A Comparison of ℓ_1 Norm and ℓ_2 Norm Multiple Kernel SVMs in Image and Video Classification

Fei Yan Krystian Mikolajczyk Josef Kittler Muhammad Tahir
Centre for Vision, Speech and Signal Processing
University of Surrey
Guildford, GU2 7XH, UK
{f.yan,k.milolajczyk,j.kittler,m.tahir}@surrey.ac.uk

Abstract

SVM is one of the state-of-the-art techniques for image and video classification. When multiple kernels are available, the recently introduced multiple kernel SVM (MK-SVM) learns an optimal linear combination of the kernels, providing a new method for information fusion. In this paper we study how the behaviour of MK-SVM is affected by the norm used to regularise the kernel weights to be learnt. Through experiments on three image/video classification datasets as well as on synthesised data, new insights are gained as to how the choice of regularisation norm should be made, especially when MK-SVM is applied to image/video classification problems.

1 Introduction

Owing to advances in both computer hardware and computer algorithms, the field of multimedia indexing and retrieval has witnessed rapid growth in recent years. Multimedia retrieval can be naturally formulated as a classification problem with probabilistic output. Support vector machine (SVM) is one of the most successful techniques for such a classification problem. Recently, a multiple kernel variant of SVM (MK-SVM) has been proposed in the machine learning community [8, 17]. When multiple kernels encoding complementary characterisations of a problem are available, MK-SVM automatically learns the "optimal" weights of kernels from the training data, thus offering improved classification performance.

In the MK-SVM framework, regularising the kernel weights with ℓ_1 norm and ℓ_2 norm leads to ℓ_1 norm and ℓ_2 norm MK-SVMs, respectively. In [7], experiments on synthesised data show that ℓ_1 norm and ℓ_2 norm MK-SVMs both can be advantageous, depending on the property of the kernels in terms of "information redundancy". In this paper, we extend the study in [7] by comparing the behaviour of ℓ_1 norm and ℓ_2 norm MK-SVMs on image and video classification problems. We also provide more insights as to how one should choose between ℓ_1 norm and ℓ_2 norm MK-SVMs.

The rest of this paper is organised as follows. In Section 2, we introduce SVM, MK-SVM and the differences between ℓ_1 norm and ℓ_2 norm MK-SVMs. Local feature based kernels for image classification are briefly described in Section 3. In Section 4, a comparison of ℓ_1 norm and ℓ_2 norm MK-SVMs is provided, based on experiments on three benchmark image/video classification datasets as well as synthesised data. Finally conclusions are given in Section 5.

2 Multiple Kernel Support Vector Machine

In this section SVM with single as well as multiple kernels are presented. We then discuss two norms that can be used in MK-SVM.

2.1 Support Vector Machine

Support Vector Machine (SVM) [18, 3] has become the state-of-the-art method for many classification problems since its introduction. In an SVM, a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ is a function defined for a pair of examples in an original input space, which captures the similarity between them. Effectively, a kernel function maps the training examples in the input space into a high dimensional feature space. A "maximal separating hyperplane" in the feature space is then found by solving an optimisation problem. Such a separating hyperplane provides a good trade-off between learning ability and model complexity, and hence a good generalisation performance.

More specifically, given training vectors $\mathbf{x}_i \in \mathcal{R}^d, i = 1, \cdots, m$ with class labels $y_i \in \{1, -1\}$, an SVM classifies a new vector $\mathbf{x}$ according to the following linear decision function:

$$y = \text{sgn}\{\sum_{i=1}^{m} y_i \alpha_i^* K(\mathbf{x}, \mathbf{x_i}) + b^*\} \tag{1}$$

where α_i^* and b^* define the maximal separating plane, and it turns out finding this plane is equivalent to solving the following quadratic programming (QP) problem [18, 3]:

$$\begin{array}{ll} \max\limits_{\boldsymbol{\alpha}} & S(\boldsymbol{\alpha}) \\ \text{subject to} & \sum_{i=1}^{m} y_i \alpha_i = 0 \\ & \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1} \end{array} \tag{2}$$

where $\boldsymbol{\alpha} \in \mathcal{R}^m$, and

$$S(\boldsymbol{\alpha}) \triangleq \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{3}$$

2.2 SVM with Multiple Kernels

Applying a kernel function on each pair of the m training examples results in an $m \times m$ symmetric matrix known as a kernel matrix. In many classification problems, multiple kernel matrices can be constructed. For example, for a given set of features in the input space, different distance metrics can be used as kernel functions to capture different "views" of the similarity. Moreover, in some cases, several information modalities are available. For example, in video classification, visual and audio information both can be used. Even when considering only visual information, various types of features can be extracted, such as texture, colour, etc. Information from each of these "channels" can be used to construct a kernel matrix, again resulting in multiple kernels.

When multiple kernels are available, the following question arises naturally: how can we combine the kernels to improve the performance of a kernel based learning algorithm, such as an SVM? Mathematically, let K_k be the k^{th} of the n available kernels, we would like to find a set of linear mixture coefficients, $\boldsymbol{\beta} \in \mathcal{R}^n$:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{n} \beta_k K_k(\mathbf{x}_i, \mathbf{x}_j) \tag{4}$$

such that the resulting kernel K gives good performance on test data.

One straightforward way of finding a good set of mixture coefficients is to use cross validation, which is also a universally applicable method for model selection. In fact, this idea has been exploited in object classification. In [2], two kernels, one capturing shape similarity of objects, the other capturing appearance similarity, are constructed. A weighted combination of the two kernels is used in a conventional single kernel SVM (SK-SVM). The weights of the kernels are learnt in a brute force search over a validation set. This approach, although demonstrated effective in the paper, quickly becomes impractical as the number of kernels grows. Another way of combining kernels is to weight them uniformly, i.e., to set $\beta_k = \frac{1}{n}$ for all kernels. In this approach the kernel weights are set rather arbitrarily, without any knowledge of the training data, and thus may not be optimal.

The idea of learning optimal kernel weights for SVM from training data was first introduced in [8], where the margin of an SVM is maximised with respect both to $\boldsymbol{\alpha}$ and to kernel weights $\boldsymbol{\beta}$. This leads to a min-max problem:

$$\begin{array}{ll} \min\limits_{\boldsymbol{\beta}} \max\limits_{\boldsymbol{\alpha}} & S(\boldsymbol{\alpha}, \boldsymbol{\beta}) \\ \text{subject to} & \sum_{i=1}^{m} y_i \alpha_i = 0 \\ & \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1} \\ & \boldsymbol{\beta} \geq \mathbf{0} \\ & ||\boldsymbol{\beta}||_1 = 1 \end{array} \tag{5}$$

where $\boldsymbol{\alpha} \in \mathcal{R}^m$, $\boldsymbol{\beta} \in \mathcal{R}^n$, and

$$S(\boldsymbol{\alpha}, \boldsymbol{\beta}) \triangleq \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j \sum_{k=1}^{n} \beta_k K_k(\mathbf{x}_i, \mathbf{x}_j) \tag{6}$$

In [8], (5) is formulated as a semi-definite program (SDP). Following [8], several other formulations have been proposed in the literature [1, 12, 17]. These formulations essentially solve the same problem, and differ only in the optimisation techniques employed.

2.3 ℓ_1 Norm and ℓ_2 Norm MK-SVMs

The multiple kernel learning framework discussed above imposes an ℓ_1 norm regularisation on the kernel weights, i.e., $||\boldsymbol{\beta}||_1 = 1, \boldsymbol{\beta} \geq \mathbf{0}$. This convex constraint makes the associated optimisation problem easier to solve. However, it has been known that ℓ_1 norm regularisation tends to produce sparse solutions (e.g. [13]), which means during the learning most kernels are assigned virtually zero weights. This behaviour may not always be desirable, since the information carried in the kernels that get zero weights is completely discarded.

A non-sparse version of MK-SVM is proposed by Kloft et al. in [7], where an ℓ_2 norm regularisation is imposed instead of ℓ_1 norm. Comparing to (5), the only difference in their formulation is that the $||\boldsymbol{\beta}||_1 = 1$ constraint is replaced by $||\boldsymbol{\beta}||_2 = 1$. The associated optimisation problem is complicated by this modification, since the set given by $\{\boldsymbol{\beta} : ||\boldsymbol{\beta}||_2 = 1, \boldsymbol{\beta} \geq \mathbf{0}\}$ is not convex. This is remedied by seeking a tight approximation rather than the exact solution of the problem.

In [7], experiments on synthesised data show that ℓ_1 norm and ℓ_2 norm MK-SVMs both can be advantageous, depending on the property of the kernels in terms of "information redundancy": ℓ_1 norm regularisation is better when the kernels contain a large amount of redundant information among them, otherwise ℓ_2 norm version is preferable. In this paper, we extend the study in [7] by comparing the behaviour of ℓ_1 norm and ℓ_2 norm MK-SVMs on image and video classification problems. We also provide more insights as to how one should choose between ℓ_1 norm and ℓ_2 norm MK-SVMs.

3 Kernels

Kernel construction involves feature extraction, example representation and similarity measure. Recently, local features have become popular in image classification and object recognition [6, 9, 14]. To extract local features from an image, first small patches in the image are chosen, either through interest point detection, e.g., with a Harris-Laplace detector [11], or by densely sampling the image at multiple scales. A local feature descriptor, e.g., a SIFT [10], is then used to characterise each patch. This results in a set of local features, which is also referred to as a bag of words, in analogy with the features used in text classification.

The sets of local features extracted from all the training images are clustered to form a codebook [5], where each cluster can be thought of as a "code". For an given image, the extracted local features are mapped onto this codebook according to which cluster each of the features belongs to. After this process, a histogram is obtained for each image whose size is equal to the size of the codebook. A function that measures the similarity between two such histograms can be used as a kernel function in kernel-based learning algorithms, such as an SVM, providing that Mercer's condition [3] is satisfied. Such functions include histogram intersection, χ^2 distance function, etc.

Several extensions to this codebook based approach have been proposed. In pyramid match kernel (PMK) [6], a vocabulary "tree" is constructed instead of a "flat" codebook, by recursively applying clustering on the training features. By mapping features onto this tree, multi-resolution histograms are generated. This allows for weighting the intersection of two such multi-resolution histograms differently according to which level of the histogram is being considered, thus offering a more accurate measure of the similarity of two feature sets.

In [9], a variant of PMK which encodes spatial information, spatial PMK (SPMK), is proposed. The basic idea of SPMK is to take into account the spatial distribution of the features when computing the similarity of two histograms. Images are divided into spatial location grids. If two features from two images in the same cluster fall into the same grid, they contribute more to the similarity function than otherwise. The improvement of SPMK over PMK is significant, especially when the objects are well aligned [9].

In our experiments, various combinations of sampling techniques, descriptors, and kernel functions, are used to generate kernels. We will discuss this in more details in next section.

4 Experiments

In this section we perform a number of experiments to demonstrate advantages and disadvantages of the two norms. We first discuss the standard datasets and evaluation criteria and then present the results.

4.1 Datasets

Three datasets for image/video classification and retrieval are used in the experiments: PASCAL visual object classes challenge 2008 development set (VOC08) [4], Mediamill video retrieval challenge set [16], and TRECVid video retrieval evaluation 2007 development set [15]. Some statistics of the three datasets are shown in Table 1. Note that although Mediamill and TRECVid07 are essentially video classification problems, we use only one keyframe from each video shot to classify the shot. Other information modalities, e.g., audio, text, are not used.

Table 1. Some statistics of the datasets

	VOC08	Mediamill	TRECVid07
no. of classes	20	101	36
size of train set	2111	30993	9060
size of test set	2221	12914	9060

4.2 Performance Criterion

The classification of different classes in a dataset are treated as independent binary problems. Take Mediamill dataset for example. There are 101 semantic concept classes, such as explosion, indoor, military, etc. The objective is to make 101 binary decisions for each given test image as to whether it contains each of the 101 concepts. In our experiments, average precision is used to measure the performance of each binary classifier. To calculate average precision, all test examples are ordered (from high to low) according to the probabilities that they belong to the class under considering, where the probabilities are give by the binary classifier trained for this class. Let $\mathcal{E}^i = \{e_1, e_2, \cdots, e_i\}$ be the subset of the ranked examples which contains the top i examples, and $\mathcal{X}$ be the set of

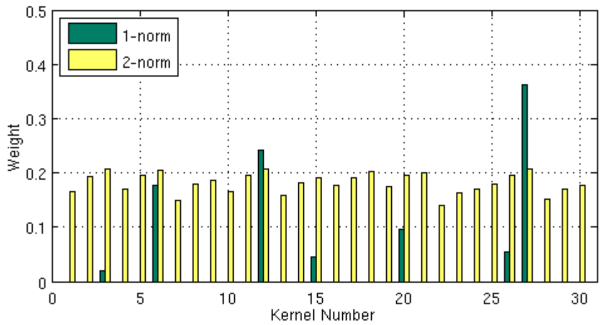

Figure 1. Kernel weights when only informative kernels are used. "motorbike" class.

examples that belong to the class. Average precision, AP, is defined as [16]:

$$\text{AP} = \frac{1}{|\mathcal{X}|}\sum_{i=1}^{l}\frac{|\mathcal{X}\cap\mathcal{E}^i|}{i}I(e_i) \tag{7}$$

where l is the number of test examples, and $I(\cdot)$ is an indicator function: $I(e_i) = 1$ if $e_i \in \mathcal{X}$ and $I(e_i) = 0$ otherwise. Once AP is defined, the mean of the APs for all classes in a dataset, MAP, serves as an indicator of the overall performance.

4.3 VOC08 Results

Using only informative kernels To generate kernels for VOC08 dataset, 2 sampling techniques: dense sampling and Harris-Laplace interest point sampling; 5 colour variants of SIFT descriptors; and 3 ways of dividing an image into spatial location grids for SPMK, are used (see [14] for details). The combination of them results in $2 \times 5 \times 3 = 30$ kernels, each of which is generated using a generalised RBF kernel with χ^2 distance.

Fig. 1 plots the learnt kernel weights for the "motorbike" class in VOC08 as an example. It is evident that ℓ_1 norm MK-SVM produces sparse kernel selection results. Since the kernels computed in our experiments carry complementary information about the images, setting the weights of some kernels to zeros means useful information carried in those kernels is completely discarded.

In Table 2, the first column shows for each class, the best performance of the 30 kernels with SK-SVMs in terms of average precision. Note that the best performance for different classes may be achieved with different kernels, so the MAP in this column is not "realistic". The next three columns of the table show the performance of three kernel level fusion schemes. The first scheme uses an SK-SVM with a kernel obtained by weighting the 30 kernels uniformly. The last two columns are the APs obtained with

Table 2. VOC08 average precisions

	sk-svm max. of 30	sk-svm uniform	mk-svm ℓ_1 norm	mk-svm ℓ_2 norm
aeroplane	0.725	0.746	0.744	**0.795**
bicycle	0.341	**0.381**	0.375	**0.381**
bird	0.424	0.489	0.477	**0.515**
boat	0.575	0.590	0.598	**0.632**
bottle	0.202	0.174	0.158	**0.176**
bus	0.445	**0.534**	0.500	0.530
car	0.515	0.538	0.517	**0.539**
cat	0.493	0.538	0.517	**0.549**
chair	0.424	0.414	0.400	**0.419**
cow	0.188	0.155	0.130	**0.171**
diningtable	0.270	0.243	0.255	**0.258**
dog	0.335	**0.340**	0.320	0.326
horse	0.411	0.451	0.442	**0.471**
motorbike	0.346	0.391	0.348	**0.401**
person	0.845	0.863	0.866	**0.885**
potted plant	0.278	**0.258**	0.212	0.257
sheep	0.296	0.298	0.247	**0.299**
sofa	0.407	0.345	0.367	**0.369**
train	0.557	0.641	0.629	**0.654**
tv monitor	0.492	**0.537**	0.530	0.530
winner of	-	5	0	16
MAP	0.428	0.446	0.432	**0.458**

ℓ_1 norm MK-SVM and ℓ_2 norm MK-SVM, respectively. It is clear that ℓ_2 norm MK-SVM outperforms its ℓ_1 norm counterpart in all 20 classes. When comparing all three kernel fusion schemes, the naive uniform approach wins in 5 classes out of 20, and ℓ_2 norm MK-SVM wins in 16.

Mixing informative and random kernels To further investigate how the properties of kernels affect the performance of ℓ_1 norm and ℓ_2 norm MK-SVMs, we introduce random kernels in our experiments. We call the 30 kernels used previously the 30 informative kernels. We then generate 30 random kernels (Gram matrices of 10 dimensional random vectors) and mix them with the informative ones. In the first run, only the 30 random kernels are used. In the following runs the number of informative kernels is increased and that of random kernels decreased, until in the 31st run, where all 30 kernels are informative.

Fig. 2 plots the MAP of the three kernel fusion schemes as the composition of the kernels changes. First of all, as the number of informative kernels increases, the performance of all three approaches also increases. Secondly, it is clear that ℓ_1 norm MK-SVM outperforms the ℓ_2 norm version when the number of random kernels is high. As the number of informative kernels increases, the MAP of ℓ_2 norm MK-SVM increases faster than ℓ_1 norm version. It surpasses that of ℓ_1 norm when there are 19 informative kernels, and widens its advantage in the rest of the runs.

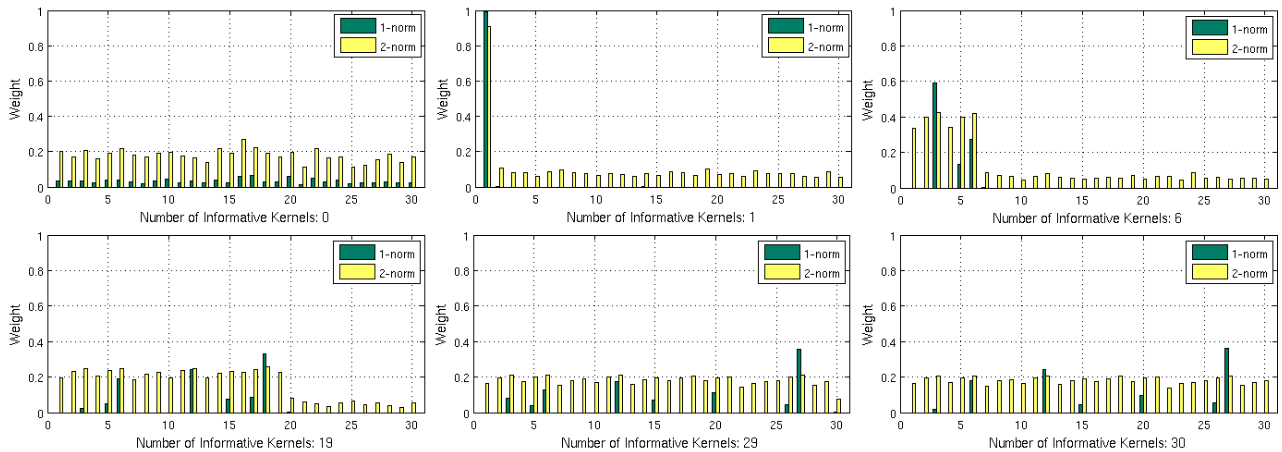

Figure 3. Kernel weights when both informative and random kernels are used. "motorbike" class.

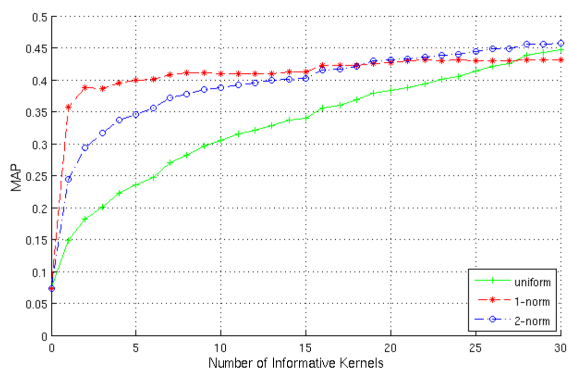

Figure 2. MAP with various informative / random kernel mixture.

The sub-figures in Fig. 3 plot the learnt kernel weights for the "motorbike" class when there are 0, 1, 6, 19, 29, 30 informative kernels, respectively. When all kernels are random (top-left), that is equally uninformative, the distributions of weights allocated by both methods are close to uniform. As soon as one informative kernel is introduced (top-middle), ℓ_1 norm MK-SVM detects it, and assigns almost all importance to it. ℓ_2 norm MK-SVM however, although also assigns a large weight to this kernel, gives a relatively significant amount of weight to the random kernels. This behaviour is responsible for its poor MAP performance in the left part of Fig. 2. Note that in each sub-figure of Fig. 3 the weights of the informative kernels are plotted towards the left end and those of random ones towards the right.

As the number of informative kernels increases, the useful information discarded by ℓ_1 norm MK-SVM due to its "over-regularisation" increases; while the noise included by ℓ_2 norm MK-SVM due to its "under-regularisation" decreases. The point where the two methods have comparable MAP in Fig. 2 can be thought of as the point at which ℓ_1 norm MK-SVM's tendency to discard useful information and ℓ_2 norm MK-SVM's tendency to include noise reach a balance.

To conclude, in addition to the information redundancy criterion discussed in [7], noise level in the kernels can also be used to help decide which version of MK-SVM to choose. Although we demonstrate this on semi-synthesised data, one can imagine practical situations where certain kernels are informative for some classes in a dataset, but are not for other. For example, kernels built from colour histograms help in concept classes such as "snow", "desert", but contribute little to classes without much colour characteristics. Another example would be that kernels specifically designed for certain concepts, e.g., a kernel based on a face detector for classifying a "face" class, may be completely useless for other classes. If there is a large number of such uninformative kernels, ℓ_1 norm MK-SVM could yield better performance than its ℓ_2 norm counterpart.

4.4 Mediamill and TRECVid07 Results

For the Mediamill and TRECVid07 video classification datasets, we use three kernels, as summarised in Table 3. MAP of the three kernels and that of the three kernel fusion schemes is shown in Table 4. These results are consistent with those obtained on the VOC08 dataset.

Table 3. The 3 kernels used for Mediamill and TRECVid07 datasets

	sampling technique	descriptor	kernel function
kernel 1	Harris-Laplace	SIFT	PMK
kernel 2	Dense	SIFT	PMK
kernel 3	Dense	Colour Histogram	PMK

Table 4. Mediamill and TRECVid07 MAP

	kernel 1	kernel 2	kernel 3	sk-svm uniform	mk-svm ℓ_1 norm	mk-svm ℓ_2 norm
Mediamill	0.311	0.339	0.252	0.383	0.382	**0.394**
TRECVid07	0.291	0.369	0.275	0.432	0.422	**0.443**

4.5 Speed of the Methods

Average train time of the three kernel fusion schemes is shown in Table 5. ℓ_1 norm and ℓ_2 norm MK-SVMs are comparable, and they are both considerably slower than SK-SVM. In terms of test time, however, ℓ_1 norm MK-SVM is advantageous over the ℓ_2 norm version. Since ℓ_1 norm MK-SVM selects a sparse set of kernels, kernels that are not selected do not even need to be computed for the test set. This can be an important factor for the choice of norm in speed-critical applications.

Table 5. Train time (second) of the algorithms

	VOC08	Mediamill	TRECVid07
size of train set	2111	30993	9060
number of kernels	30	3	3
SK-SVM uniform	0.8	161.6	13.5
MK-SVM ℓ_1 Norm	32.4	566.0	44.3
MK-SVM ℓ_2 Norm	25.3	539.5	54.9

5 Conclusions

In this paper we study how the behaviour of MK-SVM is affected by the norm used to regularise the kernel weights. Experiments on three image/video classification datasets show that when kernels carry complementary information of the classification problem, ℓ_2 norm MK-SVM outperforms its ℓ_1 norm counterpart and the uniform weighting scheme. Moreover, through experiments on semi-synthesised data, new insights are gained as to how the choice of regularisation norm should be made.

Acknowledgements

This work has been supported by EU IST-2-045547 VIDI-Video Project.

References

[1] F. R. Bach and G. R. G. Lanckriet. Multiple kernel learning, conic duality, and the smo algorithm. In *International Conference on Machine Learning*, 2004.

[2] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *ACM International Conference on Image and Video Retrieval*, 2007.

[3] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[4] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html.

[5] J. Gemert, J. Geusebroek, C. Veenman, and A. Smeulders. Kernel codebooks for scene categorization. In *European Conference on Computer Vision*, 2008.

[6] K. Grauman and T. Darrell. The pyramid match kernel: Efficient learning with sets of features. *Journal of Machine Learning Research*, 8:725–760, 2007.

[7] M. Kloft, U. Brefeld, P. Laskov, and S. Sonnenburg. Non-sparse multiple kernel learning. In *Proceedings of the NIPS Workshop on Kernel Learning: Automatic Selection of Optimal Kernels*, 2008.

[8] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[9] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2006.

[10] D. G. Lowe. Distincetive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[11] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[12] A. Rakotomamonjy, F. Bach, Y. Grandvalet, and S. Canu. Simplemkl. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

[13] G. Ratsch. Robust boosting via convex optimization. PhD Thesis, University of Potsdam, Potsdam, Germany, 2001.

[14] K. Sande, T. Gevers, and C. Snoek. Evaluation of color descriptors for object and scene recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.

[15] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, 2006.

[16] C. Snoek, M. Worring, J. Gemert, J. Geusebroek, and A. Smeulders. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *ACM Multimedia Conference*, pages 421–430, 2006.

[17] S. Sonnenburg, G. Ratsch, C. Schafer, and B. Scholkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

[18] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1999.