Painting Classification based on Art Styles

Arvind Sowmyan, Jeffrey Geevarghese
Department of Computer Science, University of Massachusetts Amherst
140 Governors Drive, Amherst, MA 01003-9264
{asowmyan, jeffrey}@cs.umass.edu

Abstract

In this report we investigate the task of automatically identifying the underlying genres and art movements of paintings based on image characteristics. We focus on four different classes of paintings collected from several artists and attempt supervised classification on a dataset collected from the internet. We extract a feature set comprising of feature channels based on the color, texture, shape and edge styles of each painting, and test different kernels with them. Subsequently, we experiment with two techniques for multiple kernel learning, namely GMKL and MKBoost, to arrive at a superior kernel function for classification.

1 Introduction

We investigate the task of classifying paintings by art movement based on image characteristics. The task involves identifying specific image features that discriminate well between art movements. Extraction of image features is a well solved problem in the Computer Vision domain, but the use of these features to categorize paintings by art style or artist is a relatively less studied problem. We focus our attention to paintings by several artists from four distinct Art Movements - Abstract Expressionism, Cubism, Renaissance and Romanticism. The first two categories come under the broad category of Modern Art, while the latter two have paintings from the Renaissance and Post-Renaissance period.

It is thought that attributes such as line, shape and color dictate the qualitative aspects of artworks, over content. An understanding of the backgrounds of each of the classes would give us a better understanding of the pertinent features we would need to focus on.

Abstract expressionism is an early fifties school of art centered around New York that focused on spontaneity and automatic drawing, in an attempt to capture imagery out of the subconscious. It is often characterized by reductionist color palettes, vivid, unrestrained brush strokes and a lack of emphasis on form. Cubism is an early 20th-century high-art movement in Paris and Puteaux, focusing on decomposition of imagery into geometric shapes, often viewed from different contrasting perspectives at the same time. Renaissance refers to the classical art style prevalent in Europe in the fourteenth and fifteenth centuries, typified by rich, realistic rendering, well-formed subjects and biblical themes. Romanticism is an eighteenth century euro-american movement that often features vivid palettes and scenes of nature, often with delicately rendered landscapes.

Table 1: Painting Classes

Art Movement	Artists
Abstract Expressionism	Jackson Pollock, Joan Mitchell, Norman Bluhm, Friedel Dzubas, Paul Jenkins, Dan Christensen, Elaine De Koonig, William De Koonig, Perle Fine.
Cubism	El Greco, Tintoretto, Raphael, Hieronymus Bosch, Sandro Botticelli, Giovanni Bellini, Correggio.
Renaissance	Pablo Picasso, Juan Gris, Richard Lindner, Gino Severini, Albert Gleizes, Louis Marcoussis, Andr Lhote, Amedee Ozenfant, Fernand Leger.
Romanticism	John Singleton Copley, Rudolf Von Alt, Fyodor Bronnikov, John Constable, Thomas Cole, Edwin Lord Weeks, John Atkinson Grimshaw.

Figure 1: Samples from the dataset.

2 Related Work

The task of general image classification has been studied extensively in the field of image processing [13]. It has been recognized that for robust scene detection, a rich collection of features are necessary. As such, several feature descriptors have been developed for the purposes of object detection and classification, such as the Scale-invariant feature transform (SIFT) by David Lowe [6]; the Histogram of Oriented Gradients (HOG) by Dalal and Triggs [1]; Saliency by Harel *et.al.* [5]; the GIST or Spatial Envelope of a scene by Torralba *et.al.* [8]. In addition, methods have been proposed for the combination of similarity measures of different image feature channels with combined kernel learning methods, for use with nearest-neighbour-based classification techniques such as SVMs, k-NNs, etc. One notable work in this area was a scheme for Multiple Kernel Learning (MKL) by Vedaldi *et.al.* [11], framing the problem of learning the optimal combination of kernels as a convex optimization task. An alternative approach is to use Boosting techniques for efficient kernel learning, as exemplified by the MKBoost framework devised by Hao *et.al.* [12]. Two works directly related to the task of classification of paintings by art movement are Icoglu *et. al.* [7], a paper examining the use of carefully selected point-features for classification; and Siddiquie *et.al.* [9], a paper based on combining kernels from various feature channels. In this project, we begin with testing the aforementioned point-features on our dataset, then compare them against kernel-classification from various independent feature channels provided by standard image-processing feature descriptors. We finally explore multiple kernel learning by comparing two competing strategies, namely the Generalized Multiple Kernel Learning (GMKL) scheme by Manik Varma *et.al.* [10] and the AdaBoost based MKBoost approach by Hao *et.al.* [12].

3 Dataset

The data was scraped from `www.wikipaintings.org`. The website contains paintings released under a public license. The images are organized broadly by artists and art movements. As mentioned before, we look at paintings from four classes - Abstract Expressionism, Cubism, Renaissance and Romanticism. The final dataset used had 1600 paintings in all - 400 paintings from each of the four art movements, featuring works from a total of 32 artists. To the best of our knowledge, this particular dataset was not used in any other related work. The image resolutions for all the 1600 paintings are not the same, with resolutions varying from 150px to 2800px on either axis. Since most of the features used in this work are fairly scale invariant, this was not an issue. As the images were jpeg images, and the feature extraction methods we describe below work directly on the images, no pre-processing was required

4 Feature Extraction

Based on previous studies on classifying images for different tasks, a large number of features are computed. The features computed are either point features or a distribution of filter responses for each image. The point features can be used directly for classification, while for the latter case, similarity of images is defined as the match between the distributions. We discuss the features used below briefly:

- **Color:** We compute six point features based on the color histogram, namely $[\mu_1 \quad \mu_2 \quad \dots \quad \mu_6]$ based on [3] as mentioned before. The features are computed based on the luminance component of an image in the RGB space, represented by 8 bits (256 grey levels), and the pixels whose values lie in $[0, 64]$ are taken as the dark pixels. We define μ_1 as the fraction of dark pixels in the image, and μ_2 is the *gradient coefficient* calculated from the gradient map of the image. The next two features are based on the histogram values for the i^{th} level. μ_3 is the number of histogram values above a threshold, and μ_4 is the gray level with the maximum histogram value. μ_5 is the sum of the standard deviations of luminance values of different blocks of the image that maintains the aspect ratio. μ_6 is the skewness of the gray level of the image with the Gaussian distribution, under the assumption that gray level distribution does not deviate much from the Gaussian because of lighting conditions. Apart from these point features, we also compute joint histograms of color [13] in CIE L*a*b color space for each image. The histograms have 4, 14 and 14 bins in L, a, b respectively for a total of 784 dimensions. We represent these features as 1×784 row vectors. The distance between these histograms for two images is computed using the χ^2 distance on normalized histograms.
- **Texture:** We compute 14 point texture features, which are known as the *Haralick* features [4]. The basis for these features is the gray-level co-occurrence matrix. Each entry is considered to be the probability that a pixel with value i will be found adjacent to a pixel of value j. The Haralick features are 14 statistics computed from the gray level co-occurrence matrix intended to describe

the texture of the image. For computing these features, we used the code publicly available based on [4].

- **Histogram of Oriented Gradients (HOG):** First, the images are resized to 100×100 pixels. As this method is scale-invariant, this helps in drastically reducing the size of the descriptor without losing much discriminative power. The HOG descriptors are densely extracted from the image grid at steps of 8 pixels, using code available from the repositories of the PASCAL VOC challenge [2]. This gives us a $14 \times 14 \times 31$ descriptor, which we flatten to a 1×6076 vector and compute distances between vectors with a χ^2 or histogram intersection kernel.

- **Shape:** To characterize the general shape or spatial characteristics of a picture (whether it exemplifies an indoor or outdoor scene, with sparse or dense content, etc.), We utilize the GIST descriptor devised by Torralba *et. al.* computed on a bank of 24 Gabor filters over 8 orientations and 4 scales. The descriptor used is a 1×996 feature vector. Learning over this descriptor constitutes learning over the spatial manifold of the image space.

- **Saliency maps:** Saliency maps offer a measure of the most important (or salient) regions of a given scene. The images are once more standardized to a 100×100 pixel size and the saliency descriptor is extracted as a 64×64 square map serialized to a 1×4096 vector using the `simpsal` implementation provided by the authors.

- **Edges:** As classes like cubism are very distinctly defined by the quality of edges and lines present, a point feature representing edge density, and two other histogram features based on line lengths and angles are used. The Canny edge detector is used to obtain line segments over the image. The edge density feature is computed as the ratio of the total number of edges detected over a grid of the image to the number of grid cells. Frequency histograms are also computed for line angles over $180°$ in a 1×200 vector, and for lengths ranging from 0 pixels to over 28 pixels in a 1×30 vector. A Gaussian Radial Basis Function kernel is used over the unnormalized histograms.

- **SIFT:** While content should ideally not be relied upon for the task of identifying art-movements, SIFT is very useful in detecting repeated image content. We compute dense SIFT descriptors using the VL Feat SIFT library over small 64×64 pixel representations of the images resulting in a 128×529 descriptor that is flattened to a 1×67712 vector. A sophisticated treatment of SIFT would involve reducing the descriptor space to a dictionary of visual words which would then be subsequently used for image matching. However, due to computational constraints, we proceed with simpler χ^2 and histogram intersection kernels. Remarkably, the performance of this feature is still shown to be fairly reasonable, indicating that there is still significant descriptive power to be had with recurring content artifacts across different paintings. The above features, namely, Shape, Saliency Maps, Edges and SIFT were implemented based on [13]

5 Classification

The initial classification we attempted used point features computed based on Color alone. The task was run using Weka. We have four classes - Abstract Expressionism (Abs), Cubism (Cub), Renaissance (Ren) and Romanticism (Rom). For these four classes, we did two-way classification as well as four way classification. We tried two-way classification using Nearest Neighbours (7-NN), Naive-Bayes Classifier, polynomial kernels, linear kernels and Random Forests. The highest classification accuracies were given by Random Forest. The results of two-way classification using Random Forests are shown in Table 2.

Table 2: Two-Way Classification using Random Forests

Classes	Accuracy(%)
Abs-Cub	75.7 ± 0.3
Rom-Ren	71.6 ± 0.3
Rom-Cub	81.6 ± 0.2
Abs-Ren	82.7 ± 0.2

The results of this two-way classification can be interpreted in the following way : The classification accuracies between Modern Art {Cubism and Abstract Expressionism} and Renaissance {Renaissance and Romanticism} are on the higher side because of distinct differences between art styles from different centuries. However, classifying between art movements from the same era seems a more difficult classification problem. This is evident from the numbers in Table 2. This can be attributed to the fact that artists often belonged to more than one Art movement, during the course of their career. For example, Picasso was an impressionist in the early part of his career and a Cubist in the latter part. Several motifs prevalent in impressionism, such as outdoor imagery, can be ascribed to the influence of the romantic period. Furthermore, the choice of color palettes appears to be heavily influenced by the era (presumably based on availability or cohesiveness of paints available in the period). Lastly, thematic choices such as the portrayal of a theme of biblical bent can conflate the classes (as does occur on occassion in our romanticism-renaissance dataset).

We attempted four way classification based on these point features in Weka. We tried the Naive Bayes Classifier, SVM-Linear, SVM-RBF, Multilayer Perceptron, 6-NN and Random Forests. The classification accuracies given by these classifiers gives us an idea of the data separability for this problem. As with two way classification, Random Forests perform the best as indicated by the numbers in Table 3.

Table 3: Four Way Classification

Classifier	Accuracy(%)
SVM-RBF	44.2 ± 0.5
Naive-Bayes	44.7 ± 0.4
SVM-Linear	45.2 ± 0.5
Multilayer Perceptron	49.0 ± 0.4
6-NN	51.0 ± 0.4
Random Forest	54.3 ± 0.4

Since we have four classes, the classification is better than random (i.e. randomly assigning a class to an image), however there is much scope for improvement. The classification accuracies for almost all classifiers seem to saturate around 45% - 50%, which tells us that point features used do not have enough discriminative power between these classes.

We now focus on more features based on textures, edge styles, shapes and saliency. Details of these features and their computations were given in the earlier section on Feature Extraction. We noticed that for most of the features, we obtain a distribution of the filter responses for each image. These features are also not of the same size, and often vary in size for each image, hence we cannot treat them as point features directly. However, given these distributions, it is straight forward to compute similarity metrics between them, making the task well posed for kernel based classification. We treat each of the features described earlier as a feature channel and then compute a kernel based on a similarity metric. The similarity metric is not uniform across these feature channels. We have used the Linear Kernel, χ^2 Kernel, Histogram Intersection Kernel and the RBF Kernel. We also use multiple kernels for each feature channel using different similarity metrics. For the feature channels which produce a distribution for each image, we compute the χ^2 and the Histogram Intersection Kernel, while for the point features (which include textures), we compute Linear and RBF Kernels. We compute RBF Kernels for the point features since RBF Kernels are known to perform well when we can assume independence between the features. This is appropriate only for the point features since all the other features are spatial distributions and possess strong local correlations between neighbouring entries. The exact kernels used for each Feature Channel are shown in Table 4.

Table 4: Kernels defined for each feature channel

Feature Channel	Type of Kernel
Point Features	{Linear, RBF}
Color	{χ^2, Histogram Intersection}
HOG	{χ^2, Histogram Intersection}
Edge Styles	{χ^2, Histogram Intersection}
Saliency	{χ^2, Histogram Intersection}
gistGabor	{χ^2, Histogram Intersection}

We can now perform classification based on each of these kernels. The classification accuracies given by each of these kernels can shed some light on the discriminative power of these kernels. We compute these matrices before hand so that they can be used later on in other kernel based techniques as well. The feature kernel computations took approximately 14 hours on an $i7$ dual core machine. The classification results using each of these kernel matrices is discussed in detail in the next section.

Since we have these mutiple kernels computed for different feature channels, we can consider combining these kernels in some way to improve the classification accuracies further. However, learning the optimal combnation of these kernels is far from easy. Multiple Kernel Learning (MKL) is a problem that has gathered a lot of attention in recent years. We look at two types of approaches to solving the MKL problem :

- Solving MKL as an optimization problem.
- Solving MKL using Boosting.

To compare and contrast these two methods as well, we choose two publicly available MATLAB implementations of MKL - Generalized Multiple Kernel Learning(GMKL) [10] is based on Gradient Descent, and the second technique - Multiple Kernel Boosting (MKBoost) [12] is based on AdaBoost. Both methods effectively solve the following optimization problem:

Given M Kernels, identify the optimal combination of these M Kernels, denoted by $\theta = (\theta_1, \theta_2, \ldots, \theta_M)$ *which minimizes the margin based classification error.*

The Generalized Multiple Kernel Learning (GMKL) algorithm that we make use of solves a more general version of this problem, trying to learn non linear combinations of the base kernels as well. The technique is from [10]. The MKBoost implementation works only on binary classification tasks, so we extend the code to perform multiclass classification as well. The method adopted is a one-to-one approach as described below:

- Create N one-vs-all classifiers. That is, for each class i, create a classifier f_i which has $+1$ as the label for all images of class i and -1 otherwise.
- Compute $f(x) = \arg\max_i f_i(x)$.

The results of the above classification methods, starting from classification using a single kernel, to Multiple Kernel Learning using two different techniques are detailed in the next section.

6 Results

We first report the results of kernel based classification based on the kernels computed for each of the feature channels, as well as the results for Multiple Kernel Learning. The GMKL method achieved a classification accuracy of 80%, which is almost the same as the **best** classification accuracies for this task. We see that MKBoost also improves well over the accuracies for the individual kernels. This is a clear illustration of the effectiveness and promise of Multiple Kernel Learning. The results are reported in Table 5. Each data point is an average over 20 iterations of different training and test data sets to reduce training bias. The test set size for all discussions below is 25% of the size of the training set. On initial inspection, there does not seem to be a significant difference between using χ^2-kernels and histogram intersection kernels. However, there appear to be subtle differences in discriminative power between these kernels on inspecting the confusion matrices; histogram intersection kernels offer better average classification accuracies across all classes, whereas χ^2 kernels appear to be biased towards correctly classifying one or two specific classes.

We also examine the *learning rates* for the individual classifers to see if increasing the size of the dataset would improve accuracies further. The learning rates for the classifers based on individual feature channels is shown in Figure 2. It seems beyond a training set of size 600, there is not much improvement in the learning rates of individual kernel based classifers - in fact, some of them showed no improvement beyond a training size of 100! (Color Histograms). From this, it appears that the classification accuracies won't improve with a bigger sized dataset.

It is interesting to examine the coefficients that GMKL learned for the the base kernels. The coefficients learned by GMKL is shown in Figure 3. From Figure 3, it is clear that Point features have the least descriptive power, and Dense SIFT has the most. This explains the low classification accuracies with the point features.

We now look at the confusion matrices for the classification by GMKL, MKBoost and two other kernel classifiers based on individual feature channels. These are shown in Figure 4. As expected, most of

Table 5: Classification Accuracies

Kernel	Accuracy(%)
Color Histogram (hist. int.)	$63.60 \pm 4.8\%$
Color Histogram (χ^2)	$63.40 \pm 2.7\%$
GIST (χ^2)	$60.75 \pm 3.3\%$
HOG (hist. int.)	$59.45 \pm 2.2\%$
HOG (χ^2)	$58.05 \pm 4.2\%$
GIST (hist. int.)	$57.70 \pm 1.6\%$
DenseSIFT (hist. int.)	$55.65 \pm 4.0\%$
DenseSIFT (χ^2)	$55.20 \pm 4.2\%$
Line Angles (χ^2)	$52.15 \pm 3.0\%$
Line Angles (hist. int.)	$51.05 \pm 2.8\%$
Point features (RBF)	$48.35 \pm 3.9\%$
Line Lengths (hist. int.)	$47.80 \pm 2.2\%$
Line Lengths (χ^2)	$44.75 \pm 5.3\%$
Saliency Maps (hist. int.)	$44.35 \pm 2.3\%$
Saliency Maps (χ^2)	$44.35 \pm 3.3\%$
Point features (linear)	$40.65 \pm 3.8\%$
GMKL (16 ker.)	**80.06 ± 2.1 %**
MKBoost + AdaBoost (16 ker.)	$73.41 \pm 1.6\%$

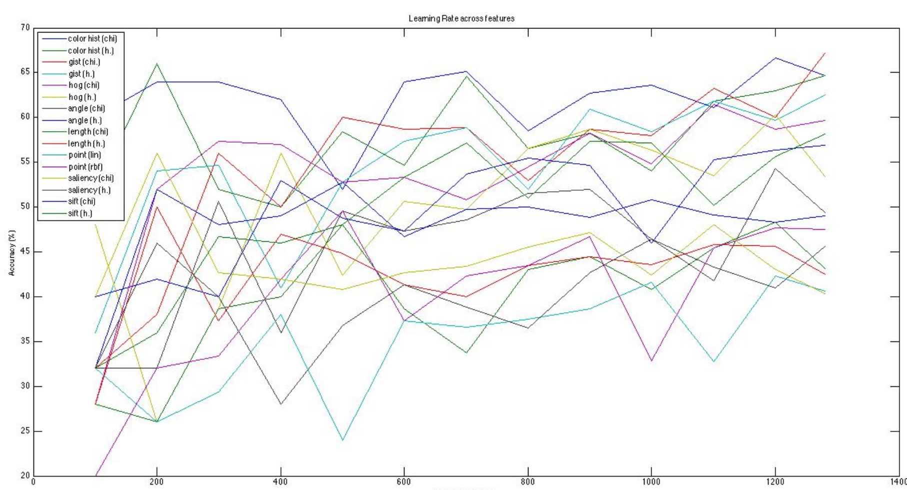

Figure 2: Learning Rates for Individual Kernel Classifers

the *confusion* for the misclassified instances is between the pairs Abstract Expressionism-Cubism and Renaissance-Romanticism. We also noted that certain features are very discriminative for one or two specific classes - for e.g., from the confusion matrices in Figure 4, we can observe that the Point Features Kernel classifies Renaissance (the 3rd class) much better than most other classifiers, and the Saliency

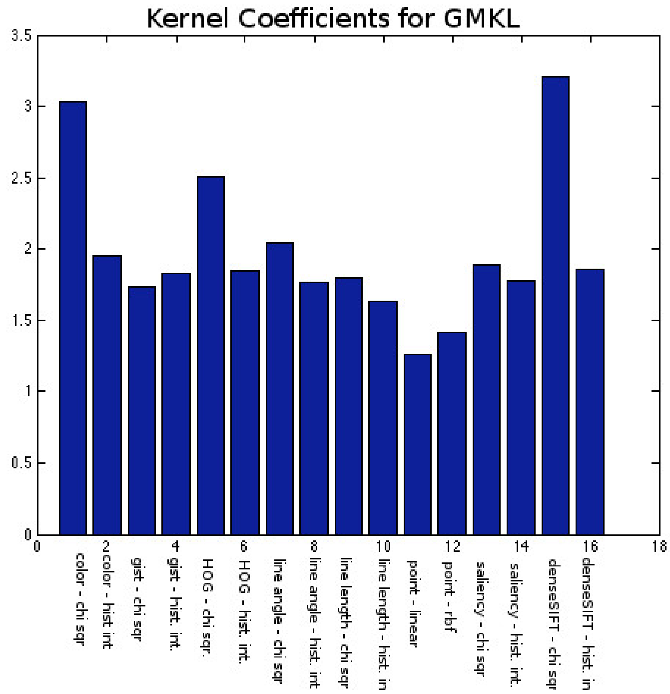

Figure 3: Coefficients of Base Kernels [GMKL].

Feature Kernel classifies Cubism (the 2nd class) far better than all the other classifiers.

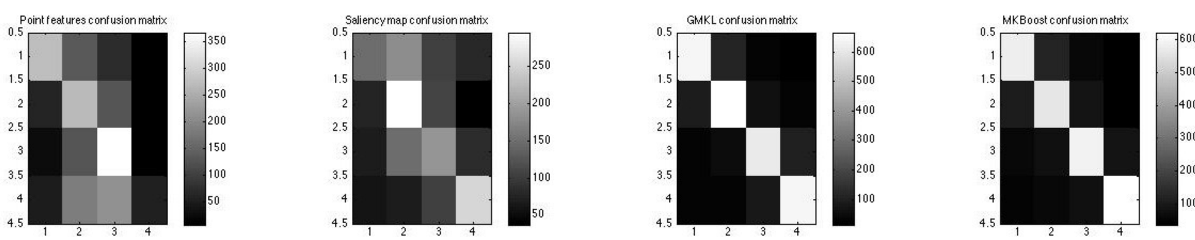

Figure 4: Confusion Matrices: (left to right): Point Feature Kernel, Saliency Map Kernel, GMKL and MKBoost.

7 Conclusion

When we started the work, we never expected to reach a classification accuracy of more than 80% for this task, given that *human performance* for this task will be below 100%. As an image-processing-intensive problem, one would typically expect a large collection of well-tuned and carefully selected features to be the key to performing effective classification in this domain. Yet, it appears that even without taking too much care in our initial choice of kernels, the efficiency and accuracy of Multiple Kernel Learning was astonishing. While we initially experimented with MKL techniques solely on histogram intersection kernels, the inclusion of χ^2-kernels despite the added redundancy of repeated feature channels still offered a good improvement to the MKL output by several points. The accuracies for the individual kernel classifiers never surpassed the performance of Decision Tree Methods like Random Forests, but MKL improved over these by close to 25 points. One possible future work would be to try the Pyramid Match Kernel (PMK) for the individual feature channels, as it lends itself better to bag-of-features formulations of feature descriptors like Sparse SIFT, etc. The system can also be augmented with more Texture Channel features (which are best represented in bag-of-features form and can only be expected to work well with the use of PMKs).

References

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In CVPR, pages I: 886–893, 2005.

[2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. International Journal of Computer Vision, 88(2):303–338, June 2010.

[3] Bilge Günsel, Sanem Sariel, and Oguz Icoglu. Content-based access to art paintings. In ICIP (2), pages 558–561, 2005.

[4] R. M. Haralick. Statistical and structural approaches to texture. Proceedings of IEEE, 67(5):786–804, May 1979.

[5] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In Bernhard Schölkopf, John C. Platt, and Thomas Hoffman, editors, NIPS, pages 545–552. MIT Press, 2006.

[6] D. G. Lowe. Object recognition from local scale-invariant features. In ICCV, pages 1150–1157, 1999.

[7] S Sariel O Icoglu, B Gunsel. Classification and indexing of paintings based on art movements. Proceedings of EUSIPCO, 2004.

[8] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. International Journal of Computer Vision, 42(3):145–175, May 2001.

[9] Behjat Siddiquie, Shiv Naga Prasad Vitaladevuni, and Larry S. Davis. Combining multiple kernels for efficient image classification. In WACV, pages 1–8. IEEE Computer Society, 2009.

[10] Manik Varma and Bodla Rakesh Babu. More generality in efficient multiple kernel learning. In Andrea Pohoreckyj Danyluk, Léon Bottou, and Michael L. Littman, editors, ICML, volume 382 of ACM International Conference Proceeding Series, page 134. ACM, 2009.

[11] Andrea Vedaldi, Varun Gulshan, Manik Varma, and Andrew Zisserman. Multiple kernels for object detection. In ICCV, pages 606–613. IEEE, 2009.

[12] Hao Xia and Steven C. H. Hoi. MKBoost: A framework of multiple kernel boosting. In SDM, pages 199–210. SIAM / Omnipress, 2011.

[13] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In CVPR, pages 3485–3492. IEEE, 2010.