AN IMPROVED AVERAGING COMBINATION METHOD FOR IMAGE AND OBJECT RECOGNITION

Yingli Wei, Wenmin Wang, Ronggang Wang*

School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University
weiyl@sz.pku.edu.cn, *wangwm@ece.pku.edu.cn, rgwang@pkusz.edu.cn

ABSTRACT

A key development in the design of visual object recognition systems is the combination of multiple features. In recent years, various popular optimization based feature combination methods have been proposed in the literatures. However, those methods obtain tiny performance improvement at the cost of enormous computation consumption. In this paper, we propose an improved averaging combination (IAC) method based on simple averaging combination. Firstly, the discriminative power of features are evaluated by dominant set clustering. Then, these features are ranked and added into the averaging combination one by one in descending order. At last, we obtain the best performance improvement of averaging combination by selecting the most powerful features and removing the weak ones. Experimental results on three challenging datasets demonstrate that our method is order of magnitude faster with competitive and even better results than other sophisticated optimization methods, which can be provided as a better baseline method for feature combination.

Index Terms— object recognition, image recognition, feature combination, averaging combination

1. INTRODUCTION

The problem of image and object recognition has been studied with much effort in the past decades. For a given test image, the learned classifier has to decide which class the image belongs to. This is a challenging task, even within datasets of relatively simple images. The main reason lies in that the instances usually have large intra-class diversity and interclass correlation. To overcome the problem of variability, some powerful feature descriptors have been proposed, e.g., SIFT [1], SURF [2], HOG [3]. But it is clear that none of the feature descriptors can be powerful enough to deal with all classes alone. In this case, it is widely accepted to combine the strengths of multiple features to produce better performance.

This project was supported by Shenzhen Peacock Plan (20130408-183003656).
978-1-4799-7082-7/15/$31.00 2015 IEEE

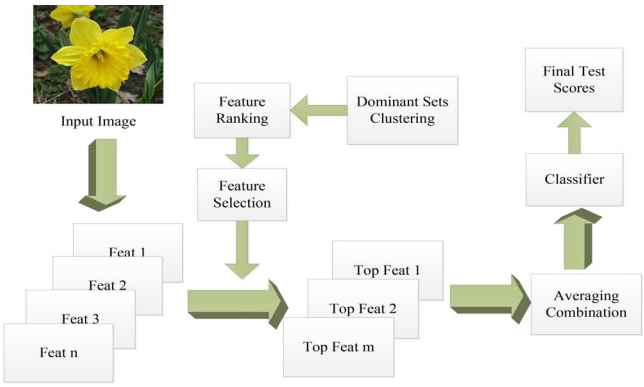

Fig. 1. Illustration of IAC method.

Finding different types of feature combination methods is a recent trend in class-level image and object recognition. There are mainly two types of fusion strategies [4], feature combination and decision fusion. The feature combination method uses several features of all individual classifiers to form a joint feature vector, which is then used in later classification. While decision fusion operates at the decision or the score level, namely, uses a set of classifiers results to form a better and unbiased result. In this paper we focus on feature combination. In the case of SVM classification, the feature combination is translated into kernel combination.

Actually the kernel combination problem is to use the weighted sum of given kernels as the final kernel, i.e. $k^*(x,y) = \sum_{i=1}^{n} w_i k_i(x,y)$, where $w_i (i = 1, ..., n)$ are the weights what we need. The simplest combination method called averaging combination is to assign all participating kernels the same weight and is always been used as the baseline for comparison. Intuitively this is not the optimal method, as it tends to believe that kernels with larger discriminate power should be given larger weights. Based on this intuition, a lot of optimization approaches have been proposed to compute the weight of each kernel in combination, such as MKL [5] and LPBoost [6]. But it has shown that, comparing with averaging combination, the popular MKL-like methods obtain tiny, if any, performance gain at the cost of enormous computation consumption [7].

In order to take advantage of the simple but powerful property of baseline averaging combination, we propose an improved averaging combination (IAC) method which considers the discriminative power of different features and selects the powerful ones for combination. First, the discriminative power of a feature is evaluated by using dominant sets clustering method with its corresponding kernel matrix. The experiments in [8] show that additional features of different orders can obtain diverse results and the best result is obtained in the descending order. So in the next step we add the features into averaging combination in descending order one by one according to their discriminative power. We get the best recognition rate when a sample of the most powerful features is combined. The procedures of improved averaging combination (IAC) method are demonstrated in Fig. 1. Experiments in Section 4 indicate that our approach produces competitive and even better results compared with other existing methods, along with much less computation consumption on various datasets.

The remainder of this paper is organized as follows. Section 2 reviews some of the major research advances in kernel combination, and how they inspire our work in this paper. Section 3 describes our improved averaging combination (IAC) method in detail. Experiments and results comparison are demonstrated in Section 4. Section 5 summarizes the conclusion and mentions future work.

2. RELATED WORK

The two simplest kernel combination methods are averaging combination and product combination. Their final kernel functions are defined as $k^*(x,x^{'}) = \frac{1}{F}\sum_{m=1}^{F} k_m(x,x^{'})$ and $k^*(x,x^{'}) = (\prod_{m=1}^{F} k_m(x,x^{'}))^{\frac{1}{F}}$ respectively. Multiple kernel learning (MKL) is a popular approach which seeks to obtain the best combination performance by jointly optimizing the weights w_i of all kernels in $k^*(x,y) = \sum_{i=1}^{n} w_i k_i(x,y)$ together with the SVM parameters α and b [5, 9, 10]. Unlike the canonical MKL, a new method is to determine the kernel weights based on both kernel functions and the samples. Based on this idea, a sample-specific MKL algorithm is presented in [11]. This algorithm produces some performance improvement at the cost of a large computation load and the risk of over-fitting. In contrast with MKL, the LPBoost algorithm has been presented in [6] where the weights and SVM parameters are trained separately in two steps. The SVMs are trained separately on each kernel in the first step and the weights of all kernels are optimized in the second step. An overview of different methods can be found in the Table 1.

There are still many important problems left unsolved even with various MKL-like kernel combination algorithms. On one hand, existing combination methods are usually computation expensive. The popular MKL-like methods determine the weights of kernels based on the optimization among all participating kernels, which usually leads to enormous computation consumption. On the other hand, the real effectiveness of these algorithms in improving performance has been called in question. In [6, 8] it is noticed that if each participated feature is individually designed to be discriminative, the demonstrated improvement of MKL in literature might also be obtained by the simple averaging combination. With such consideration in mind, we propose an improved averaging combination method which selects the powerful features in combination and sets the weights of the weak features as zero to reduce the effect of weak features. At the same time, we use dominant set clustering to evaluate the discriminative power of features.

Table 1. Comparison of multiclass learning approaches to the feature combination problem in image and object recognization.

Name	Kernel function	References
Averaging	$(\frac{1}{F}\sum_{m=1}^{F} k_m(x))^T\alpha_c + b_c$	
Product	$((\prod_{m=1}^{F} k_m(x)^{\frac{1}{F}})^T\alpha_c + b_c$	
CG-Boost	$\sum_{m=1}^{F} k_m(x))^T\alpha_{c,m} + b_c$	[12]
MKL	$\sum_{m=1}^{F} \beta_m^c(k_m(x)^T\alpha_c + b_c)$	[5, 9, 10]
LP-β	$\sum_{m=1}^{F} \beta_m(k_m(x)^T\alpha_{c,m} + b_{c,m})$	[6]
LP-B	$\sum_{m=1}^{F} B_m^c(k_m(x)^T\alpha_{c,m} + b_{c,m})$	[6]

3. IMPROVED AVERAGING COMBINATION

As we have shown above, the averaging combination method can perform excellently with careful designed features. To explore the full potential of the averaging combination, we present the improved averaging combination (IAC) method to explore the best performance improvement of averaging combination. As shown in Fig. 1, firstly we extract multiple features and then rank the features according to their discriminative power. According to the rank list, we add the features into combination one by one in different orders. At last, we explore the best order and obtain the best performance.

3.1. Problem statement

We start with a brief review of general definition of the feature combination problem [6]. An image $x_i \in \mathbf{X}$ with a class label $y_i \in 1, ..., C$ compose a training set $(x_i, y_i)_{i=1,...,N}$ of N instances. For an image, a set of image features $f_m \in \mathbf{R}^{d_m}$, $m = 1, ..., F$ are extracted, where d_m denotes the dimensionality of the m·th feature. Feature combination is the problem of learning a classification function $y : \mathbf{X} \rightarrow \{1, ..., C\}$ from the features and training sets.

Kernel methods are often used to address the problem of learning a multiclass classifier from training data in computer

vision. Kernel methods utilize kernel functions to define a measure of similarity between pairs of instances. It is useful to associate a kernel to each image feature in the context of feature combination. We simplify the kernel function k between real vectors as

$$k_m(x, x^{'}) = k(f_m(x), f_m(x^{'})), \tag{1}$$

such that the image kernel k_m only considers similarity with regard to image feature f_m. The subscript m of the kernel can be understood as indexing into the set of features. The kernel responding to the $m-$th feature for a given sample $x \in \mathbf{X}$ to all training samples $x_i, i = 1, ..., N$ is denoted as $K_m(x) \in \mathbf{R}^N$ with

$$K_m(x) = [k_m(x, x_1), k_m(x, x_2), ..., k_m(x, x_N)]^T. \tag{2}$$

In case x is the $i-$th training sample, i.e. $x = x_i$, then $K_m(x)$ is simply the $i-$th column of the $m-$th kernel matrix.We study a class of kernel classifiers that aim to combine several kernels into a single model. Since we associate image features with kernel functions, kernel combination is translated into feature combination naturally.

3.2. Evaluation of feature discriminative power

The chance of a kernel matrix to produce a high recognition rate reflects the discriminative power of the kernel and will be used in this paper to define the kernel$^{'}$s accuracy. Inspired by [7], we define the accuracy of a kernel matrix based on dominant sets clustering method as following. Suppose that there are N dominant sets with C classes. The entropy of each dominant set $i(i = 1, ..., N)$ is as follows:

$$H_i = -\sum_{j=1}^{C} \frac{n_{ij}}{N_i} log \frac{n_{ij}}{N_i}, \tag{3}$$

where n_{ij} is the number of elements in dominant set i which belongs to class j, and N_i is the number of all the elements in dominant set i. H_i will be 0 if and only if dominant set i gets a unique label, and the maximum value (namely, log C) will be obtained when it is uniform distribution of all classes in dominant set i.

Finally, an overall accuracy measure of a kernel K is defined in the following way:

$$w_s(K) = \frac{1}{N} \sum_{i=1}^{N} (1 - \frac{H_i}{logC}). \tag{4}$$

Obviously w_s equals to 1 in the ideal case where all the dominant sets are of single class, and becomes 0 when all dominant sets are distributed uniformly of all classes. For each kernel, w_s is calculated to determine the kernel's weight. We select w_s^3 as the weight as it produced the best overall performance [7]. Then it is used to estimate the discriminative power of each feature.

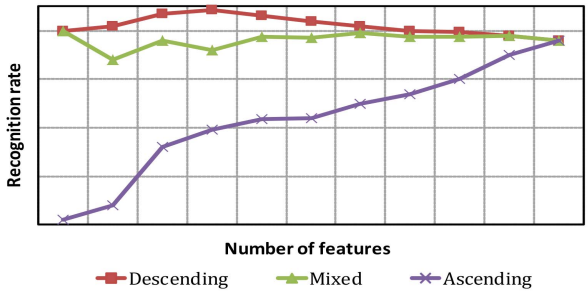

Fig. 2. Illustration of recognition rate curves of different orders.

3.3. Feature selection and combination

In order to get the largest performance improvement of averaging combination, we will figure out the most powerful combination of features. Firstly we sort all these extracted features according to their discriminative power. Then we investigate the influence of the ordering of features being added into averaging combination on combination performance. There exists four different orders [8], which is descending order, ascending order, mixed order and random order. With descending order, we add the features into combination one by one according to the rank list of discriminative power from high to low. The ascending order is just the opposite. With the mixed order, the features are taken into combination one by one from the top and bottom of the rank list alternately. The performance of three different orders are shown in Fig. 2.

We can draw some conclusions from the comparison experiment results. Firstly, the descending order has shown a rise-peak-drop shape of recognition rates with the addition of features into combination and the best performance comes from several most powerful features of combination. Secondly, with the ascending order, the addition of stronger features always improve the combination performance and get the best results with all the features. Thirdly, with mixed order, the addition of weak features tends to decrease the performance. So, in a nutshell, the descending order outperforms the ascending and mixed orders. There also exists another possible order, that is random order. With this order, we randomly select a number of features for averaging combination. The experiments in [13] have shown that random order can not outperform the descending order. Then we can get the conclusion that the best recognition results come from the descending order. Therefore we select the descending order for combination.

Only combination of the most strong features can improve the recognition performance evidently, and the addition of weak features into combination even produce worse results. So a proper stopping criterion is needed to determine when the optimal solution is reached. We consider that if addition

of features does not improve the result any more or even bring down the accuracy of the feature set, the algorithm should be stopped. Then we find out the best recognition accuracy, that is, the peak is reached. The kernel function is as follows:

$$k^*(x, x^{'}) = \xi \sum_{m=1}^{F-M} k_m(x, x^{'}) + \frac{1}{M} \sum_{m=1}^{M} k_m(x, x^{'}), \qquad (5)$$

where F is the number of all the features, the feature M corresponding to the peak. In order to eliminate the bad influence of relatively weak features, we set the weights of features after M as zero. ξ will be infinitely small. In other words, we remove the features which have negative contributions to the averaging combination.

Our method is simple and intuitive as it assigns a meaningful weight to powerful features and removes the ones with bad effects in averaging combination. It is simple in that the weights of weak features are just set as zeros and it is still based on the baseline averaging combination method. In the case of a large set of features are combined, our IAC method implies a less computation consumption than optimization based methods and obtains competitive result. Our method provides a novel approach and a better baseline for feature combination.

4. EXPERIMENTS AND RESULTS

In the experiments, we evaluate our IAC method compared with the state-of-the-art algorithms on several datasets: Scene-15, Caltech-101 and Flower-17. We also quote some results directly from the literature [7, 6]. In all our experiments, the multi-class SVM is trained in the one-versus-all mode and the regularization parameter C is fixed to be 1000. When distances are used to build kernels, kernel matrices are computed as $k(x, y) = exp(-d_0^{-1} d(x, y))$ with d being the distance and d_0 being fixed to the mean of pairwise distances. With all the three datasets, the experimental setups and accuracy measures are selected to be same as in the literature used for comparison.

4.1. Experiment I: Flower-17 dataset

Flower-17 dataset [14] is composed of flower images of 17 categories with 80 images in each category. The dataset comes with three predefined splits into test (17×20 images), train (17×40 images) and validation set (17×20 images). Furthermore the authors of [14, 15] have provided seven precomputed distance which used for the experiments. Each matrix is computed using a different feature type, namely three matrices derived from colour, shape and texture vocabularies, clustered HSV values, SIFT features on the foreground region and boundary.

The results using a SVM with single kernel only and the performance comparison with combination is shown in Fig. 3.

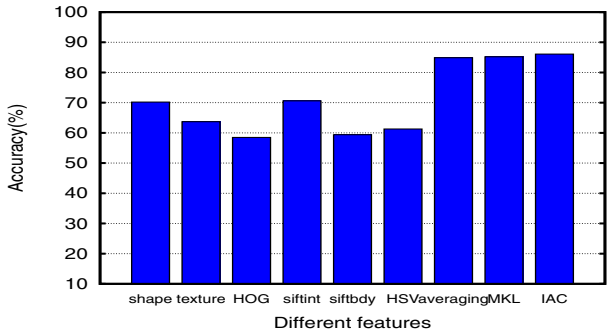

Fig. 3. Performance comparison of individual features and combination methods on Flower-17.

It is evident that feature combination methods dramatically improve the classification performance. The overall recognition rate and comparison with the literature are shown in Table 2. Our method has produced the best result on Flower-17 dataset, better than the results obtained with MKL in [15] and with LP-β method in [6]. We also note that our method is magnitude faster than other optimization based methods.

Table 2. Classification rate(%) and comparison on Flower-17 and the total time for model selection, training and testing in seconds.

Method	Accuracy	Time
product	85.5 ± 1.2	2
averaging	84.9 ± 1.9	10
CG-Boost	84.8 ± 2.2	1225
MKL(simple) [15]	85.2 ± 1.5	152
LP-β [6]	85.5 ± 3.0	80
LP-B [6]	85.4 ± 2.4	98
IAC	**86.1 ± 1.3**	32

4.2. Experiment II: Scene-15 dataset

Scene-15 dataset [16, 17] contains totally 4485 images from 15 categories with 200 to 400 images in each category. Following the experimental setup in [17], i.e., 100 randomly selected images per class as training and all the others as testing, and report the mean recognition rate per class. In the following we briefly describe the image features used for the experiments in Scene-15 dataset.

HOG Shape Descriptor. Oriented and unoriented HOG descriptors [3] are constructed. The oriented histogram contains 40 bins and is denoted by $hog360$, the unoriented contains 20 bins and is denoted by $hog180$.

Bag-of-SIFT-Descriptor. SIFT descriptors [1] are extracted on a regular grid on the image with a spacing of 10 pixels and for the four different radii $r = 4, 8, 12, 16$. The

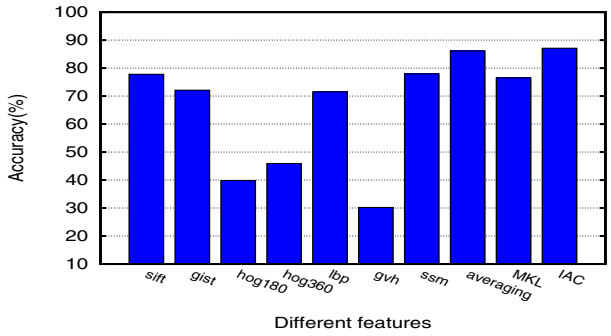

Fig. 4. Performance comparison of individual features and combination on Scene-15.

descriptors are extracted of codebook sizes (500 elements) in gray spaces. The descriptors are denoted by $sift$.

Locally Binary Patterns. The 256-bin histograms of the basic locally binary patterns (LBP) [14] are extracted and clustered to create a descriptor for one image. The descriptor is referred to as lbp.

Gist Descriptor. The gist descriptors [18] are extracted and denoted as $gist$.

Self-similarity Descriptor. The self-similarity descriptors [19] of 30 dimensions (10 orientations and 3 radial bins) with spacing of 5 pixels are quantized to build a 500-bin vocabulary. The histograms are then built and denoted by ssm.

Gabor filters. Gabor filters [20] are used to build histograms (500 bins). The descriptor is referred to as gab.

Gray Value Histogram. We use the simple 64-bin gray value histograms. This gray value histogram is denoted by gvh.

We use these features to build kernels with χ^2 distance and obtained 8 kernels. Here the selection of χ^2 distance in building kernels is based on the work in [6, 8], where χ^2 based kernel was shown to outperform some other kernels. To demonstrate the effects of individual feature on combination performance, we show the performance comparison of individual features with combination in Fig. 4. Table 3 shows the classification results of our algorithm and comparison with the literature. From Table 3 we observe that our IAC method outperforms the state-of-the-art result of 86.7% reported in [21] and other results. The result further confirms the effectiveness of our method.

It is evident from Fig. 3 and Fig. 4 that when the performance variance of individual features is small, the advantage of IAC method over averaging combination is not so obvious. According to the theory of our IAC method, if all participated kernels are of similar discriminative power, the IAC is turned into the ordinary averaging combination. In such case, it also performs well and this is consistent with the trend observed in Gehler and Nowozin [6].

Table 3. Classification rate(%) comparison on Scene-15.

Method	Accuracy
Best single	79.6 ± 0.7
Average	86.1 ± 0.5
CV-weight	86.4 ± 0.6
MKL(simple) [15]	76.5 ± 0.6
[21]	86.7 ± 0.4
[17]	81.4 ± 0.5
[22]	73.4 ± 1.0
IAC	$\mathbf{87.1 \pm 0.2}$

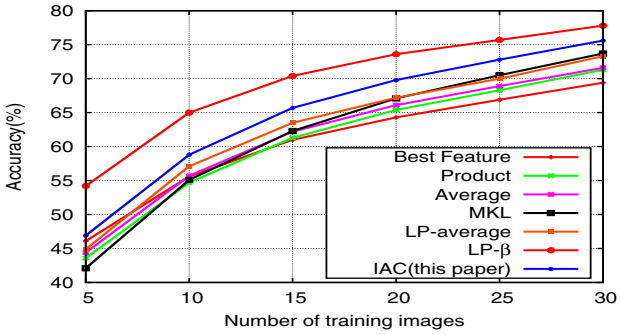

Fig. 5. Classification rate(%) comparison on Caltech-101.

4.3. Experiment III: Caltech-101 dataset

Caltech-101 [23] is a popular dataset for general image classification, which includes 101 categories of object with strong shape variability. In the experiment, we follow the conventional setup for Caltech-101, specifically, for each of the 102 classes, the number of training images is varied using 5, 10, 15, 20, 25, 30 images per category for training and up to 50 images per category for testing. For comparison, we adopt the same features as in Gehler and Nowozin [6] to build kernel matrices and in total 39 kernels are used in combination.

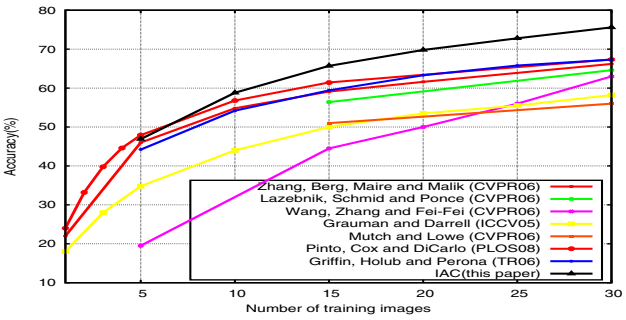

Fig. 6. Caltech-101 comparison to literature

Fig. 5 shows the result of combining 39 kernels corresponding to different image features and comparisons. In this

experiment our weighting scheme is outperformed by the LP-β method, but it compares favorably with all the other combination methods, including LP-average and MKL. In Fig. 6 we compare IAC to several other results published in the literature. Note that our IAC method yields better performance than all competitors.

These comparisons indicate that our improved averaging combination method based on the baseline method can be as powerful as more sophisticated optimization methods. Noticing its good performance and advantage in computation complexity, we think that we have proposed an novel approach for feature combination and the method can be used as a better benchmark combination method.

5. CONCLUSION

In this paper, we proposed a simple yet effective feature combination method for image and object recognition. Based on the ordinary averaging combination method, the improved averaging combination discovers the most powerful combination of features and gets the largest improvement of recognition performance. It selects the most strong features and removes the weak ones to improve the performance. Extensive experiments on several datasets have demonstrated systematic improvement over benchmark combination methods. The proposed method performs competitive and even better results with much smaller computation consumption compared with more sophisticated, state-of-the-art methods and can be used as a better baseline combination method.

6. REFERENCES

[1] David G Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, pp. 91–110, 2004.

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "Surf: Speeded up robust features," in *ECCV,2006*.

[3] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *CVPR,2005*.

[4] Sergey Tulyakov, Stefan Jaeger, Venu Govindaraju, and David Doermann, "Review of classifier combination methods," in *MLDAR*, pp. 361–386. 2008.

[5] Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan, "Learning the kernel matrix with semidefinite programming," *JMLR*, pp. 27–72, 2004.

[6] Peter Gehler and Sebastian Nowozin, "On feature combination for multiclass object classification," in *Computer Vision*, 2009, pp. 221–228.

[7] Jian Hou and Marcello Pelillo, "A simple feature combination method based on dominant sets," *Pattern Recognition*, pp. 3129–3139, 2013.

[8] Jian Hou, Bo-Ping Zhang, Nai-Ming Qi, and Yong Yang, "Evaluating feature combination in object classification," in *Advances in Visual Computing*, pp. 597–606. 2011.

[9] Ankita Kumar and Cristian Sminchisescu, "Support kernel machines for object recognition," in *ICCV,2007*.

[10] Yen-Yu Lin, Tyng-Luh Liu, and Chiou-Shann Fuh, "Local ensemble kernel learning for object category recognition," in *CVPR,2007*.

[11] Mehmet Gönen and Ethem Alpaydin, "Localized multiple kernel learning," in *ICML*, 2008.

[12] Jinbo Bi, Tong Zhang, and Kristin P. Bennett, "Column-generation boosting methods for mixture of kernels," KDD,2004.

[13] Jian Hou, Wei-Xue Liu, and Hamid Reza Karimi, "Exploring the best classification from average feature combination," in *Abstract and Applied Analysis*, 2014.

[14] M-E Nilsback and Andrew Zisserman, "A visual vocabulary for flower classification," in *CVPR,2006*.

[15] M-E Nilsback and Andrew Zisserman, "Automated flower classification over a large number of classes," in *ICVGIP,2008*.

[16] Li Fei-Fei and Pietro Perona, "A bayesian hierarchical model for learning natural scene categories," in *CVPR,2005*.

[17] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR,2006*.

[18] Aude Oliva and Antonio Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *IJCV*, pp. 145–175, 2001.

[19] Eli Shechtman and Michal Irani, "Matching local self-similarities across images and videos," in *CVPR,2007*.

[20] Manik Varma and Andrew Zisserman, "A statistical approach to texture classification from single images," *IJCV*, pp. 61–81, 2005.

[21] Liefeng Bo, Xiaofeng Ren, and Dieter Fox, "Kernel descriptors for visual recognition," in *ANIPS*, 2010, pp. 244–252.

[22] Hongping Cai, Fei Yan, and Krystian Mikolajczyk, "Learning weights for codebook in image classification and retrieval," in *CVPR,2010*.

[23] Li Fei-Fei, Rob Fergus, and Pietro Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *CVIU*, pp. 59–70, 2007.