Beyond Spatial Pyramids: A New Feature Extraction Framework with Dense Spatial Sampling for Image Classification

Shengye Yan[1], Xinxing Xu[1], Dong Xu[1], Stephen Lin[2], and Xuelong Li[3]

[1] School of Computer Engineering, Nanyang Technological University
[2] Microsoft Research Asia
[3] OPTIMAL, State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences
{syyan,xuxi0006,dongxu}@ntu.edu.sg,
stevelin@microsoft.com, xuelong_li@opt.ac.cn

Abstract. We introduce a new framework for image classification that extends beyond the window sampling of fixed spatial pyramids to include a comprehensive set of windows densely sampled over location, size and aspect ratio. To effectively deal with this large set of windows, we derive a concise high-level image feature using a two-level extraction method. At the first level, window-based features are computed from local descriptors (e.g., SIFT, spatial HOG, LBP) in a process similar to standard feature extractors. Then at the second level, the new image feature is determined from the window-based features in a manner analogous to the first level. This higher level of abstraction offers both efficient handling of dense samples and reduced sensitivity to misalignment. More importantly, our simple yet effective framework can readily accommodate a large number of existing pooling/coding methods, allowing them to extract features beyond the spatial pyramid representation.

To effectively fuse the second level feature with a standard first level image feature for classification, we additionally propose a new learning algorithm, called Generalized Adaptive ℓ_p-norm Multiple Kernel Learning (GA-MKL), to learn an adapted robust classifier based on multiple base kernels constructed from image features and multiple sets of pre-learned classifiers of all the classes. Extensive evaluation on the object recognition (Caltech256) and scene recognition (15Scenes) benchmark datasets demonstrates that the proposed method outperforms state-of-the-art image classification algorithms under a broad range of settings.

Keywords: Image Classification, Spatial Pyramid, Sliding Window, Multiple Kernel Learning, Adapted Classifier.

1 Introduction

A well-established approach to image classification was introduced in [1], where an image is subdivided into increasingly finer regions according to a spatial pyramid representation (SPR), and then a Bag-of-Features (BoF) [2, 3] is computed

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part IV, LNCS 7575, pp. 473–487, 2012.

within each of the subregions. In the past few years, many sophisticated feature extraction techniques have been extended from this framework [4–10].

While the spatial pyramid representation has become widely used in image classification, the grid cells within a pyramid correspond to a rather limited set of spatial regions where features are defined: the cells have a fixed aspect ratio; their areas vary only by multiples of four; and their locations must align with a grid. Many of the possible spatial regions are excluded, though some of them may provide important discriminative information.

Motivated by the success of sliding windows in object detection [11], we seek in this paper a general framework for image classification that accounts for a comprehensive set of windows densely sampled with respect to location, size, and aspect ratio, while allowing existing methods for encoding and pooling to be incorporated. However, two important issues arise from a direct approach. One is that the feature vector would become extremely large, since it is formed as a concatenation of features from each of the windows. Such large feature vectors would make image classification computationally very inefficient. The other issue that seriously impairs this approach is that different images are often not aligned with each other in image classification scenarios. Feature vectors with a strong spatial structure can therefore be detrimental when corresponding features do not coincide in image position.[1]

To avoid these issues, we propose a simple but effective image feature derived from densely sampled windows that is relatively compact and less sensitive to misalignment. This feature represents an image-level abstraction of the window-based features used in [1]. It is obtained via a two-level feature extraction method in which the first level computes window-based features from local descriptors (e.g., SIFT, spatial HOG, LBP), and the second level repeats the encoding and pooling procedure on the window-based features to acquire the new image feature. Feature pooling over the image yields a feature vector with the same number of elements as the codebook. Moreover, as in window-based features [1], exact positional information within the image is left out of the image feature in the same manner. This image feature implicitly captures useful spatial information, and will be shown to enhance classification performance when added to SPR. Furthermore, various pooling/coding techniques [6–10, 12] which extract features only from fixed spatial pyramids can be easily extended to go beyond the spatial pyramid representation within our proposed feature extraction framework.

For SVM classification, we propose a new learning method called Generalized Adaptive ℓ_p-norm Multiple Kernel Learning (GA-MKL), which is motivated by the recent success of MKL methods for various vision applications, such as object categorization [13, 14] and action recognition [15]. GA-MKL allows for different features such as our new second level feature and the standard first level feature to be effectively combined for classification. Moreover, GA-MKL takes advantage of pre-learned classifiers of other classes, based on the intuition that some classes

[1] We note that certain image categories tend to share a common spatial arrangement, such as people located in the middle of images, which works to the benefit of features based on SPR.

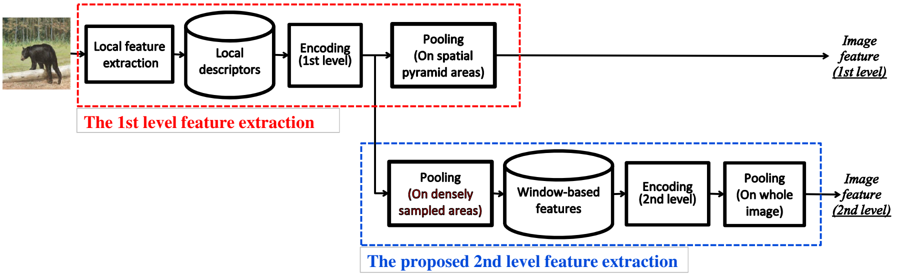
Local feature extraction
Local descriptors
Encoding (1st level)
Pooling (On spatial pyramid areas)
Image feature (1st level)
The 1st level feature extraction
Pooling (On densely sampled areas)
Window-based features
Encoding (2nd level)
Pooling (On whole image)
Image feature (2nd level)
The proposed 2nd level feature extraction

Fig. 1. Overview of the proposed two-level feature extraction framework

may share common information that can benefit each other. For example, classes like "Swan", "Duck" and "Goose" may share the same background of "Water" and similar components like beaks. Therefore it may be beneficial to train an adapted classifier for "Swan" that leverages on pre-learned classifiers for "Duck" and "Goose". GA-MKL takes advantage of this by learning an adapted classifier using multiple sets of base kernels and multiple sets of pre-learned SVM classifiers from other classes.

This work provides the first practical unsupervised feature extraction framework for going beyond spatial pyramids with densely sampled windows in image classification, in a general manner that easily accommodates existing encoding and pooling schemes. Through extensive experiments conducted on two widely-used benchmarks – Caltech256 [16] and 15Scenes [1, 17, 18] – we demonstrate the effectiveness of our feature extraction framework based on the second level feature and leveraging pre-learned classifiers from other classes through GA-MKL. These results show that our work consistently outperforms the state-of-the-art over a broad range of test cases.

2 Related Work

Different variants of the spatial pyramid representation have been employed for image classification. Though the original work of [1] found no performance improvement with pyramids expanded beyond the conventional three levels, others have reported better classification when a fourth level is included [14, 19]. In [20], adding overlapping spatial areas to the non-overlapped grid for the second and third levels was shown to capture more spatial information. The novel spatial pyramid layout used by the winner of VOC 2007 [21] has been adopted by many recent state-of-the-art methods [22–24]. In [25], fan-shaped areas are used in place of rectangular spatial areas in SPR. In contrast to these aforementioned methods, our work effectively and efficiently processes a complete set of rectangular windows, instead of a handcrafted subset.

In feature extraction, spatial information has been accounted for on two levels: in the local descriptor (such as the SIFT feature) and in the code of the local descriptor (as done in SPR). Kulkarni et al. [26] used affine SIFT to handle pose and viewpoint variance. Boureau et al. [4] proposed a mid-level feature based on sparse coding on local groups of SIFT features, instead of individual ones. They also presented a pooling scheme that can effectively handle large codebooks [12]. Feng et al. [10] proposed geometric ℓ_p pooling that places different importance on different geometric positions. Yang et al. [5] took advantage of spatial pyramid co-occurrence for overhead aerial imagery. For object recognition, Bosch et al. [27] utilized a region of interest detection procedure before applying BoW feature extraction. Our method differs from these techniques by introducing a higher level of feature that accounts for densely sampled windows of any location, size and aspect ratio.

The work in [28, 29] proposed to extract new types of higher level feature representations to exploit spatial or spatial-temporal co-occurrences beyond local descriptors. In both works, for final classification, their proposed features are pooled to obtain a global histogram for the whole image (i.e., a 1x1 spatial pyramid). In contrast, our method goes beyond spatial pyramids such that the final feature is extracted from windows densely sampled over location, size and aspect ratio. Jia et al. [30] also presented a method to go beyond spatial pyramids, by learning optimal pooling parameters for an over-complete set of receptive field candidates.

Another stream of research takes advantage of attribute or object level classifiers to extract high level features directly [31, 32] or use them indirectly for visual word disambiguation [33]. All these methods involve supervised learning of attribute classifiers using an extra training set collected from Google search or other sources. By contrast, our feature extraction framework does not use any extra training set, and the entire feature extraction process is unsupervised.

Several feature extraction techniques have been presented for purposes other than image classification. Duchenne et al. [34] proposed a graph-matching method that matches corresponding object points in different images for object classification. Boiman et al. [35] applied the nearest-neighbor classifier directly on different categories of SIFT features. Gehler et al. [36] combined different kinds of features and showed high performance with multiple kernel combinations. Bo et al. [37] framed image recognition as an image matching problem and solved it by kernel matching.

Recent work [15, 38] demonstrated that it is generally beneficial to utilize the pre-learned classifiers from other classes for event/action recognition. In contrast to the ℓ_1-norm constraint used in existing methods like [15, 38], in GA-MKL, we utilize the more general ℓ_p-norm constraint (*e.g.*, $p = 2$ in this work) which can preserve *complementary and orthogonal information* [39]. This is particularly important when base kernels contain complementary information as in our two level feature extraction framework. Furthermore, GA-MKL also learns the weights for multiple sets of pre-learned classifiers. Using the pre-learned classifiers for other classes also distinguishes GA-MKL from the existing ℓ_p-MKL technique.

3 Two-Level Feature Extraction

3.1 First Level Image Feature

For the first level, we employ BoF image feature extraction, which consists of four key components – local feature extraction, dictionary learning, feature encoding and feature pooling – which are illustrated in the upper part of Fig. 1. This is performed using the SPR framework of [1]. First, local descriptors such as SIFT are extracted from image patches. A visual word dictionary is then generated from these local features via clustering. This visual dictionary thereafter is used to encode each local feature into a coded vector by soft assignment [9]. Next, max pooling [6] is performed on the coded vectors in each window of the spatial pyramid. We note that other advanced encoding [6–9] or pooling [10, 12] methods can be readily used in our framework to improve classification performance. In this work, we take soft assignment [9] and max pooling [6] as an example to illustrate our framework because of their efficiency and reasonable effectiveness.

A spatial pyramid subdivides the input image into a sequence of grids with incrementally finer non-overlapping regions of the same size. As illustrated at the left of Fig. 2, the grid at level l has 2^l cells along each dimension, for a total of $D = 2^l \times 2^l$ cells. The vectors generated for each window by max pooling are all concatenated to form the first level image feature. This feature extraction procedure is the same as that used in [9].

3.2 Second Level Image Feature

Dense Sampling of Spatial Areas. The conventional spatial pyramid representation can greatly boost the performance of image classification, and with our second level image feature we aim to go beyond SPR by transplanting the idea of sliding windows [11] into image classification. Towards this end, we sample the spatial areas densely with respect to location, aspect ratio and size. This is achieved as follows. Suppose each spatial area is denoted by $Area(x, y, w, h)$, where (x, y) denotes the image position of the upper-left corner of the window, and (w, h) denotes the window width and height. All 4-tuples of $Area(x, y, w, h)$ are enumerated to obtain a comprehensive set of spatial areas.

The dense sampling procedure is illustrated in the right part of Fig. 2. For each window size $(\hat{w}, \hat{h})$, each image position $(\hat{x}, \hat{y})$ is scanned as shown by the red arrows. The window is iteratively shifted from left to right (X-direction), and from top to bottom (Y-direction). Sampling of different window widths and heights is illustrated along the black horizontal and vertical axes, respectively. The size and aspect ratio of windows are shown at the top-left of each image.

By dense sampling, windows that tightly bound an object or other potentially meaningful image patch are captured. This is shown by yellow rectangles in Fig. 2 for the bear's head and leg, and also a log on the ground.

In practice, we do not exhaustively sample the spatial areas pixel by pixel. Our implementation uses a step size of 30 pixels for x, y, w, h.

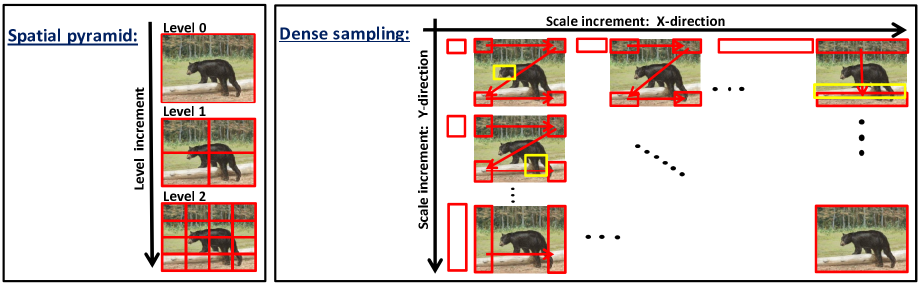

Fig. 2. Illustration of dense spatial sampling. The left side shows spatial pyramid sampling in [1]. The right side shows dense sampling as done in our proposed framework.

Second Level Coding and Pooling. We now have a set of spatial areas from dense sampling. Feature pooling is then performed on each spatial area to produce a feature vector which we refer to as a window-based feature. From the window-based features (one per spatial area), we compute an image feature vector that is the final output of feature extraction.

To go from window-based features to the final image feature, we propose to do a second level of feature extraction. This second level differs from the first level in that clustering is carried out on the window-based features instead of local SIFT descriptors. The secondary codebook learned in this clustering step is used to encode the window-based features. Finally, pooling of the encoded window-based features is done over the entire image to determine the image feature vector, which contains the same number of elements as the secondary codebook. As mentioned previously, we use soft assignment [9] and max pooling [6] in this work, but any encoding and pooling methods may be used instead.

Similar to the way the first level image feature relates each pyramid window to SIFT codewords, the second level feature relates the entire image to window-based codewords. The window-based codewords essentially represent a set of "window clusters" that each have similar first level feature content. These "window clusters" can be considered as a form of higher level SIFT-based feature defined over larger areas. We will later show in the experiments that this higher level abstraction of standard window descriptors provides a useful complement to first level image features.

3.3 Extension to Multiple Local Descriptors

The two-level feature extraction framework offers the generality to incorporate any kind of local descriptor, such as SIFT [40], Spatial HOG [41, 42] and LBP [43]. Two-level feature extraction for spatial HOG follows the exact same procedure as for SIFT. For LBP, histograms are extracted at the first level feature extraction, then LBP histograms are further processed by the proposed second level feature extraction.

4 Generalized Adaptive ℓ_p-norm Multiple Kernel Learning

In the following, we define the ℓ_p-norm of the M dimensional vector $\mathbf{d}$ as $||\mathbf{d}||_p = (\sum_{m=1}^{M} d_m^p)^{1/p}$, and specially denote the ℓ_2-norm of $\mathbf{d}$ simply as $||\mathbf{d}||$ for brevity. We also use the superscript $'$ to signify the transpose of a vector, and denote the element-wise product between two vectors $\boldsymbol{\alpha}$ and $\mathbf{y}$ as $\boldsymbol{\alpha} \odot \mathbf{y} = [\alpha_1 \mathrm{y}_1, \cdots, \alpha_l \mathrm{y}_l]'$. Moreover, $\mathbf{1} \in \mathbb{R}^l$ denotes an l dimensional vector with all elements of 1, and the inequality $\mathbf{d} = [d_1, \ldots, d_M]' \geqslant 0$ indicates that $d_m \geqslant 0$ for $m = 1, \ldots, M$.

Multiple Kernel Learning (MKL) has been widely utilized to fuse different types of visual features. The traditional ℓ_1-norm MKL selects a very sparse set of base kernels, which may discard some useful information. The recent ℓ_p-norm Multiple Kernel Learning (ℓ_p-MKL) [39] utilizes the more general ℓ_p-norm constraint (*e.g.*, $p = 2$ in this work) for the kernel coefficients, which can preserve *complementary and orthogonal information* [39] in contrast to ℓ_1-norm MKL.

In our work, we wish to additionally take advantage of existing SVM classifiers trained from different types of visual features for different classes. Our intuition is that different classes may share some common information that benefits others. We thus propose a new learning method called Generalized Adaptive ℓ_p-norm Multiple Kernel Learning (GA-MKL) to learn a robust adapted classifier that not only fuses different types of visual features (*e.g.* first and second level image features) but also incorporates pre-learned classifiers trained on different types of features for all of the classes.

We consider one-versus-rest classification in this work. For any given class, let us denote the training set as $\{(\mathbf{x}_i, \mathrm{y}_i)|_{i=1}^{l}\}$ where $\mathbf{x}_i$ is the i^{th} training image with $\mathrm{y}_i \in \{+1, -1\}$ being the corresponding label. Suppose that we have a total number of H classes and S sets of pre-learned classifiers $\{f_s^1(\mathbf{x}), \cdots, f_s^H(\mathbf{x})\}|_{s=1}^{S}$, each set of which can be learned from some kind of image representation (In this work, different representations are coming from different types of visual features). Motivated by [38], we assume that the decision function for the new classifier is a linear combination of all the pre-learned classifiers with a perturbation function modeled by using the original visual feature, and define the decision function as

$$f(\mathbf{x}) = \sum_{s=1}^{S} \boldsymbol{\beta}_s' \boldsymbol{f}_s(\mathbf{x}) + \Delta f(\mathbf{x}), \tag{1}$$

where $\boldsymbol{f}_s(\mathbf{x}) = [f_s^1(\mathbf{x}), \cdots, f_s^H(\mathbf{x})]'$ is the s^{th} decision value vector for the input image $\mathbf{x}$ from the pre-learned classifiers, $\boldsymbol{\beta}_s = [\beta_s^1, \cdots, \beta_s^H]'$ is the corresponding weight vector to be optimized, and $\Delta f(\mathbf{x})$ can be any perturbation function from the original visual feature space. If we utilize the decision function of Multiple Kernel Learning as the perturbation function, and assume that a total number of M base kernels are used, then $\Delta f(\mathbf{x}) = \sum_{m=1}^{M} d_m \mathbf{w}_m' \varphi_m(\mathbf{x}) + b$, where $\varphi_m(\cdot)$ is the mapping of the m^{th} base kernel, $\mathbf{d} = [d_1, \ldots, d_M]'$ is the vector of base kernel coefficients, and $\mathbf{d}$,$\mathbf{w}_m|_{m=1}^{M}, b$ are the variables of the MKL.

The new adapted classifier $f(\mathbf{x})$ can be learned by minimizing the following objective function:

$$\min_{d_m,\mu_s} \min_{\mathbf{v}_m,b,\xi_i,\boldsymbol{\beta}_s} \frac{1}{2}\sum_{s=1}^{S}\frac{\|\boldsymbol{\beta}_s\|^2}{\mu_s}+\frac{\lambda}{2}\sum_{s=1}^{S}\mu_s^2+\underbrace{\frac{1}{2}\sum_{m=1}^{M}\frac{\|\mathbf{v}_m\|^2}{d_m}+C\sum_{i=1}^{l}\xi_i}_{\mathbf{J}(\Delta f)} \tag{2}$$

$$\text{s.t. } \mathrm{y}_i\left(\sum_{s=1}^{S}\boldsymbol{\beta}_s'\boldsymbol{f}_s(\mathbf{x}_i)+\sum_{m=1}^{M}\mathbf{v}_m'\varphi_m(\mathbf{x}_i)+b\right)\geqslant 1-\xi_i, \xi_i\geqslant 0, i=1,\cdots,l,$$
$$\mathbf{d}\geqslant 0, ||\mathbf{d}||_p^2\leqslant 1, \boldsymbol{\mu}\geqslant 0,$$

where $C > 0$ is the MKL regularization parameter, $\mathbf{v}_m = d_m\mathbf{w}_m$, $\mathbf{J}(\Delta f)$ is the MKL structural risk functional, and $p \geqslant 1$ is the norm parameter for the base kernel coefficients introduced in ℓ_p-MKL [39]. Besides the structural risk term $\mathbf{J}(\Delta f)$ for standard MKL, the coefficients $\boldsymbol{\beta}_s|_{s=1}^S$ for the pre-learned classifiers are also penalized as $\|\boldsymbol{\beta}_s\|^2|_{s=1}^S$. Considering that the pre-learned classifiers from different visual features have different classification capacity, we further introduce an intermediate vector $\boldsymbol{\mu} = [\mu_1, \cdots, \mu_S]'$ to control the contributions of the penalty terms from different pre-learned classifier sets. The regularization term $\frac{\lambda}{2}\sum_{s=1}^S \mu_s^2$ with regularization parameter $\lambda > 0$ is included to avoid a trivial solution for $\boldsymbol{\mu}$. In this way, we not only fuse different types of visual features but also utilize the pre-learned classifiers of all the classes.

Since the optimization problem in (2) is convex w.r.t. $\mathbf{v}_m, b, \xi_i, \boldsymbol{\beta}_s, \mathbf{d}, \boldsymbol{\mu}$, the global optimum can be obtained by using the block-wise coordinate descent algorithm [39]. We thus alternatively optimize these variables with the following two steps.

Optimize $\mathbf{v}_m, b, \xi_i, \boldsymbol{\beta}_s$ with Fixed $\mathbf{d}, \boldsymbol{\mu}$: With fixed $\mathbf{d}, \boldsymbol{\mu}$, the problem in (2) is a convex problem w.r.t. $\mathbf{v}_m, b, \xi_i$ and $\boldsymbol{\beta}_s$. By introducing the non-negative Lagrangian multipliers $\alpha_i|_{i=1}^l$, the dual can be derived as follows:

$$\max_{\boldsymbol{\alpha}} \ \boldsymbol{\alpha}'\mathbf{1}-\frac{1}{2}(\boldsymbol{\alpha}\odot\mathbf{y})'\left(\sum_{m=1}^{M}d_m\mathbf{K}_m+\sum_{s=1}^{S}\mu_s\mathbf{F}_s\right)(\boldsymbol{\alpha}\odot\mathbf{y}) \tag{3}$$
$$\text{s.t. } \boldsymbol{\alpha}'\mathbf{y}=0, 0\leqslant\boldsymbol{\alpha}\leqslant C,$$

where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_l]'$, $\mathbf{y} = [\mathrm{y}_1, \ldots, \mathrm{y}_l]'$, $\mathbf{K}_m(\mathbf{x}_i, \mathbf{x}_j) = \varphi_m(\mathbf{x}_i)'\varphi_m(\mathbf{x}_j)$ and $\mathbf{F}_s(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{f}_s(\mathbf{x}_i)'\boldsymbol{f}_s(\mathbf{x}_j)$. It can be seen that (3) is in a standard form of the SVM dual problem with the kernel $\mathbf{K} = \sum_{m=1}^M d_m\mathbf{K}_m + \sum_{s=1}^S \mu_s\mathbf{F}_s$. Therefore, it can be solved via existing SVM solvers such as libsvm [44].

With the optimum $\boldsymbol{\alpha}$ obtained from problem (3), we can recover the primal variables $\mathbf{v}_m, \boldsymbol{\beta}_s$ accordingly:

$$\mathbf{v}_m = d_m\sum_{i=1}^{l}\alpha_i\mathrm{y}_i\varphi_m(\mathbf{x}_i),\ m=1,\ldots,M, \tag{4}$$

$$\boldsymbol{\beta}_s = \mu_s\sum_{i=1}^{l}\alpha_i\mathrm{y}_i\boldsymbol{f}_s(\mathbf{x}_i),\ s=1,\ldots,S. \tag{5}$$

Algorithm 1. Block-wise coordinate descent algorithm for GA-MKL.

```
Initialize $\mathbf{d}^1$ and $\boldsymbol{\mu}^1$; set $t = 1$.
repeat
    Obtain $\boldsymbol{\alpha}^t$ by solving (3) using the SVM solver with $\mathbf{d}^t$ and $\boldsymbol{\mu}^t$.
    Calculate $\|\mathbf{v}_m^t\|^2$ by using (4) and solve for $\mathbf{d}^{t+1}$ by using (7).
    Calculate $\|\boldsymbol{\beta}_s^t\|^2$ by using (5) and solve for $\boldsymbol{\mu}^{t+1}$ by using (8).
    $t = t + 1$.
until The convergence criterion is reached.
```

Optimize $\mathbf{d}, \boldsymbol{\mu}$ with Fixed $\mathbf{v}_m, b, \xi_i, \boldsymbol{\beta}_s$: With fixed $\mathbf{v}_m, b, \xi_i, \boldsymbol{\beta}_s$, the problem in (2) reduces to the following constrained convex minimization problem:

$$\begin{aligned} \min_{d_m, \mu_s} \ & \frac{1}{2}\sum_{s=1}^{S} \frac{\|\boldsymbol{\beta}_s\|^2}{\mu_s} + \frac{\lambda}{2}\sum_{s=1}^{S} \mu_s^2 + \frac{1}{2}\sum_{m=1}^{M} \frac{\|\mathbf{v}_m\|^2}{d_m} \\ \text{s.t.} \ & \mathbf{d} \geqslant 0, ||\mathbf{d}||_p^2 \leqslant 1, \boldsymbol{\mu} \geqslant 0. \end{aligned} \tag{6}$$

Similar to the derivations in [39], we obtain the closed-form solutions as follows:

$$d_m = \frac{||\mathbf{v}_m||^{\frac{2}{p+1}}}{(\sum_{r=1}^{M} ||\mathbf{v}_r||^{\frac{2p}{p+1}})^{1/p}}, \ m = 1, \ldots, M, \tag{7}$$

$$\mu_s = \sqrt[3]{\frac{||\boldsymbol{\beta}_s||^2}{2\lambda}}, \ s = 1, \ldots, S, \tag{8}$$

where $\|\mathbf{v}_m\|^2$ and $\|\boldsymbol{\beta}_s\|^2$ can be calculated by using (4) and (5), respectively.

The entire optimization procedure is summarized in Algorithm 1. After obtaining the optimal $\mathbf{d}$, $\boldsymbol{\mu}$ and $\boldsymbol{\alpha}$ using Algorithm 1, the final classifier for the test images can be expressed as

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i \mathrm{y}_i \left(\sum_{s=1}^{S} \mu_s \boldsymbol{f}_s(\mathbf{x})' \boldsymbol{f}_s(\mathbf{x}_i) \right) + \sum_{i=1}^{l} \alpha_i \mathrm{y}_i \left(\sum_{m=1}^{M} d_m \mathbf{K}_m(\mathbf{x}, \mathbf{x}_i) \right) + b.$$

5 Experiments

In this section, we evaluate the proposed two-level feature extraction framework and GA-MKL on two broadly recognized image databases for object and scene classification: Caltech256 [16] and 15Scenes [1, 17, 18].

5.1 Experimental Setup

Local Descriptor Extraction: Three types of local descriptors – dense SIFT [40], spatial HOG [42] and LBP [43] – are used in our experiments. SIFT is extracted from densely located patches centered at every 4 pixels in the image, with a patch size of 16×16 pixels. For spatial HOG, the HOG descriptors are extracted from densely located patches centered at every 8 pixels in the image, with a patch size of 8×8 pixels. Then the spatial HOG descriptor is formed by stacking together 2×2 neighboring local HOG descriptors. For LBP, the uniform LBP as described in [43] is adopted.

Dictionary Learning: K-means clustering is employed for both levels of feature extraction. The dictionary size for all second level feature extractions is set to 4,096. The dictionary size for the first level SIFT feature extraction is set to 4,096 as well. All other dictionary sizes are set to 1,024.

Encoding: Localized soft assignment [9] is used for both levels of encoding.

Pooling: The first level feature extraction of LBP is pooled by average pooling. In all other cases, the codes are pooled via max pooling. A three level spatial pyramid of 1×1, 2×2 and 4×4 is used.

Feature Normalization and Designation: The first level image features of the LBP local descriptor are normalized with the ℓ_1-norm equal to 1. The other types of image features are each normalized with the ℓ_2-norm equal to 1.

The first level image feature is referred to as a Spatial Pyramid Representation (SPR) feature. The first level feature together with the second level feature is referred to as the Beyond Spatial Pyramid Representation (BSPR) feature.

Kernel Learning: ℓ_p-MKL and GA-MKL are implemented using the libsvm software package [44]. Linear kernels with C set to 10 are used throughout the experiments. In ℓ_p-MKL and GA-MKL, we fix p to 2. In GA-MKL, we empirically set λ to 10 for both datasets. For the pre-learned classifiers in GA-MKL, there are six sets in total, with each set learned by using each type of BSPR feature. From the six sets of pre-learned classifiers and the six linear kernels generated by the six kinds of BSPR features, the GA-MKL classifier is learned.

All experiments on each dataset are repeated five times with different randomly selected training images and the same experimental settings. The results are reported in terms of the mean and standard deviation from all five runs.

5.2 Results on the Caltech256 Dataset

Caltech256 [16] provides challenging data for object recognition. It consists of 30,608 images with 256 object categories and 80 to 827 images per category. In our series of experiments on Caltech256, we take 30, 45 and 60 images from each category for training and use the rest as test samples.

Performance comparisons with the baseline method are listed in the upper part of Table 1. From it, one can see that the classification accuracy with BSPR features consistently yields better results than the one with SPR features in all three of the training scenarios. With ℓ_p-norm MKL, the improvements of the BSPR feature over the SPR feature are 2.03%, 2.38% and 2.73% respectively. This demonstrates that the proposed second level features provide additional information which is complementary to the SPR with the first level features. Also,

Table 1. Classification accuracy (%) on the Caltech256 dataset. SPR feature (ℓ_p-MKL) is the baseline method implemented in this paper. BSPR feature (ℓ_p-MKL) and BSPR feature (GA-MKL) correspond to our proposed BSPR feature learned with ℓ_p-MKL and our proposed GA-MKL. Note: - indicates unavailability of results.

Method	30 training	45 training	60 training
SPR feature (ℓ_p-MKL)	43.75 ± 0.20	47.23 ± 0.23	48.92 ± 0.31
BSPR feature (ℓ_p-MKL)	45.78 ± 0.18	49.61 ± 0.16	51.65 ± 0.35
BSPR feature (GA-MKL)	**46.82 ± 0.23**	**50.69 ± 0.15**	**52.91 ± 0.59**
Sparse coding [6]	34.02 ± 0.35	37.46 ± 0.55	40.14 ± 0.91
Improved Fisher Kernel [24]	40.80 ± 0.10	45.00 ± 0.20	47.90 ± 0.40
Efficient Match Kernel [37]	30.50 ± 0.40	34.40 ± 0.40	37.60 ± 0.50
Affine sparse codes [26]	45.83	49.30	51.36
Locality-constrained linear coding [7]	41.19	45.31	47.68
Geometric ℓ_p-norm Feature Pooling [10]	43.17	47.32	-
Nearest-neighbor [35]	42.70	-	-
Random Forest [27]	44.00	-	-
Graph-matching kernel [34]	38.10 ± 0.60	-	-
Multi-way local pooling [12]	41.70 ± 0.80	-	-

it is shown in the table that the results using the BSPR feature and our proposed GA-MKL are better than those using BSPR and ℓ_p-MKL by 1.04%, 1.08% and 1.26%, which indicates that it is beneficial to learn an adapted classifier that leverages on pre-learned classifiers from other classes. This is consisted with the previous work [15, 38, 45]. In total, the proposed BSPR feature and GA-MKL improves upon the baseline method by 3.07%, 3.46% and 3.99% respectively.

After learning the adapted classifiers, we observe that similar concepts have higher weights than dissimilar ones. Taking for instance the concepts of "Swan" and "Gorilla", the two largest β values are as follows: Swan($\beta_{duck} = 0.092$, $\beta_{goose} = 0.078$), Gorilla($\beta_{chimp} = 0.195$, $\beta_{raccoon} = 0.106$). These learned values also reflect the benefit of leveraging pre-learned classifiers of other classes.

Comparisons with State-of-the-Art: In the lower part of Table 1, comparisons with state-of-the-art methods are provided. The listed methods include the most recently reported techniques as well as the highest achieving methods from the past. Our method is seen to outperform all the existing methods with various numbers of training samples. To be exact, Our method exceeds the existing best results [26] (underlined in Table 1) by 0.99%, 1.39% and 1.55% for 30, 45 and 60 training samples, respectively.

5.3 Results on the 15Scenes Dataset

The 15Scenes dataset is composed of 15 classes of scenes and contains 4,485 images in total, reported in [1, 17, 18]. Following the common evaluation protocol on this dataset, we randomly select 100 images from each class as training samples and use the rest as test samples. Table 2 presents performance comparisons.

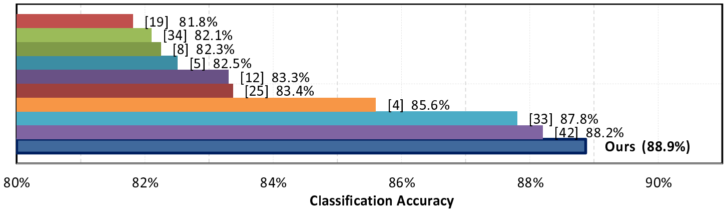

Fig. 3. Comparison with state-of-the-art results on 15Scenes

Table 2. Classification accuracy (%) on 15Scenes with 100 training images

Method	Classification Accuracy
SPR feature (ℓ_p-MKL)	86.60 ± 0.66
BSPR feature (ℓ_p-MKL)	88.32 ± 0.72
BSPR feature (GA-MKL)	**88.87 ± 0.56**

Using ℓ_p-MKL, classification accuracy with the BSPR features exceeds that of the baseline method with SPR features, which again demonstrates the effectiveness of our proposed two level feature extraction framework. The result using the BSPR feature and GA-MKL is also better than that from the BSPR feature and ℓ_p-MKL, which validates the effectiveness of GA-MKL in leveraging pre-learned classifiers from other classes. In total, our proposed BSPR feature with our GA-MKL brings an overall improvement in classification accuracy of 2.27% over the baseline.

Performance of Individual Features: For individual BSPR features, the results are 83.2%, 84.6% and 70.4% (resp. 75.8%, 69.8%, 69.5%) using SIFT, SHOG and LBP features at the first (resp. second) level. Note that the result after combining all three first level features (86.6%) is better than the results from each individual feature at the first level, which shows the effectiveness of ℓ_p-MKL. Though the individual results at the second level are not as good as those corresponding to the first level, they are complementary to the first level features, and the combination of two levels of features using ℓ_p-MKL leads to a better result (i.e., 88.32% vs. 86.6% in Table 2).

Comparisons with State-of-the-Art: In Fig. 3, comparisons with state-of-the-art methods are provided. The listed methods include the latest techniques and top performers. Our method still achieves the best results on this dataset.

5.4 Computation Time

The proposed two-level feature extraction framework involves a second round of encoding and pooling that adds to the computation time. Processing speed additionally depends on the codebook sizes in the first level and second level feature extraction, the number of local descriptors in the first level, and the number of windows in the second level. For the methods and settings used in this work, with the SIFT descriptor as an example, the CPU times for the first level (5,184 SIFT descriptors with the feature dimension of 128) and second level (3,025 windows with the window-based feature dimension of 4,096) feature extraction are about 10s and 15s on a 300×300 image for Caltech256, with an IBM workstation (3.33GHz CPU with 18GB RAM) and Matlab implementation.

6 Conclusion

We presented two technical contributions for image classification. The first is a novel feature extraction framework that generalizes window-based features to the image level in a manner that efficiently accounts for densely sampled windows and allows for existing encoding and pooling techniques to be used. Secondly, we proposed Generalized Adaptive ℓ_p-norm Multiple Kernel Learning (GA-MKL) to incorporate the two different levels of features and to leverage multiple sets of pre-learned classifiers from other classes. Our extensive experimental results on benchmark datasets show that our work outperforms the state-of-the-art.

Acknowledgement. This research is supported by the Singapore National Research Foundation under its Interactive & Digital Media (IDM) Public Sector R&D Funding Initiative and administered by the IDM Programme Office. This research is also supported by the National Natural Science Foundation of China (Grant No: 61125106).

References

1. Lazebnik, S., Schmid, C., Poncer, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR (2006)
2. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: ICCV (2003)
3. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV (2004)
4. Boureau, Y.L., Bach, F., LeCun, Y., Ponce, J.: Learning mid-level features for recognition. In: CVPR (2010)
5. Yang, Y., Newsam, S.: Spatial pyramid co-occurrence for image classification. In: ICCV (2011)
6. Yang, J., Yu, K., Gong, Y., Huang, T.S.: Linear spatial pyramid matching using sparse coding for image classification. In: CVPR (2009)
7. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T.S., Gong, Y.: Locality-constrained linear coding for image classification. In: CVPR (2010)
8. Huang, Y., Huang, K., Yu, Y., Tan, T.: Salient coding for image classification. In: CVPR (2011)
9. Liu, L., Wang, L., Liu, X.: In defense of soft-assignment coding. In: ICCV (2011)
10. Feng, J., Ni, B., Tian, Q., Yan, S.: Geometric ℓ_p-norm feature pooling for image classification. In: CVPR (2011)

11. Yan, S., Shan, S., Chen, X., Gao, W., Chen, J.: Locally assembled binary (lab) feature with feature-centric cascade for fast and accurate face detection. In: CVPR (2008)
12. Boureau, Y.L., Roux, N.L., Bach, F.: Ask the locals: Multi-way local pooling for image recognition. In: ICCV (2011)
13. Varma, M., Ray, D.: Learning the discriminative power-invariance trade-off. In: ICCV (2007)
14. Yang, J., Li, Y., Tian, Y., Duan, L., Gao, W.: Group-sensitive multiple kernel learning for object categorization. In: ICCV (2009)
15. Wu, X., Xu, D., Duan, L., Luo, J.: Action recognition using context and appearance distribution features. In: CVPR (2011)
16. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical report, California Institute of Technology (2007)
17. Oliva, A., Torraba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelop. IJCV (2001)
18. Li, F.F., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: CVPR (2004)
19. Harada, T., Ushiku, Y., Yamashita, Y., Kuniyoshi, Y.: Discriminative spatial pyramid. In: CVPR (2011)
20. Wu, J., Rehg, J.M.: Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel. In: ICCV (2009)
21. Marszałek, M., Schmid, C., Harzallah, H., Van De Weijer, J.: Learning object representations for visual object class recognition. In: ICCV, Visual Recognition Challenge Workshop (2007)
22. Zhou, X., Yu, K., Zhang, T., Huang, T.S.: Image Classification Using Super-Vector Coding of Local Image Descriptors. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 141–154. Springer, Heidelberg (2010)
23. Yang, J., Yu, K., Huang, T.: Efficient Highly Over-Complete Sparse Coding Using a Mixture Model. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 113–126. Springer, Heidelberg (2010)
24. Perronnin, F., Sánchez, J., Mensink, T.: Improving the Fisher Kernel for Large-Scale Image Classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 143–156. Springer, Heidelberg (2010)
25. Wang, X., Bai, X., Liu, W., Latecki, L.J.: Feature context for image classification and object detection. In: CVPR (2011)
26. Kulkarni, N., Li, B.: Discriminative affine sparse codes for image classification. In: CVPR (2011)
27. Bosch, A., Zisserman, A., Munoz, X.: Image classification using random forests and ferns. In: ICCV (2007)
28. Agarwal, A., Triggs, B.: Multilevel image coding with hyperfeatures. IJCV (2008)
29. Kovashka, A., Grauman, K.: Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In: CVPR (2010)
30. Jia, Y., Huang, C., Darrell, T.: Beyond spatial pyramids: Receptive field learning for pooling image features. In: CVPR (2012)
31. Li, L.J., Su, H., Xing, E.P., Fei-Fei, L.: Object bank: A high-level image representation for scene classification & semantic feature sparsification. In: NIPS (2010)
32. Torresani, L., Szummer, M., Fitzgibbon, A.: Efficient Object Category Recognition Using Classemes. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 776–789. Springer, Heidelberg (2010)

33. Su, Y., Jurie, F.: Visualword disambiguation by semantic contexts. In: ICCV (2011)
34. Duchenne, O., Joulin, A., Ponce, J.: A graph-matching kernel for object categorization. In: ICCV (2011)
35. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: CVPR (2008)
36. Gehler, P., Nowozin, S.: On feature combination for multiclass object classification. In: ICCV (2009)
37. Bo, L., Sminchisescu, C.: Efficient match kernels between sets of features for visual recognition. In: NIPS (2009)
38. Duan, L., Xu, D., Tsang, I.W.H., Luo, J.: Visual event recognition in videos by learning from web data. TPAMI (2012)
39. Kloft, M., Brefeld, U., Sonnenburg, S., Zien, A.: ℓ_p-norm multiple kernel learning. JMLR (2011)
40. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV (2004)
41. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: CVPR (2010)
42. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
43. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. TPAMI (2002)
44. Chang, C.C., Lin, C.J.: Libsvm: A library for support vector machines. ACM TIST (2011)
45. Chen, L., Xu, D., Tsang, I.W.H., Luo, J.: Tag-based image retrieval improved by augmented features and group-based refinement. IEEE Trans. on Multimedia (2012)