the number such that n-$|m|$ is even and $m \leq n$, $r=sqrt(x^2+y^2)$ and $\theta=tan^{-1}(x/y)$. The magnitude of Zernike moments has the property of rotational invariance and is defined as:

$$A_{nm}=\left|\frac{n+1}{\pi}\sum_x\sum_y I(x,y)*V_{nm}(x,y)\right| \tag{8}$$

where $x^2+y^2\leq 1$ and * denotes the complex conjugate. For our task, we calculate 32 Zernike moments.

To perform the annotation task, we represent brushwork as the set of mutually exclusive classes. Thus, each pattern in our dataset belongs to only one class of brushwork. However several properties of brushwork significantly complicate the annotation process. First, brushwork patches often bear some resemblance to each other. For example, “divisionism” can be quite similar to “impasto” brushwork class. Next, brushwork significantly varies in the areas along the object borders and areas of minor details. Next, each brushwork class includes a variety of patterns that might not be close to each other in the feature space. Lastly, our collection includes paintings captured under varying lighting conditions and this introduces additional difficulty.

We employ all of the above features for adequate representation of brushwork patterns. This yields high-dimensionality of the feature space, leading to the 'curse of dimensionality'. It essentially means that the sparseness of data increases exponentially with the dimensionality of the input space given a constant amount of data, with points tending to become equidistant from one another at a certain high dimension [6]. This will degrade the quality of the traditional inference methods. To tackle this problem, we adopt a framework that combines several experts, each of which assigns candidate classes to the unlabelled patterns based on a subset of features. Next section describes the multi-expert framework.

4. MULTIPLE SERIAL EXPERTS FOR ANNOTATION

The multi-expert approaches are beneficial, because the combination of targeted experts aimed at focused subtasks often results in higher accuracy and require smaller datasets as compared to single-expert approaches. Also, such approaches can be more natural for applications with pre-defined subtasks and known relationships among subtasks [20]. Multi-expert approaches facilitate dimensionality reduction for the domains where relationships between the features and the properties of subtasks are known. Configurations that combine experts in several sequential levels are called serial configurations. The main attraction of the serial approaches for decision combination is that these configurations implement a step-wise decision-making procedure [19]. After the first intermediate decisions are taken at the preliminary level in the decision hierarchy, the final decision is reached through a step-wise refinement procedure. As the decision hierarchy is traversed in the forward direction, the decisions of individual experts become more and more refined, and the confidence associated with the decision increases.

In our work we utilize the decision tree configuration of multiple experts that consists of a root-node, a number of non-terminal nodes and a number of terminal nodes. Associated with the root node is the entire set of classes into which a pattern may be classified. A non-terminal node represents an intermediate decision and its immediate descendant nodes represent the decisions originating from that particular node. The decision making process terminates at a terminal node. The unlabelled patterns found in the terminal node receive the class label associated with that particular terminal node. Each level of the decision hierarchy includes several experts that operate simultaneously and independently of each other. The decision process sequentially traverses the tree in top-bottom sequential fashion. Figure 2 demonstrates the decision hierarchy that incorporates these ideas.

With multi-level approach, we progressively reduce the subset of classes to which a pattern might belong at each level of the decision hierarchy. We denote the subset of candidate classes *the target size*. The reduction of the target size is performed by individual experts. Each node of the decision hierarchy in Figure 2 is associated with an individual expert. Each expert receives the subset of data and its target size from the ancestor node at the previous level. Individual expert splits its respective dataset in accordance to the decision hierarchy, thus reducing the target size and refining the ambiguous decisions.

We formalize the reduction of target size as follows. The expert at the *i-th* level has the input vector (X,S_{i-1}) and generates the output set S_i, where X represents a pattern. S_i represents the set of classes to which the expert of *i-th* level believes the pattern X might belong and set S_i is a subset of its respective set S_{i-1} $(S_n \subset S_{n-1} \subset S_i \ldots \subset S_0)$ During the classification process, if the terminal node is reached, then patterns under this node are labeled with a single element of S_i.

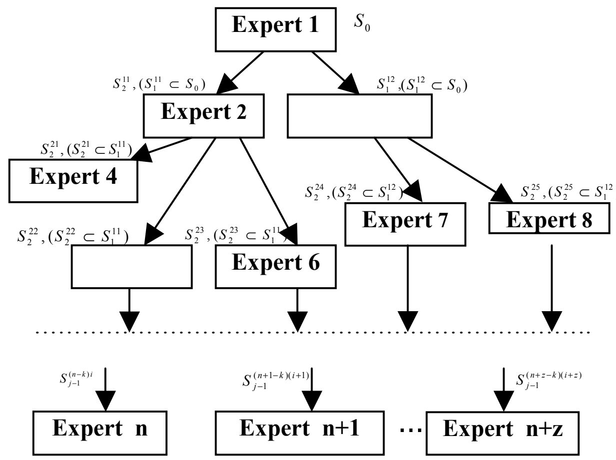


Figure 2. Serial Combination of Multiple Experts

The performance of the serial multi-expert approaches relies on the performance of individual experts. Suppose we denote the performance function $f_i(P_c,P_e)$ of the *i-th* expert in terms of the probability of giving the correct and erroneous assignments respectively (P_c and P_e). Given the performance function $f_T(P_c,P_e)$ of the terminal nodes of the combined configuration, the higher recognition rate of the combined system with respect to each individual expert is achieved if $f_i(P_c,P_e) < f_T(P_c,P_e)$. Since the serial expert approach sequentially refines its decisions, then the multi-expert configuration cannot exceed the performance of its terminal nodes, provided that all experts operate on the same feature space and dataset. Therefore, the final performance can be either lower or identical to the performance of terminal nodes if all the experts utilize the same feature. However, if the experts operate on different subsets of feature, and their corresponding feature subsets are relevant to their respective sub-goals, then the sparseness and noise of the feature space from the point of view of the expert can be reduced. This results in the increase in expert's accuracy and the overall accuracy of the combined system is expected to be better.

Another important issue in the overall system's performance is the order and optimal number of the experts. It can be observed that the performance of the subsequent levels of experts depends on the performance of the pervious levels. Hence, the order of the experts influences the overall system performance. As the introduction of each additional expert may increase the overall error, it is thus important to employ an optimal number of experts such that the increase of performance achieved by incremental enhancement does not diminish as more experts are combined. In Section 5, we define the decision hierarchy and individual experts used in annotation of brushwork patterns.

There exist two major strategies for the serial multi-expert approaches [20]. In Class Set Reduction strategy the target size is reduced continuously when the decision process traverses the decision hierarchy. This strategy requires the decision process to carry unlabelled samples until the terminal nodes are reached. We provide details of the Class Reduction strategy in Section 4.1. In contrast, Class Reevaluation strategy is concerned only with patterns that are rejected and recognized with low confidence. In accordance to this strategy, the decision process does not need to carry the unlabelled instances if they are recognized with high degree of confidence at the intermediate nodes. We will discuss the Class Reevaluation strategy in Section 4.2.

4.1 Class Set Reduction strategy

The Class Set Reduction strategy requires that the experts generate a subset of class indices from the original set of class indices received from the previous layer. There are two sources of information for any expert, the first being the current unlabelled patterns itself, and the other being the list of candidate classes passed on by the ancestor expert. The candidate class labels reflect the choice of the previous expert in identifying the current set of unlabelled patterns. Thus, the expert at *i-th* level of the decision hierarchy needs to produce a candidate class subset S_i of its own preferences as a function of each unlabelled pattern *X*. The subset S_i should have a high probability of containing a true label among the candidate class labels corresponding to the patterns. Naturally, this condition will greatly depend on the performance of individual experts within the decision hierarchy. Assuming:

- $w(X)$ is the true class of pattern *X*,
- $d(X,S_i)$ is the candidate class generated by the current expert,
- P_{ei} is the probability that S_i does not contain true class, $P_{ei}=P[w(X)\not\in S_i]$,
- P_{ci} is the probability that S_i contains true class, $P_{ci}=P[w(X)\in S_i]$,,
- $P_{e(i+1)}$ is the probability that the expert at $(i+1)$ level assigns *X* to the wrong class, although S_i contains the true class label $P_{e(i+1)}= P[d(X,S_i) \neq w(X) \mid w(X)\in S_i)]$,
- $P_{c(i+1)}$the probability that the expert at $(i + 1)$ level assigns *X* to the correct class, given that S_i contains true class index
- $P_{ei+1}= P[d(X,S_i) = w(X) \mid w(X)\in S_i)]$.

Then the overall correct classification of n-level serial network is

$$P_{cT} = P_{c1} \times P_2 \times ... \times P_{cn} \tag{9}$$

and the overall error of n-level serial network is

$$\begin{aligned} P_{eT} &= (P_{e2} + P_{e1} \times P_{c2}) + (P_{e3} + P_{c1} \times P_{e2} \times P_{c3}) + \\ &+ .... + (P_{en} + P_{c1} \times P_{c2} \times P_{c3} \times P_{e(n-1)} \times P_{cn}) \end{aligned} \tag{10}$$

One important aspect of the Class Reduction strategy is the assumption of zero rejection rate of the experts associated with the nodes. This contradicts the conventional inference methods, where a pattern can be rejected due to the low confidence. The rejection rate is defined as:

$$R = \frac{N_{low\_conf}}{N_{all}} \tag{11}$$

where N_{low_conf} denotes the number of patterns that are assigned candidate classes with the confidence lower then some predefined threshold (t_{reject}), N_{all} denotes the number of input patterns for the expert. Naturally, the rejection rate contributes towards the overall error. With the incorporation of the rejection capacity within an expert, the accuracy of recognition increases, but the overall number of recognized patterns decreases.

In the Class Set Reduction strategy, the ability to pass samples to the next level is far more important than the absolute recognition rate, since it increases the chance of an unlabelled pattern being assigned the true label. Due to this, we employ the individual experts without rejection capacity for this strategy.

4.2 Class Reevaluation strategy

The Class Reevaluation strategy extends the intermediate nodes to facilitate additional analysis: if the unlabelled patterns are assigned labels with high confidence, then these assignments become final and the decision process does not evaluate these patterns further. In essence, this strategy incorporates rejection capability and reevaluates patterns that are assigned with confidence lower than some predefined threshold (t_{accept}). Such strategy requires the individual experts to perform recognition with respect to all classes, and pass the patterns with ambiguous assignments to the next level.

We now formalize the decision process for unlabelled pattern *X*. Assuming:

- $w(X)$ is the original class associated with the current pattern,
- $d(X, t_{accept})$ denotes the candidate class of pattern *X* generated by the current expert based on the confidence threshold ,
- α denotes the confidence of expert in assigning a candidate class to pattern *X*,
- P_{ci} is the probability that expert generates the true class, $P_{ci} = P[d(X,t_{accept}) = w(X)]$,
- P_{ei} is the probability that expert doesn't generate true class. We define $P_{ei} = P_{error} + P_{rejection}$, where
- $P_{error} = P[d(X,t_{accept}) \neq w(X) \mid (\alpha > t_{accept})]$ denotes the probability of erroneous class label assigned to the unlabelled pattern *X* with confidence α higher then threshold t_{accept}, and
- $P_{rejection} = P[d(X,t_{accept}) = w(X) \mid (\alpha < t_{accept})]$ denotes the probability of the correct class label assigned and being rejected due to the confidence lower then the threshold,

Similarly to the Class Reduction strategy, the probability of correct decision is defined as:

$$P_{cT} = P_{c1} \times P_2 \times ... \times P_{cn} \tag{12}$$

with the errors given by $P_{eT}=1-P_{cT}$.

4.3 Individual Experts

Several machine learning methods can be used for individual experts. Currently the majority of inference methods are based on the two-step supervised inductive process. During the inductive step, the expert is trained based on the labeled data. During the deductive step, the expert predicts the labels of the unlabelled data. One possible drawback of such inference methods is that they do not readily incorporate unlabelled samples at the training stage and thus do not account for the distribution of unlabelled samples. In semi-supervised case, expert observes both labeled and unlabeled patterns, results in tighter upper error bound as compared to the supervised approach [24].

From the examples given in Table 1, each class includes a variety of patterns. These patterns form clusters of certain brushwork class used by the same artist, in the same painting etc. However, such clusters can be far from each other in the feature space. Therefore, it might be difficult for the conventional inference methods to learn single discriminate function to recognize such classes. Moreover, the traditional classification methods require the labeled data set for training and do not facilitate semi-supervised learning. On the other hand, clustering techniques readily offer a basis for modeling a class as a set of clusters, they do not require clusters to be spatially close to each other in the feature space and they can be easily extended to incorporate unlabelled patterns and facilitate semi-supervised annotation. In the next section we discuss the semi-supervised annotation of brushwork patterns.

5. ANNOTATION WITH BRUSHWORK TECHNIQUE CONCEPTS

5.1 Decision hierarchy

In this section, we combine the serial multi-expert approach with semi-supervised annotation techniques and apply it to the task of brushwork class annotation. To perform the annotation, we traverse the decision hierarchy from coarser to finer decisions. This approach bears a lot of resemblance with multi-class annotation task, where some classes are more similar to each other than the others. In this situation, it is easier for an expert to recognize the most dissimilar or distinctive subsets of classes and then carry on and resolve the ambiguities in each of the subsets to achieve the final annotation.

In our task we know apriori the characteristics of the brushwork classes. For example, “pointillism” and “mezzapasta” classes are weakly homogeneous patterns with decreasing levels of roughness. We rely on such characteristics to formulate the sub-goals at the intermediate and terminal nodes.

In the serial multi-expert configurations, the sub-goals can be identified and combined in arbitrary number of levels. Rahman et al. [20] demonstrated experimentally that the two-level configurations produce very good results. In our study, we employ the three-level hierarchy of decisions with the single brushwork class corresponding to the terminal node. Figure 3 represents the decision hierarchy for the brushwork annotation. For the successful annotation process it is important that the experts generate their decisions based on the relevant subset of features. Figure 3 shows the features used by each of the experts in non-shaded boxes. Intermediate decisions are found in the first two levels of the hierarchy, while the last level includes only terminal nodes. Dotted lines in Figure 3 distinguish three levels of the decision hierarchy.

The decision process starts with all classes and the original dataset. At the first level, we arrange the brushwork classes in the subsets based on the degree to which they exhibit similar linear components. We define the three sub-goals as {“impasto”, “grattage” and “devisionism”}, followed by {“scumbling”, “glazing” and “shading”} and, finally, {“pointillism” and “mezzapasta”}. The expert associated with the top-level decision utilizes such features as the Gabor transform features, edge-based and gradient-based directional histograms and CIE L*u*v values of the major colors.

At the next level, there are three experts working simultaneously on their respective datasets. The first second-level expert aims to split “grattage” class from “impasto” and “divisionism” classes. Since patterns in “grattage” class exhibit long edges and high chiaroscuro contrast, we employ edge information and color contrasts with CIE L*u*v values of the major colors. The second expert assesses its input patterns by roughness. This leads to the terminal node “shading”, since this class exhibits more roughness as compared to “scumbling” and “glazing”. The expert uses multi-scale fractal dimension features, since all of the analyzed classes include non-homogeneous patterns. In addition, it employs chiaroscuro contrast because the patterns of “shading” class often exhibit light and dark colors. The third expert analyzes the patterns belonging to only two classes, and hence produces the terminal nodes for “mezzapasta” and “pointillism”. Patterns in these two classes vary with respect to the roughness and the number of colors they exhibit, and thus we employ multi-scale fractal dimension and color histogram statistics to recognize these two classes.

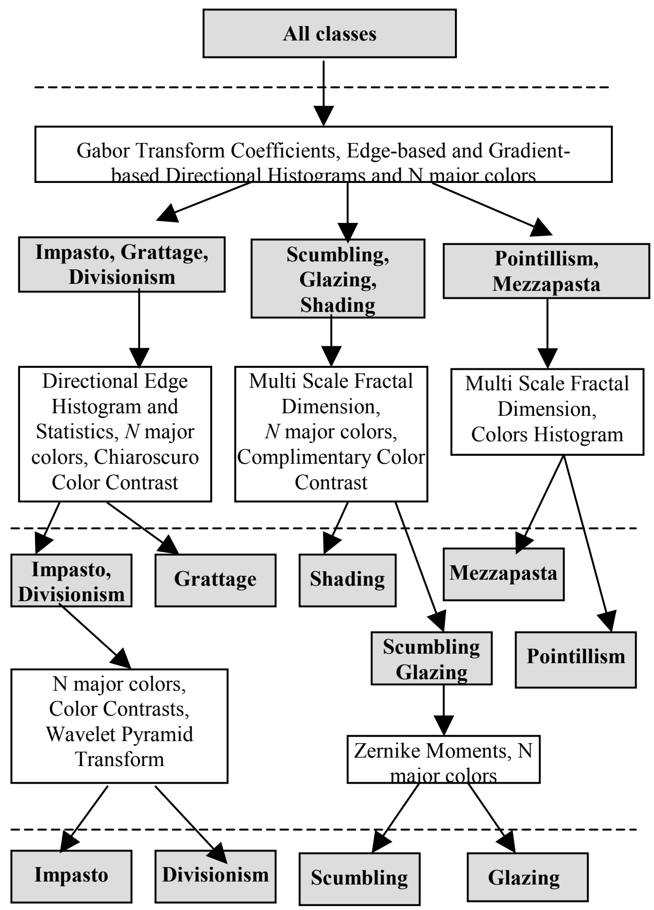


Figure 3. The decision hierarchy for brushwork annotation

At the third level, the decision hierarchy has only two unresolved intermediate nodes: {“impasto”, “divisionism”} and {“scumbling” “glazing”}. To disambiguate “impasto” and “divisionism” classes, we employ the color information with Wavelet Pyramid Transform (WPT), which uses the Daubechies filter bank to detect the linear component and a variety of other texture properties. This representation is suitable to recognize class “impasto”, since its patterns exhibits a large number of linear components in several directions. In addition, the expert employs complimentary color contrast because "divisionism" patterns exhibit more colors of various hues as compared to "impasto" patterns. The last expert annotates the patterns with "scumbling" or "glazing" classes. While "scumbling" class exhibits large a number of soft gradients due to the spatially close brushworks of varying intensity, "glazing" exhibits more homogeneous surface with very long soft gradients, if any. We use Zernike moments that model the surface of a patch to recognize these two classes. Also, the expert employs CIE L*u*v values of major colors for this sub-goal, since the patterns of class "glazing" quite often exhibit reddish, pinkish and nude-like colors.

In the decision hierarchy above, we pre-define sub-goals that refine annotation process at each level based on the domain knowledge and characteristics of the brushwork patterns. Apart from the decision hierarchy structure, performance of the multi-expert configurations depends on the implementation of its individual experts. In the next two sub-sections we discuss the annotation process of individual experts.

5.2 Individual Experts

We utilize agglomerative clustering and iterative K-means clustering to perform assignment of labels by each expert. For both clustering methods, *cluster purity* serves as the basis for annotation of unlabelled patterns. We define cluster as "pure" if it more than 85% of its labeled patterns belong to class (or a subset of classes) *X*. Cluster purity represents the degree to which the cluster contains patterns of class *X* and is defined as:

$$\rho(c) = \frac{N_X}{N_{all}} \tag{13}$$

where N_X denotes the number of labeled patterns from the class (or a subset of classes) *X* within cluster *c*, N_{all} denotes number of patterns in this cluster. Since in semi-supervised approach clusters include both labeled and unlabelled data, pure clusters represent a decision as to which class or a subset of classes the unlabelled pattern is likely to belong to.

In case of iterative K-means clustering, we increase the number of clusters at each iteration and attempt to maximize the following function:

$$\varphi = \sum_{i=1}^{d} N_i^p \times d \tag{14}$$

where N_i^p denotes the number of patterns in the *i-th* pure cluster and *d* is the average size of pure clusters at each iteration. In case of the agglomerative clustering, we build a cluster tree and search it in a top-down fashion for pure clusters. Similarly to K-means clustering, we attempt to maximize function φ given in the formulae above.

In the multi-expert framework, information flow among the experts is predefined within the decision hierarchy. Each *mn-th* expert at *j-th* level receives the subset of classes and associated patterns and further splits it into several pre-defined subsets of classes $S_j^{m(k+1)}, S_j^{m(k+2)} \ldots S_j^{m(k+i)}$, where *m* is the unique identifier of the expert, *(k+1), .., (k+i)* denote the unique identifier of the generated candidate list. During the clustering process, the expert uses each of the subsets $S_j^{m(k+1)}, S_j^{m(k+2)} \ldots S_j^{m(k+i)}$ to guide its search for pure clusters. In case of the terminal node, the expert searches for the pure clusters that contain patterns of a single class, and all unlabelled patterns found in this cluster will receive its label. In essence, the subsets $S_j^{m(k+1)}, S_j^{m(k+2)} \ldots S_j^{m(k+i)}$ represent pair-wise constraints in terms of "can" and "can not" links, specifying which labels could be grouped together.

5.2.1 *Class Reduction strategy*

Class Reduction strategy assumes no rejection rate and, thus, requires that each expert should assign the subset of classes to each unlabelled pattern. In this strategy, the fact that all samples are observed is more important than the absolute recognition rate at each level. In case of the rejected patterns, the expert passes them to all intermediate experts of the subsequent level. The experts of the subsequent level evaluate each rejected pattern and generate the confidence value for it. Based on this value, the patterns is either accepted or rejected again. If pattern is accepted, the decision process includes it in the input dataset for the next corresponding expert. If the pattern is rejected, the above process repeats until the terminal nodes are reached. At the terminal nodes, the labels of the patterns are assigned based on the highest confidence value. In case of K-means clustering, the expert searches for the closest pure clusters for unlabelled patterns and generates the best possible confidence value for them. In case of agglomerative clustering, the expert traverses its cluster tree in the bottom-up fashion. The cluster tree contains the paths connecting unlabelled patterns to the clusters, which they originate from. The expert searches these paths, trying to find clusters with the higher degree of purity and generates the highest possible confidence values. The decision process maximizes the confidence values from the terminal nodes and assigns unlabelled patterns their respective classes of the highest confidence.

5.2.2 *Class Reevaluation strategy*

In contrast to the Class Reduction strategy, Class Reevaluation strategy allows rejection threshold within each individual expert. In this case, rejected patterns are passed for reevaluation to the next level. With the Class Reevaluation strategy, the serial multi-expert framework assigns labels to unlabelled patterns in the intermediate nodes if the system has high confidence in it. The use of clustering methods provides the basis for such a conditional assignment. Lets consider the cluster space *C* with *k* clusters $\{C_i\}$ for $i = 1 \ldots k$. In this space, we have several cluster types $C = C^p \cup C^{np} \cup C^{nl}$, where C^{nl} denotes clusters that include testing samples only and, thus, carry no class information (labels), C^{np} and C^p denote clusters that include testing and training samples and, consequently, represent some mixture of labels. However, clusters in C^p are pure and mostly exhibit labels from the certain predefined subset. Further, C^p includes clusters that contain majority of labels of a single class. These clusters demonstrate high confidence of the expert in its assignment. In accordance to Class Reevaluation strategy, the decision process assigns the unlabelled patterns found in these clusters with the corresponding labels. These labels represent the final decision and are excluded from further analysis.

6. EXPERIMENT RESULTS

In this section we demonstrate the performance of our framework. For our experiments, we extract 4880 patches of size 32x32 from 30 paintings of the painting styles: Renaissance, Fauvism, Impressionism, Post-Impressionism, Expressionism and Pointillism. We randomly select 75% of the dataset for training and use the remaining patterns for testing. Figure 4 demonstrates the distribution of brushwork classes in the training and testing sets.

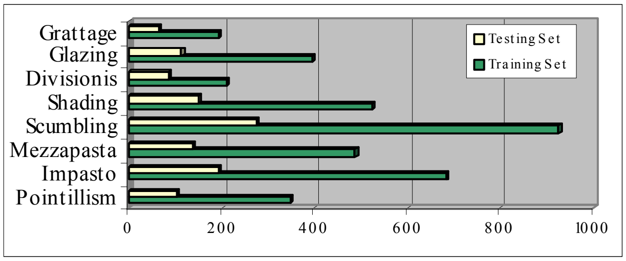


Figure 4. Distribution of the class labels in the dataset

In our experiments, we test the multi-level serial framework with respect to the Class Reduction and Class Reevaluation strategy. Within each strategy, we test the framework performance based on the several clustering techniques implemented for individual experts. These techniques include: 'complete-link', 'average-link', 'single-link' agglomerative clustering and iterative K-means clustering.

Table 2. Overall annotation

Framework	Kmeans	Complete	Average	Single
Baseline	**74.6%**	**74.73%**	**75.08%**	**57.64%**
Multi-expert, Class Reevaluation	**87.2%**	**87.5%**	**88.15%**	**60.47%**
Multi-expert, Class Reduction	**91.4%**	**92.06%**	**94.89%**	**68.32%**

We compare the performance of these systems with the baseline system, which is a one-step semi-supervised annotation method that employs the full feature set. It performs the annotation of the unlabelled instances into the brushwork classes on the basis of the pure clusters. It can be viewed as a single expert operating on the full feature set. The baseline system employs the same set of clustering techniques as above-mentioned to achieve pure clusters. Table 2 demonstrates the performance of the multi-expert and baseline systems in terms of the overall annotation accuracy.

Both the baseline and multi-level frameworks obtain significantly higher accuracy using 'complete-link', 'average-link' and iterative K-means techniques as compared to the 'single-link' method. Since the 'single-link' method merges two clusters based on the smallest minimum pair-wise distance, it tends to group together patterns of the different classes, leading to a large number of impure clusters.

Both Class Reduction and Class Reevaluation strategies result in better accuracy as compared to the baseline methods. This improvement results from the sequential refinments of class labels as well as from the use of relevant features at the level of individual experts. Figure 5 demonstrates how terminal nodes benefit from the disambiguation process. Here, the task of the expert assosiated with the current terminal node is to assign the input patterns to one of the two classes ("divisionism" or "impasto"). The *Input Set* in Figure 5 is the set of the unlebelled patterns given as the input to the current terminal node. It represents the coarse decision of the ancestor node. From the point of view of the current expert assosiated with the terminal node, these unlabelled patters are likely to be "divisionism" or "impasto". Figure 5 demonstrates distribution of the unlabelled patterns in the *Input Set* with respect to their true labels (*Y* axis). It can be seen that candidate class labels ("divisionism" or "impasto") include the true class label for the majority of unlabelled patterns. Based on the input patterns, current expert generates its own decision. It output the *Output Set 1* ("impasto") and *Output Set 2* ("divisionism"). Figure 5 demonstrates distribution of the patterns in *Output Set 1* and *Output Set 2* with respect to their true patterns.

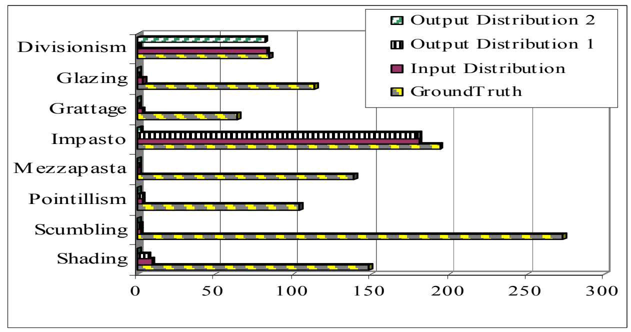


Figure 5. Example of the terminal node

From the distribution of unlabelled patterns in the *Input Set,* it is clear that the sequential refinement disambiguates patterns before they reach the current expert (terminal node) and receive their final label. This refinement naturally leads to higher accuracy achieved by the individual experts since the probability of the true class being assigned to the disambiguated patterns is high, resulting in better performance of the multi-expert framework.

Table 2 shows that the performance of Class Reevaluation strategy is worse than that of the Class Reduction strategy. This is because for the Class Reduction strategy, some patterns receive their final labels at the intermediate nodes. Such conditional assignments result from high confidence of the experts at these nodes. However, the decision process annotates such patterns at the level of coarse intermediate decisions and disambiguation of these patterns is only partial, which results in additional 5% to 6% erroneous labels as compared to the Class Reduction Strategy.

Figure 6 examines the distribution of annotation error of the multi-level framework with respect to the brushwork classes.

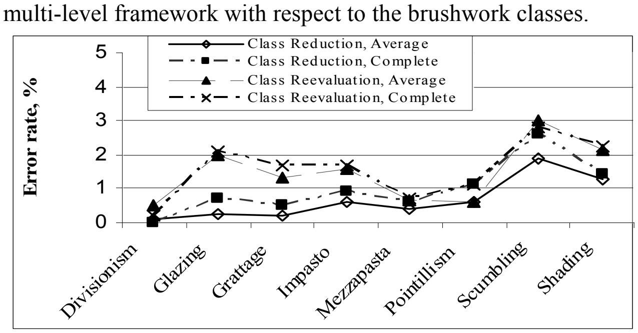
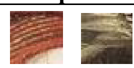
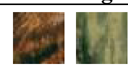


Figure 6. Error distribution with respect to the classes

The figure plots the error rates of annotation based on the Class Reduction and Class Reevaluation strategies using 'complete link' and 'average-link' techniques. Figure 6 demonstrates that Class Reductions strategy paired with 'average-link' clustering technique performs the best for all brushwork classes. The majority of erroneous assignments are in "shading" and "scumbling" classes due to the fact that patterns in these classes may resemble other classes to higher extent. Both Class Reduction and Class Reevaluation strategy produce relatively smaller error for such classes as "divisionism", "mezzapasta" and "pointillism".

Table 3. Examples of misclassifications in "shading" class

Pointillism	Divisionism	Impasto	Scumbling
			

This is due to the fact that patterns of these classes are adequately represented by a number of texture features, resulting in low intra-cluster pair-wise distances in the feature space. Table 3 shows examples of misclassifications for "shading" class patterns into "pointillism", "divisionism", "impasto" and "scumbling" classes.

7. CONCLUSIONS AND FUTURE WORK

In this paper we proposed the semi-supervised multi-expert approach for annotation of brushwork in paintings. Explicit annotation of brushwork is desirable since it helps in the annotation of paintings with higher-level semantic concepts such as artist names, periods of art and paintings styles. To perform annotation, we employed serial combination of multi-experts. This framework benefits from sequential refinement of the assigned labels as well as facilitates dimensionality reduction. To facilitate annotation at the level of individual experts, we employ edsemi-supervised clustering techniques. These techniques model the brushwork classes as tight clusters in the feature space as well as benefit from distribution of unlabelled patterns.

There are several directions for future work. First, we aim to experiment with kernel-based semi-supervised clustering methods at the level of individual experts. These methods employ more sophisticated distance measures, which may lead to improved results. Second, it is preferable to automate feature selection for individual experts in the decision hierarchy and provide the systematic analysis of the relevant features. Lastly, we aim to employ brushwork classes for further annotation of paintings with high-level semantic concepts.

8. REFERENCES

[1] Arnheim. Art and visual perception: A psychology of the creative eye, *University of California Press*, 1954.

[2] Art & Architecture Thesaurus. *Getty Research Institute*,2000.

[3] Canny J. A Computational Approach to Edge Detection, *IEEE PAMI* (8)- 6, 1986.

[4] Chua T.-S., Lim S.-K., Pung H.-K.. "Content-based retrieval of segmented images". *ACM MM*, 211 – 218, 1994.

[5] Feng H., Chua T.-S. A Learning-based Approach for Annotating Large On-Line Image Collection. The International Multi-Media Modelling, 249-256, 2004.

[6] Friedman J. H., An overview of predictive learning and function approximation, From *Statistics to Neural Networks*, Springer Verlag, NATO/ASI, 1-61,1994.

[7] Gluskman H. A. Multicategory classification of patterns represented by high-order vectors of multilevel measurements. *IEEE Tran son Computers* (20), 1593–1598.

[8] Herik, H.J. van den, Postma, E.O. Discovering the Visual Signature of Painters. In *Future Directions for Intelligent Systems and Information Sciences*, 129-147, 2000.

[9] Kaplan L. M. and Kuo C.-C. J., "Texture roughness analysis and synthesis via extended self-similar (ESS) model," *IEEE Trans. Pattern Anal. Machine Intell* (17), 1043–1056, 1995.

[10] Kittler J, Hatef M. Improving recognition rates by classifier combination. *5th Int Workshop on Frontiers of Handwriting Recognition,* 81–102, 1996.

[11] Klein, D., Kamvar, S. D., & Manning, C. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. *Proceedings of ICML,* 307–314, 2002.

[12] Li J., Wang J. Z. Studying Digital Imagery of Ancient Paintings by Mixtures of Stochastic Models, *IEEE Transactions on Image Processing*, vol. 13 (3), 2004.

[13] Mallat S. A theory for multi-resolution signal decomposition: the wavelet representation, IEEE PAMI (11), 674-693, 1989.

[14] Mandelbrot B. B, *The Fractal Geometry of Nature*. San Francisco, CA: Freeman, 1982.

[15] Manjunath B. S., Ma W. Y., "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Machine Intell* (18), 837–842, 1996.

[16] Marchenko Y., Chua T.-S., Aristarkhova I., Jain R. Representation and Retrieval of Paintings based on Art History Concepts. *IEEE Int'l Conf. on Multimedia and Expo (ICME)*, 2004.

[17] Marchenko Y., Chua T.-S., Aristarkhova I., Analysis of paintings using Color Concepts. *IEEE ICME*, 2005.

[18] Melzer, T., Kammerer, P., Zolda E. Stroke detection of Brush Strokes in Portrait Miniatures using Semi-Parametric and a Model-Based Approach. In *Proc. of 14th ICPR,* 1998.

[19] Pudil P, Novovicova J, Blaha S. Multistage pattern recognition with reject option,*1th IAPR ICPR*, 92–95, 1992.

[20] Rahman A. F. R, Fairhurst M. C: Serial Combination of Multiple Experts: A Unified Evaluation. *Pattern Anal. Appl. 2*(4), 292-311, 1999.

[21] Schueermann J, Doster W. A decision theoretic approach to hierarchical classifier design. *Pattern Recognition*; 17(3), 359–369, 1983.

[22] Tung A., Han J., Lakshmanan L., Ng R., Constraint Based Clustering in Large Databases *Proc. Int'l Conf. Database Theory Conf.,* pp. 405-419, 2001.

[23] Teague, M.R. Image Analysis via the General Theory of Moments, *Journal of the Optical Society of America*, 70 (8), 920-930.

[24] Vapnik, V. Estimation of Dependences Based on Empirical Data. Springer Verlag, New York, 1982.

[25] Zhang D. S.and Lu G. Content-Based Shape Retrieval Using Different Shape Descriptors: A Comparative Study, *IEEE ICME, 2001*.