Combining Multiple Kernels for Efficient Image Classification

Behjat Siddiquie
University of Maryland
College Park, MD 20742
behjat@cs.umd.edu

Shiv N. Vitaladevuni
Janelia Farm Research Campus
Howard Hughes Medical Institute
vitaladevunis@janelia.hhmi.org

Larry S. Davis
University of Maryland
College Park, MD 20742
lsd@cs.umd.edu

Abstract

We investigate the problem of combining multiple feature channels for the purpose of efficient image classification. Discriminative kernel based methods, such as SVMs, have been shown to be quite effective for image classification. To use these methods with several feature channels, one needs to combine base kernels computed from them. Multiple kernel learning is an effective method for combining the base kernels. However, the cost of computing the kernel similarities of a test image with each of the support vectors for all feature channels is extremely high. We propose an alternate method, where training data instances are selected, using AdaBoost, for each of the base kernels. A composite decision function, which can be evaluated by computing kernel similarities with respect to only these chosen instances, is learnt. This method significantly reduces the number of kernel computations required during testing. Experimental results on the benchmark UCI datasets, as well as on a challenging painting dataset, are included to demonstrate the effectiveness of our method.

1. Introduction

We address the problem of combining multiple heterogenous features for image classification. Categorizing images based on stylistic variations such as scene content and painting genre requires a rich feature repertoire. Classification is accomplished by comparing distributions of features, e.g., color, texture, gradient histograms [11, 23, 18]. For instance, Grauman and Darrell proposed the Pyramid Match Kernel (PMK) to compute Mercer kernels between feature distributions for Support Vector Machine (SVM) based classification. This has been shown to be effective for object categorization [11] and scene analysis [18]. Approaches such as PMK would compute a kernel matrix for each feature distribution. We explore techniques for combining the kernels from multiple features for efficient and robust recognition.

A number of techniques have been proposed to learn the optimal combination of a set of kernels for SVM-based classification. Lanckriet et al. proposed an approach for Multiple Kernel Learning (MKL) through semi-definite programming [17]. Sonnenburg et al. generalized MKL to regression and one-class SVMs, and enhanced the ability to handle large scale problems. Rakotomamonjy et al. increased the efficiency of MKL and demonstrated its utility on several standard datasets including the UCI repository [22]. They compute multiple kernels by varying the parameters of polynomial and Gaussian kernels, and apply MKL to compute an optimal combination. Bosch et al. learn the optimal mixture between two kernels - shape and appearance - using a validation set [6]. Varma and Ray propose to minimize the number of kernels involved in the final classification by including the L_1 norms of the kernel weights in the SVM optimization function [25]. Bi et al. proposed a boosting-based classifier that combines multiple kernel matrices for regression and classification [5].

The efficiency of MKL-based SVM classifiers during the testing phase depends upon the number of support vectors and the number of features. In general, multi-class problems requiring subtle distinctions entail a large number of support vectors. The computational cost is substantial when the kernels are complex, e.g., matching similarity of feature distributions. Is it possible to reduce the number of complex kernel computations while maintaining performance? We propose an approach for combining multiple kernels through a feature selection process followed by SVM learning. Let $K_m(.,.)$ be the kernel values for the m^{th} feature channel computed using approaches such as the Pyramid Match Kernel. The columns of K_m are considered to be features embedding the images in a high-dimensional space based on similarity to training examples. During the training phase, a subset of the columns are chosen using Gentle Boost [1] based on their discriminative power, and a new kernel K is constructed. This is provided as input to an SVM for final classification. Kernels of test images need to be computed for only the chosen set of columns - much smaller than the full set of kernel values. This results in substantial reductions in computational complexity during the testing phase. The consequent approach is simple and relies on well understood techniques of Boosting and SVMs. Boosting methods have previously been used for feature selection [27], to learn kernels directly from data [8, 14], and for selecting a subset of kernels for concept detection in [15].

We compare our Boosted Kernel SVM (BK-SVM) approach with the Efficient Multiple Kernel Learning (EMKL) approach proposed in [22]. EMKL has been shown to increase the efficiency of kernel learning while enabling the use of a large number of kernels within SVM. It uses all the kernel values for classification - a superset of the features obtained by the greedy Boosting-based selection. BK-SVM and EMKL are tested in two scenarios: standard datasets from the UCI repository [2] and a novel Painting dataset. Results indicate that BK-SVM's classification accuracy is comparable to that of EMKL, with the additional advantage of a much smaller number of complex kernel computations.

Currently, paintings are being extensively digitized in order to preserve them and make them more widely accessible. Digital collections of paintings play an important role in preserving our cultural heritage. Automatic indexing and annotation of such painting collections according to style, artist or period would considerably reduce the manual effort required for such tasks. Supporting query and retrieval on such collections over the internet would make many rare paintings more widely accessible. In this paper, we apply our BK-SVM method to the task of annotation of paintings according to their genre, which could be applied to indexing as well as query and retrieval from painting collections.

The Painting dataset consists of nearly 500 images downloaded from the Internet - the task being to classify images into 6 genres. This provides a good testbed as the classification is subtle, requiring a large variety of features. Recently, there have been studies on the classification of paintings based on their style, artist, period and brushwork [29, 16, 13, 20, 28]. A semi-supervised method employing a variety of feature channels to annotate painting brushwork was presented in [29]. In [16] paintings are classified according to artist. Li et al. [20] have used 2D multi-resolution HMMs with multi-level Debauchies wavelet coefficient features to identify the artists of ancient Chinese paintings. In [19], high level semantic concepts are combined with low level image features to annotate paintings based on period, style and artist. In some of these methods such as [29, 19] a high level of domain knowledge has been used to develop the hierarchy of classes and to select appropriate image features. We use a large repertoire of simple features and rely on machine learning to obtain the combination best suited for the classification. This provides the potential for application in other categorization tasks.

The next section presents details of combining multiple kernels, followed by experiments on the UCI datasets. Section 4 presents the Painting dataset, the features used and the experimental results.

2. Learning a Mixture of Kernels

Content-based image categorization typically represents images with histograms or distributions of features from channels such as texture, color and local gradients [9, 21]. Classification is performed by comparing such distributions. Grauman and Darrell [11] proposed the Pyramid Match Kernel (PMK) for efficiently computing Mercer kernels between feature distributions and apply it to SVM based object categorization [11]. A closely related approach used spatial distributions of features for scene recognition [18]. These techniques use SVM to learn the manifold of image categories and show good generalization. However, classifying images based on subtle style variations, e.g., painting genres, requires a large repertoire of feature channels. Techniques such as PMK would compute a kernel matrix for each feature channel. We are thus faced with the problem of determining the best mixture of the kernels for a given classification task.

A number of Multiple Kernel Learning (MKL) techniques have been proposed to compute linear combinations of kernels for classification by SVM [17, 22, 24]. Let $\{K_1, K_2, \ldots, K_M\}$ be the kernel matrices computed for various feature modalities. MKL computes an optimal classification kernel

$$K(q_i, q_j) = \sum_{m=1}^{M} \beta_m K_m(q_i, q_j) \tag{1}$$

where $\{q_1, q_2, \ldots, q_N\}$ are the training images and β_m is the weight assigned to kernel K_m. Recent MKL techniques have progressively improved training efficiency, e.g., [22, 6]. However, classifying a test image x, using a non-linear SVM, requires computing its kernel value with respect to the selected set of training support vectors S for all feature channels with $\beta_m \neq 0$, i.e. $K_m(q, x)\ \forall q \in S$ and $\forall m$ where $\beta_m \neq 0$. This has $O(c\tilde{N}\tilde{M})$ computational complexity where

- c is the complexity of computing the kernels. This is significant when computing the similarity of distributions.
- $\tilde{N}$ is the number of support vectors, which is less than or equal to the size of the training set, N. Classification problems with difficult decision boundaries require a large set of support vectors. Some approaches propose to reduce this by approximating S with a reduced set of vectors, e.g., [7]. However, they are unsuitable for our case as each kernel is constructed from a different feature modality. Moreover, it is desirable to include as many training images as possible for good generalization (large N).
- $\tilde{M} = |\{m|\beta_m \neq 0\}|$. MKL methods reduce $\tilde{M}$ by imposing sparsity constraints on the weights β_m [24]. However, this may not provide significant benefits when a large variety of features are required for classification.

Is it possible to reduce the number of kernel computations while maintaining performance?

Consider a vector constructed for a test image by concatenating its kernel values with all the training images. For an image x, this would be an NM dimensional vector

$$\mathbf{f}(x) = \langle K_1(q_1, x) \ldots K_1(q_N, x) \ldots K_M(q_1, x) \ldots K_M(q_N, x)\rangle \tag{2}$$

We use Gentle Boost to determine the set P, containing the most discriminative dimensions of $\mathbf{f}(\mathbf{x})$, for the classification problem. The size of P is chosen such that $|P| \ll \tilde{N}\tilde{M}$. This results in a reduced dimensional vector for each image, denoted by $\tilde{\mathbf{f}}(x)$. An SVM is trained to classify images based on the $\tilde{\mathbf{f}}$'s. E.g., for a linear SVM, the kernel between two images x and y would be

$$\Phi(x, y) = \sum_{\langle n,m\rangle \in P} K_m(x, q_n) K_m(q_n, y) \tag{3}$$

For each test image, this requires $O(|P|)$ complex kernel $K_m(.,.)$ computations, and $O(N|P|)$ computations of a simpler kernel such as linear or RBF. This significantly reduces the computational complexity.

To better understand the nature of $\Phi(.,.)$, notice that the Pyramid Match Kernel between two images x and y can be abstracted as a dot-product between two bit-vectors, $\psi_m(x)^T \psi_m(y)$, where m is the feature channel [11]. Therefore, eq.(3) is equivalent to

$$\begin{aligned}
\Phi(x, y) &= \sum_{\langle n,m\rangle \in P} \psi_m(x)^T \psi_m(q_n) \psi_m(q_n)^T \psi_m(y) \qquad (4)\\
&= \sum_m \psi_m(x)^T \left[\sum_{\langle n,m\rangle \in P} \psi_m(q_n) \psi_m(q_n)^T \right] \psi_m(y)
\end{aligned}$$

The inner matrix, $A_m = \sum \psi_m(q_n) \psi_m(q_n)^T$, is a semi-definite matrix. It is easy to show that for a RBF SVM

$$\Phi(x, y) = \exp \frac{1}{\sigma^2} \sum_m \|\psi_m(x) - \psi_m(y))\|^2_{A_m} \tag{5}$$

Intuitively, A's are akin to covariance matrices of the exemplar images in P, the important difference being that $\psi_m(q_n)$ are not zero mean. When P is constructed to maximize discrimination between classes, A defines a discriminative projection.

We note that the approach does not restrict the number of support vectors chosen by the SVM. It only restricts the SVM's kernel to be based on a limited number of base kernel columns.

2.1. Boosting for Feature Selection

Discriminative feature selection is a well studied problem in machine learning, e.g., Xiao et al. propose a variant of boosting called Joint Boost for feature selection [27]. We use Gentle Boost for its simplicity and robustness [1, 10]. Let $\mathbf{f}$'s be d dimensional vectors. d is typically large; in our case $d = NM$. The basic version of Gentle Boost defines a set of weak learners $h(\mathbf{f})$ where each $h(.)$ is a linear classifier along a single dimension. The algorithm iteratively chooses a set of weak learners to maximize classification accuracy. The weak learner chosen at the t^{th} iteration, namely $h_t(.)$, is the one providing maximal increase in classification accuracy with respect to the set of previously chosen classifiers $h_1, \ldots, h_{t-1}$. Thus, the choice of dimensions depends upon the incremental benefit relative to previous choices. Inspite of the greedy nature of the selection process, Boosting has been shown to perform well in many classification tasks [10]. The outline of Gentle Boost is given below:

- Given: $(x_1, y_1), \ldots, (x_n, y_n)$ where $x_i \in X$ and $y_i \in \{-1, 1\}$
- Initialize the weights corresponding to the training samples $W(i) = \frac{1}{n}$
- For $t = 1, \ldots, T$
  - Choose confidence value $\alpha_t \in R$
  - Find the classifier h_t which minimizes the classification error with respect to the distribution W_t
  - Update the weights $W_{t+1}(i) = \frac{W_t(i) e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$ where Z_t is a normalization factor.
- $\{h_t\}$ are the selected features.

3. Experiments with UCI Datasets

The boosting-based feature selection is an efficient but greedy approach. To observe its performance penalties, BK-SVM was applied to four datasets from the UCI repository, specifically the Liver, Ionosphere, Pima and Sonar datasets. The kernels were simple polynomial and Gaussian functions. Here, the motivation was solely to empirically observe the performance on standard datasets. The efficiency gains become evident for more complicated kernel functions used later in the Painting datasets.

The classification results were compared with those of the Efficient Multiple Kernel Learning (EMKL) algorithm described in [22]. For each dataset, a large number of Gaussian and polynomial kernels are computed as described in [22]. The base kernels include Gaussian kernels with 10 different bandwidths σ on all variables and on each single variable, and also polynomial kernels of degree 1 to 3. EMKL and BK-SVM are used to learn a mixture of the kernels appropriate for classification. During the testing phase, the number of kernel computations required in EMKL is a product of the number of kernels selected and the number of support vectors. While in BK-SVM, this depends upon the number of kernel columns chosen by boosting.

The classification results are summarized in Table 1. They indicate that BK-SVM performs close to the baseline EMKL approach even though the number of kernel computations is more than an order of magnitude lower. The loss of performance of approximately 2% may be ascribed to the greedy selection of kernel columns. The results also

Table 1. Experiments on UCI Dataset

Dataset			**BK-SVM**		**EMKL**	
name	size	kernels	accuracy	kernel computations	accuracy	kernel computations
Liver	345	91	66.2 ± 4.7	40	65.0 ± 2.3	1607 ± 324
Ionosphere	351	442	92.1 ± 3.6	40	92.3 ± 1.4	1496 ± 266
Pima	768	117	73.7 ± 6.4	60	75.8 ± 1.6	3123 ± 526
Sonar	208	793	76.3 ± 4.9	20	78.6 ± 4.2	2538 ± 351

demonstrate the scalability of our method, which performs comparably to EMKL even in the case of the Sonar dataset where a large number of kernels(793) are used for learning with only a small number of training examples(104). These trends are reflected in the experiments with the painting dataset - described in the next section. The modest loss in performance is outweighed by the large decrease in computational complexity.

4. Painting Dataset

BK-SVM was applied to painting genre classification. A dataset of 81 Abstract Expressionist, 84 Baroque, 84 Cubist, 82 Graffiti, 89 Impressionist and 78 Renaissance (total of 498) paintings was collected from the Internet. The painting styles along with the painters of each style are listed in Table 2. Some of the public domain images are shown in Figure 1. The distinguishing features for painting styles are not clearly defined due to its abstract nature. There is high intra-class variation due to differences between the painters of a particular style and also between the different paintings of individual painters [3]. The content in the paintings varies significantly and occasionally paintings of different styles depict the same scene, further complicating the problem. Having been compiled from a variety of sources, the images have variations in scale and illumination as well. The classification task is complex, requiring a rich set of features, making this a good testbed for BK-SVM.

4.1. Features

Inspired by previous studies on painting classification, a large variety of features are computed. Each feature channel produces a distribution of filter responses for a given image. The similarity of images is defined as the match between the distributions.

4.1.1 Texture

Texture features capture brushwork and characteristics of the depicted scene. They have been shown to be effective in classification of paintings [20, 29, 13]. We employ the MR8 filter bank [26] as it responds to both isotropic and anisotropic textures and was observed to perform better than Gabor filter banks. The MR8 filter bank consists of a Gaussian and a Laplacian of Gaussian with $\sigma = 10$ and oriented edge and bar filters at 3 scales $(\sigma_x, \sigma_y) = \{(1,3), (2,6), (4,12)\}$ and 6 orientations. Only the maximum response is recorded at each scale for each of the edge and bar filters across all orientations. The responses at all the pixels are combined to form a set of vectors, denoted by F_{texture}.

4.1.2 Histograms of Oriented Gradients (HOG)

HOG based descriptors have been extensively used for representing local shape [9, 4, 21]. They have some degree of invariance to illumination and geometric transformations. We compute two types of features using HOG:

1. F_{HOGdense}: set of HOG features on overlapping 8×8 sized patches placed on a dense regular grid with a spacing of 4 pixels - similar to [9].

2. $F_{\text{HOGsparse}}$: sparse set of HOG features computed on 8×8 patches centered on all edge points. This was inspired by [4].

4.1.3 Color

Color features have been previously employed for classifying paintings [13, 28]. We use local histograms to represent color features consisting of 10 bins of the pixel intensities of each color channel. The histograms are computed in 8×8 sized patches centered on a dense grid over the image. This generates a set of vectors denoted by F_{color}. The histograms of different color channels were concatenated because the joint histograms were quite sparse. Experiments indicated that RGB, HSV and LUV had similar performance. Only results for RGB color-space are presented here.

4.1.4 Saliency

Edge Continuity is used to enhance the saliency of long continuous curves relative to scattered and cluttered edges. We use the technique described in [12] for computing the saliency maps of the images. HOG features are extracted from these saliency maps from patches centered on edges having high saliency. The obtained set of HOG vectors is denoted by F_{HOGsal}.

4.2. Pyramid Match Kernel

Each of the features produces a set of vectors for a given image. For each feature channel, similarity between images is computed based on the similarity between the two sets of vectors, computed using Pyramid Match Kernel (PMK) [11]. The sets can have different cardinalities. The approach has been shown to be efficient and effective

Table 2. Painting Classes

Painting Style	Artist
Abstract Expressionist	Arshile Gorky, Helen Frankenthaler, James Brooks, Jane Frank, Jean Paul Riopelle, Kenzo Okada, Paul Jenkins
Baroque	Anthony Van Dyck, Artemisia Gentileschi, Carravagio, Diego Velazquez, Jan Vermeer, Nicholas Poussin, Peter Paul Reubens, Rembrandt
Cubist	Fernand Leger, Georges Barque, Gino Severini, Jacques Villon, Juan Gris, Lyonel Feininger, Pablo Picasso
Graffiti	-
Impressionist	Alfred Sisley, Camille Pissarro, Claude Monet, Frederic Bazille, Mary Cassatt, Pierre Auguste Renoir, Edouard Manet
Renaissance	Correggio, Raphael, Leonardo Da Vinci, Sandro Botticelli, Titan, Giorgione, Pieter Brueghel, Michelangelo

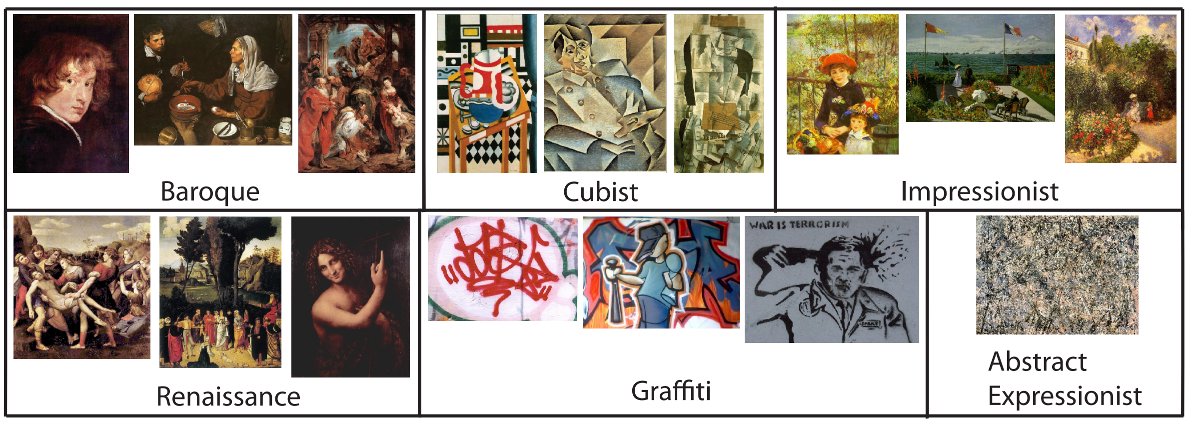

Figure 1. Example images from the Painting database

for image classification. In this section we briefly describe the kernel. Let X and Y be two sets of feature vectors in a d-dimensional feature space. Now consider $L+1$ levels of histograms $H^0, H^1, \ldots, H^L$. Level 0 of the histogram consists of just 1 bin which is the entire space, level 1 of the histogram consists of 2^d bins equally dividing the feature space into two parts along all dimensions. Similarly level l of the histogram consists of $D = 2^{dl}$ bins. Let H^l_X and H^l_Y denote the histograms of X and Y at level l with $H^l_X(i)$ and $H^l_Y(i)$ being the number of feature vectors of X and Y respectively falling into the ith bin at level l. A histogram intersection gives the number of matches at this level.

$$I(H^l_X, H^l_Y) = \sum_{i=1}^{D} \min(H^l_X(i), H^l_Y(i))$$

But note that all the matches at level $l+1$ are also matches at this level and hence the number of new matches at level l is $I(H^l_X, H^l_Y) - I(H^{l+1}_X, H^{l+1}_Y)$. The matches at level l are weighted by $\frac{1}{2^{L-l}}$ in order to give higher weights to matches which happen at smaller bin sizes and hence have a higher similarity. The total match between X and Y at all levels is defined as the similarity between X and Y

$$K(X,Y) = I(H^L_X, H^L_Y) + \sum_{l=0}^{L-1} \frac{1}{2^{L-l}} I(H^l_X, H^l_Y) - I(H^{l+1}_X, H^{l+1}_Y)$$

To avoid biasing the kernel toward larger input sets it is normalized

$$K(X,Y) = \frac{K(X,Y)}{\sqrt{K(X,X)K(Y,Y)}}$$

This normalization also ensures that $\forall X,Y\ K(X,Y) \in [0,1]$. It has been shown that Pyramid Match produces a Mercer kernel and can be directly used in an SVM.

4.3. Classification Results

Training the BK-SVM consists of the following steps:

- Each of the M described features is extracted for all training images q_i.
- PMK was used to compute kernel values $K_m(q_i, q_j)$, $\forall q_i, q_j, m$, producing M kernel matrices, $K_1, \ldots, K_M$.
- A vector $\mathbf{f}_i$ is constructed for each q_i by concatenating the kernel values as defined in eq.(2).

(a) Original Image (a) Gradient Magnitude Map (a) Saliency Map

Figure 2. Salient Edges

Table 3. Painting Classification Results

Feature	Accuracy
texture	73.5 ± 1.1
color	70.6 ± 1.1
dense HOG	69.3 ± 1.2
sparse HOG	69.0 ± 1.0
Saliency	62.2 ± 0.7
Combined EMKL	82.4 ± 0.9
Combined Our Method	81.3 ± 0.6

- Boosting is used to select a set of L dimensions that best classify $\mathbf{f}_i$'s into the painting genres. The number of exemplar images selected is equal to the number of iterations of boosting and thus can be easily controlled.
- A new RBF kernel matrix Φ is constructed from the selected dimensions (i.e. columns of K_m's) through the relation in eq.(5). A one-vs-all multi-class SVM is trained on Φ.

During the testing phase, PMK is computed between a given test image and the L selected training images. Classification is performed through the trained RBF SVM.

For comparison, EMKL was employed for the same classification task. For EMKL, we learn separate kernels for each individual classifier, using the same parameters that were used for the UCI datasets($C = 100$,maximal number of iterations=500,duality gap=0.01). All the experiments were repeated 10 times with a 5-fold cross-validation.

4.3.1 Individual Features

Table 3 shows the performance of the individual classifiers, only the net results are shown due to space constraints. We now discuss the performance of individual features.
Color: In Baroque and Renaissance paintings, darker colors are used and this makes color histograms particularly useful for discriminating them from the other classes. With color features alone, Baroque paintings had a recognition rate of 91%. Color features are also useful for identifying Impressionist paintings as they tend to depict outdoor scenes with sunlight, landscapes and greenery.
Texture: The texture feature proved useful for distinguishing Impressionist images as they have distinctive brush strokes. Baroque paintings being darker, generate low responses with the filter banks and are also easily identified.
HOG: The cubist paintings are composed of dense geometrical structures such as straight lines, cubes and cylinders. Consequently, local shape features such as the dense HOG are useful in distinguishing them. The sparse HOG features encode the local shape around the edge points and prove useful for identifying Impressionist paintings.
Saliency: Graffiti paintings tend to have smooth continuous contours, which get enhanced in the saliency maps(Fig. 2), computed using edge continuity techniques[12]. The local shape features around these salient contours help discriminate them. These saliency based features help recognize Graffiti paintings with an accuracy of 82%.

4.3.2 Combination of Features

The features, in general, perform quite well individually and also complement each other resulting in a significant improvement in performance when combined. For instance, on the sole basis of color, a dark colored graffiti painting may be confused as a baroque painting. However, local shape information provided by saliency maps helps reduce this confusion. The results are listed in Table 3. They indicate that both the EMKL and our method perform much better than each of the individual feature channels. The confusion matrix obtained after combining features using BK-SVM is shown in Fig. 3. There is some degree of confusion between abstract expressionist and cubist paintings and most of the misclassifications happened to be abstract expressionist paintings containing geometrical structures characteristic to cubist paintings or cubist paintings lacking these geometrical shapes. There are also some errors between impressionist and renaissance paintings.

4.3.3 Feature Selection

To gain further insight into the construction of the individual one-vs-all classifiers, we looked at the average weights allocated by EMKL to the kernels for each individual classifier(Fig. 4). Color being an important feature was assigned a high weight in each of the individual classifiers and as expected, it turned out to be the most dominant feature for distinguishing Baroque paintings. Similarly the saliency kernel is weighted relatively high in the Cubist and Graffiti classifiers. Texture is also important in case of the Baroque, Impressionist and Renaissance classes. Sparse HOG features are assigned high weights in all the classifiers, indicating the significance of local shape information. Dense

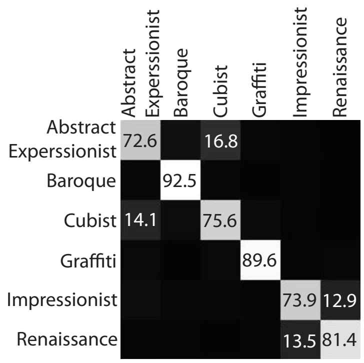

Figure 3. Confusion Matrix for the painting dataset

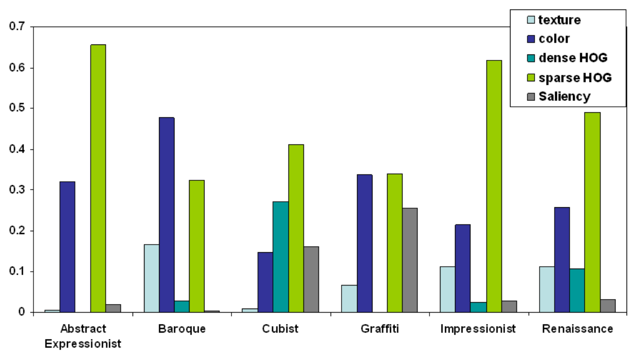

Figure 4. Avg. kernel weights learnt by EMKL for each classifier

HOG features are allocated high weights in the Cubist classifier as expected. On the whole, the weights seemed quite intuitive with features that distinguished a particular class well, being assigned a higher weight in the respective classifier. However, texture was weighted relatively low which is surprising, given the fact that it performs quite well individually. We conjecture that since both texture and HOG are based on local edges, they contain redundant information resulting in texture being ignored.

We did a similar study for BK-SVM, where we examined the proportion of exemplar images selected from each kernel for the individual classifiers(Fig. 5). Though some of the above mentioned trends were observed, like color and saliency being important for the baroque and graffiti paintings respectively, no single feature dominated the individual classifiers. We hypothesize that this is a result of the lack of any external constraints imposed by our method unlike the sparsity constraint imposed by EMKL.

4.3.4 Efficiency

Figure 6 plots the performance of our method as a function of the number of kernel computations required. It can be seen that BK-SVM reduces the number of kernel computations required by a factor of 10, while suffering only a minor (1-2%) reduction in accuracy. It can also be observed that the performance decreases gradually as the number of features selected is decreased. At a relative speedup of 100 with respect to MKL, BK-SVM is still better than each of the individual feature channels and only 7% less accurate than MKL.

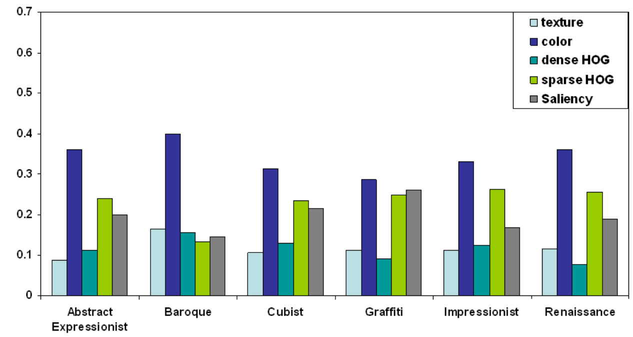

Figure 5. Avg. proportion of exemplar images selected from the feature channels for each classifier

We also apply our method on the individual kernels and compare the performance of BK-SVM with that of a SVM using a single kernel. Here BK-SVM is used to learn a kernel from a subset of the training images, while SVM uses the kernel computed from the entire training set. As expected, the performance increases with the increasing number of features selected and approaches that of a SVM while being more efficient. Figure 6 once again underscores the importance of combing multiple features for improving accuracy both for EMKL as well as BK-SVM.

In the Painting dataset, BK-SVM requires nearly 10 times fewer kernel computations than EMKL for achieving comparable accuracy. This speedup, though substantial, is less compared to the 40-120 time reduction achieved on the UCI datasets. There are two plausible explanations. Firstly, the painting dataset has multiple classes, which makes the decision boundaries more complex than in case of the UCI datasets, which have only two classes. Secondly, the UCI dataset experiments use base kernels produced by varying the parameters of Gaussian and polynomial kernels, many of which are likely to be redundant. Hence, a sparse set of features selected by Boosting is sufficient to accurately approximate the optimal kernel. In case of the painting dataset, each of the base kernels are computed from different feature channels containing complementary information. Consequently, a number of exemplar instances are selected from each base kernel.

5. Summary

This paper has presented a simple and efficient approach for learning a mixture of kernels. Our method, which learns a mixture of kernels by greedily selecting exemplar data instances corresponding to each kernel using AdaBoost, has been shown to compare well to multiple kernel learning methods, while simultaneously reducing the number of kernel similarity computations required. The effectiveness of our method with respect to MKL has been demonstrated on

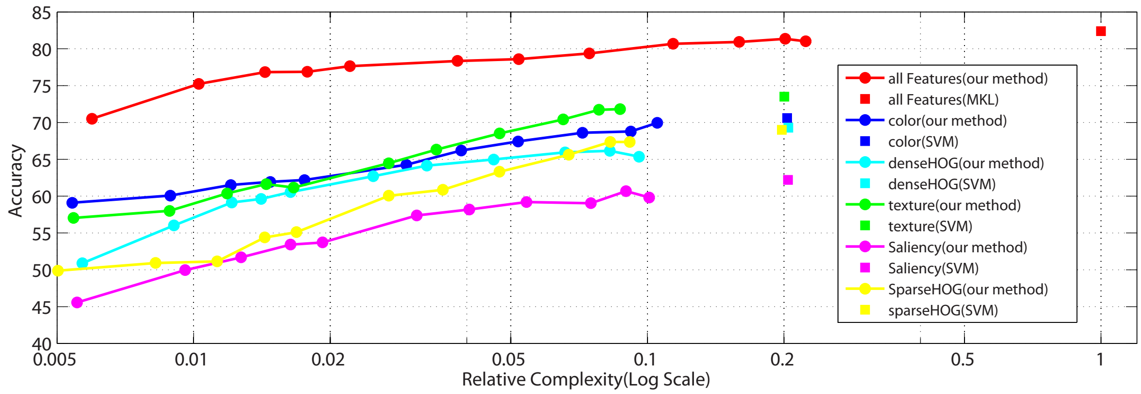

Figure 6. Variation in performance as a function of the number of features selected for the painting dataset

some of the benchmark UCI datasets. We have also tested our method on an extremely diverse and challenging painting dataset, where a single feature channel is inadequate for classification. We combine multiple kernels computed from different feature channels, obtaining results comparable to the MKL method. The results provide evidence that our method is almost as accurate as the multiple kernel learning method, while being computationally much more efficient.

Acknowledgement: This research was partially supported by the ONR grant N000140910044.

References

[1] http://research.graphicon.ru/machine-learning/gml-adaboost-matlab-toolbox.html. 1, 3

[2] http://archive.ics.uci.edu/ml/. 2

[3] R. Arnheim. Art and visual perception. a psychology of the creative eye. 1955. 4

[4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on PAMI*, 2002. 4

[5] J. Bi, T. Zhang, and K. P. Bennett. Column-generation boosting methods for mixture of kernels. *ACM-SIGKDD*, 2004. 1

[6] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. *ACM-CIVR*, 2007. 1, 2

[7] C. J. C. Burges. Simplified support vector decision rules. *ICML*, 1996. 2

[8] K. Crammer, J. Keshet, and Y. Singer. Kernel design using boosting. *NIPS*, 2002. 1

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *CVPR*, 2005. 2, 4

[10] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 2000. 3

[11] K. Grauman and T. Darrell. The pyramid match kernel: discriminative classification with sets of image features. *ICCV*, 2005. 1, 2, 3, 4

[12] G. Guy and G. Medioni. Inferring global perceptual contours from local features. *IJCV*, 1996. 4, 6

[13] H. v. d. Herik and E. O. Postma. Discovering the visual signature of painters. 2, 4

[14] T. Hertz, A. B. Hillel, and D. Weinshall. Learning a kernel function for classification with small training samples. *ICML*, 2006. 1

[15] W. Jiang, S.-F. Chang, and A. Loui. Kernel sharing with joint boosting for multi-class concept detection. *CVPR*, 2007. 1

[16] D. Keren. Recognizing image "style" and activities in video using local features and naive bayes. *PRL*, 2003. 2

[17] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *JMLR*, 2004. 1, 2

[18] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *CVPR*, 2006. 1, 2

[19] L. Leslie, T.-S. Chua, and R. Jain. Annotation of paintings with high-level semantic concepts using transductive inference and ontology-based concept disambiguation. *ACM-MM*, 2007. 2

[20] J. Li and J. Z. Wang. Studying digital imagery of ancient paintings by mixtures of stochastic models. *IEEE Transactions on Image Processing*, 2004. 2, 4

[21] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2003. 2, 4

[22] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. More efficiency in multiple kernel learning. *ICML*, 2007. 1, 2, 3

[23] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *IJCV*, 2000. 1

[24] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *JMLR*, 2006. 2

[25] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. *ICCV*, 2007. 1

[26] M. Varma and A. Zisserman. Classifying images of materials: Achieving viewpoint and illumination independence. *ECCV*, 2002. 4

[27] R. Xiao, W. Li, Y. Tian, and X. Tang. Joint boosting feature selection for robust face recognition. *CVPR*, 2006. 1, 3

[28] M. Yelizaveta, C. Tat-Seng, and A. Irina. Analysis and retrieval of paintings using artistic color concepts. *ICME*, 2005. 2, 4

[29] M. Yelizaveta, C. Tat-Seng, and R. Jain. Semi-supervised annotation of brushwork in paintings domain using serial combinations of multiple experts. *ACM-MM*, 2006. 2, 4