Image and Vision Computing 55 (2016) 64–76

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Robust geometric ℓ_p-norm feature pooling for image classification and action recognition[☆]

Teng Li[a], Zhijun Meng[b,*], Bingbing Ni[c], Jianbing Shen[d], Meng Wang[e]

[a] Anhui University, Hefei, PR China
[b] Beihang University, Beijing, PR China
[c] Shanghai Jiaotong University, Shanghai, PR China
[d] Beijing Institute of Technology, Beijing, PR China
[e] Hefei University of Technology, Hefei, PR China

ARTICLE INFO

Article history:
Received 16 September 2015
Received in revised form 25 March 2016
Accepted 7 April 2016
Available online 19 April 2016

Keywords:
Feature pooling
Spatial distribution
Image classification
Action recognition

ABSTRACT

Feature pooling is a key component in modern visual classification system. However, the conventional two prevailing pooling techniques, namely average and max poolings, are not theoretically optimal, due to the unrecoverable loss of the spatial information during the statistical summarization and the underlying over-simplified assumption about the feature distribution. Addressing these issues, this paper proposes to generalize previous pooling methods toward a weighted ℓ_p-norm spatial pooling function tailored for class-specific feature spatial distribution. Optimizing such a pooling function toward discriminative class separability that is subject to a spatial smoothness constraint yields a so-called geometric ℓ_p-norm pooling (GLP) method. Furthermore, to handle the variation of object scale/position, which would affect not only the learning of discriminative pooling weights but also the applicability of the learned weights, we propose a simple yet effective self-alignment step during both learning and testing to adaptively adjust the pooling weights for individual images. Image segmentation and visual saliency map are utilized to construct a directed pixel adjacency graph. The discriminative pooling weights are diffused using random walk on the constructed graph and therefore the discriminative pooling weights are propagated onto the salient and foreground region. This leads to a robust version of GLP (RGLP) which can cope with the misalignment of object position and scale in images. Comprehensive experiments validate the effectiveness of the proposed GLP feature pooling framework. The proposed random walk based self-alignment step can effectively alleviate the image misalignment issue and further boost classification accuracy. State-of-the-art image classification and action recognition performances are attained on several benchmarks.

1. Introduction

Driven by the increasing amount of image and video data from internet or surveillance cameras, computer vision areas such as image classification [59,63], image re-ranking [58,60], and action recognition [45,61] have made significant progresses in recent years. As an important step in many practical visual recognition tasks, feature selection is of great interests to many researchers [32,33,34,44]. With the prevalence of the bag-of-words (BoW) model [31] for image classification or image-based action recognition [46], feature pooling has become a common practice for image/video feature representation and selection. For a typical image classification task, local image features are first extracted and quantized according to a visual dictionary. Then, the quantization indices of all the local features are summarized to form the global feature representation. A most common summarization method is to form the histogram, *i.e.* to sum up all the occurrences of each index throughout the entire image in an orderless manner. From the viewpoint of feature pooling [12,28], histogram representation is equivalent to average pooling. Despite its conceivable ease and compactness, average pooling is not immune to local feature noise.

To overcome this limitation, max pooling has been proposed [39,40]. Instead of performing averaging operation, max pooling adopts the element-wise maximum values of feature vectors over the whole image or the region of interest as the pooled features. Max

[☆] This paper has been recommended for acceptance by Ling Shao.
[*] Corresponding author.
E-mail address: mengzhijun@buaa.edu.cn (Z. Meng).

http://dx.doi.org/10.1016/j.imavis.2016.04.002

pooling has proved to be more robust against local feature noise and can achieve better classification performance [55].

The simple assumption associated with average or max pooling, that the spatial distribution for each visual feature is uniform across different classes, causes severe information loss. However, spatial distribution of available features can be important for visual recognition. In the image classification task, if we assume the objects/regions in the images are roughly aligned, the image local features do possess class-specific discriminative geometric information, *i.e.* spatial distribution patterns. Fig. 1 illustrates such an issue for the average and max pooling methods. For images from a specific class, their visual features indexed by the same visual word often share similar spatial distribution. Besides, such class-specific spatial distributions are quite distinct from each other and encode discriminative information. However, as shown in this figure, neither average nor max pooling can capture the underlying difference and produce discriminative features due to the loss of spatial information in the pooling process.

Moreover, these two deterministic pooling methods either treat all the local features uniformly or only select the most salient one, and they both assume local features are distributed independently. By comparison, a discriminative pooling scheme is expected to be more flexible and able to capture the spatial correlation of features.

Motivated by the above considerations, we propose a so-called geometric ℓ_p-norm pooling method. Overall, the proposed method aims to learn a pooling function that implicitly encodes the class-specific geometric information of feature distribution in the form of weighted norm. This function is optimized toward best class separability, and in the meantime, it takes into account the following prior knowledge: nearby local image pixels often present similar characteristics, thus a regularization term is employed that encodes the correlation of local features.

Another inevitable problem for image classification or action recognition is the misalignment of image foreground, which is caused by large variation in object position/scale in each image. Misalignment of the foreground regions/objects in the training image degrades the effectiveness of the learned discriminative feature pooling function. Moreover, if the object position and scale of a testing image is not aligned with those of the training images, the learned common pooling function cannot capture the discriminative features for classification.

In this work, we propose a simple yet effective self-alignment method using the side information from visual saliency [21] and image segmentation [2], which can not only adaptively adjust the discriminative pooling weights for individual images during the training process, but also tailor the learned pooling function for individual testing images. A basic observation is that within a visually consistent (*e.g.*, homogeneous color) image local region, pixels convey similar discriminative information, thus the pooling weights for the pixels within the same local region should be similar. Motivated by this observation, we construct an adjacency graph where nodes represent pixels and edges encode the spatial and color adjacency between pixels. Simple random walk algorithm can effectively and efficiently diffuse and adapt the learned common pooling function onto individual images based on the constructed adjacency graph. Further, visual saliency map is utilized to convert the adjacency graph into a directed graph and it can direct the pooling weights propagation toward the object (foreground) region of the given image. This random walk based self-alignment step results in an image-specific adaptive feature pooling scheme which is robust to image foreground misalignment.

Based on the GLP framework originally developed for image classification [10], we further consider the misalignment problem and propose the RGLP algorithm, which can be then applied to several applications including image classification and action recognition. To this aim, the contents of introduction, the experiments, and other related parts are extended correspondingly. Our experimental results show that the proposed robust geometric ℓ_p-norm pooling scheme is insensitive to median level image foreground misalignment. To sum up, the proposed robust geometric ℓ_p-norm pooling framework possesses the following advantages:

- As the pooling function is learned by directly maximizing the class separability, it is designed to bear good discriminating capability.
- The pooling function exactly corresponds to the class specific spatial pattern of each visual word, thus the spatial distribution information of visual words is properly utilized.
- It models the correlations among local features and makes a more reasonable assumption about feature distribution. Also it can naturally unify the average and max pooling in a more flexible framework.

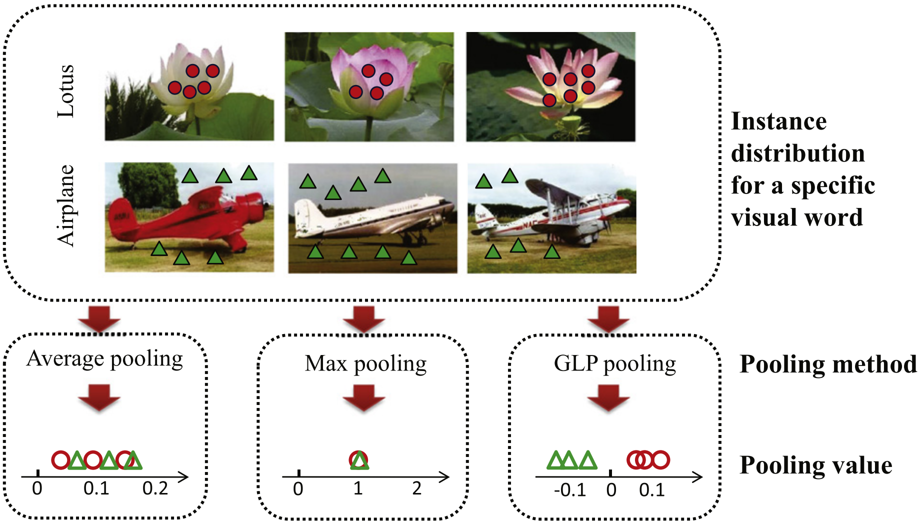

Fig. 1. Illustration of the importance of the visual word spatial distribution for visual classification purposes. In the top block, the distributions of a specific visual word in two classes are indicated by circles and triangles respectively. In the bottom blocks, circles and triangles represent the pooled statistic values of the two classes. By utilizing the class-specific local feature spatial distributions, geometric ℓ_p-norm pooling (GLP) can generate more separable pooled values, compared with the average and max pooling.

- Using the simple random walk based self-alignment module, the learning pooling weights can be tailored to individual images according to the object (foreground) position and scale, as well as the image segmentation results. Therefore the object (foreground) misalignment problem is alleviated and the resulting image representation is more robust and the classification performance is further boosted.

The remainder of this paper is organized as follows. The related literature is discussed in Section 2. Section 3 then elaborates on the geometric ℓ_p-norm feature pooling method and provides the theoretical comparison with the max and average pooling methods. An iterative optimization procedure for learning the discriminative pooling weights is also presented. In Section 5 we introduce the random walk based self-alignment method to alleviate the image misalignment problem, which results in an image-specific adaptive pooling scheme. In Section 6 extensive experimental results on benchmarks are presented and conclusions are drawn in Section 7.

2. Related work

The idea of feature pooling originates in the research on complex cells in the striate cortex [20]. In [20], they proposed a model in which responses of simple cells are fed into higher complex cells through some pooling operations, thereby endowing the complex cells with phase-invariance. Inspired by this seminal work, several extensions in the direction of pooling mechanisms have been proposed afterwards and widely applied in recent computer recognition systems. In the neocognitron model [12], a sigmoid-like function is used to pool the input signals into a single output. And convolutional networks [28] take the average value of the input signals for subsequent processing. Besides, another type of pooling via max operation is used in the HMAX class of models [42]. Wang et al. [50] have achieved impressive classification performance on several benchmarks through such max pooling. As pointed out by Jarrett et al. [23], pooling type matters more for classification tasks than careful unsupervised pre-training of features. However, most of the studies on the pooling methods are purely empirical. Boureau et al. [6] provided a theoretical analysis on the binary feature pooling in the context of classification. Based on the i.i.d. Bernoulli distribution assumption, they demonstrated that several factors, including the pooling cardinality and the sparsity of the features, affect the discriminative powers of the pooling results. Neither the average nor max pooling can always outperform the other in classification problems, which raises a question: in which way can we optimally pool the features? The proposed work is dedicated to address this problem.

As the global feature pooling method cannot encode spatial information, Lazebnik et al. [27] extended the bag-of-words (BoW) model with spatial pyramid matching (SPM) kernel by exploiting the spatial information of location regions. However, object positions and scales in the image usually have large variations and therefore the fixed spatial pyramid feature pooling approach is not always optimal. In image/object classification, several adaptive feature pooling schemes have been studied to address this issue. Bosch et al. [5] define a region-of-interest in the image and take the maximum response over the coarse image grid as the output of the classifier. Chai et al. [7] proposed to segment the images into foreground and background within the co-segmentation scenario to improve image classification performance. Yakhnenko and Verbeek [52] used a latent-SVM model, which scores an image using all regions and associates each region with a latent variable indicating whether the region represents the object of interest. Recently, Sadeghi and Tappen [41] proposed a latent pyramidal region (LPR) representation for scene image classification and all possible sub-windows of the images are input into a latent SVM training procedure to make the classification robust to different spatial configurations of images. In contrast, our proposed method directly learns a discriminative pooling function toward maximal class separability. The learned pooling function is a function that takes pixel position as input, and therefore it can encode richer discriminative information than average/max pooling.

Visual saliency map [21] has been used as guidance for object recognition as it roughly reflects region-of-interest of the object in the image. Moosmann et al. [36] presented an approach for object category recognition using the visual attention technique. It combines saliency maps very closely with the extraction of random sub-windows for classification purpose. Khan et al. [43] used color to guide attention by means of a top-down category specific attention map. The color attention map is deployed to modulate more shape features from regions within an image that are likely to contain an object instance. Wang and Forsyth [49] jointly learn object attribute, label and visual attention by exploring multiple instance learning respectively to classify images by the highest scored image region. Kanan and Cottrell [25] attempted to solve image classification using a biologically-inspired model to approximate the human eye fixations. Chen et al. [8] derived a hierarchical matching kernel that utilizes side information for hierarchically partitioning the image into irregular feature pooling regions. Sharma et al. [47] treated the saliency maps as latent variables and allowed them to adapt to the image content to maximize the classification score. Inspired by a similar motivation, in this work we adopt the saliency map as prior knowledge on the object region of interest for adaptively adjusting the learned pooling function.

Meanwhile a body of works have been devoted to discriminative feature encoding [3,13,50,51,55] and dictionary learning [26,56], which consider the visual feature encoding setup and visual dictionary formation step in the BoW representation pipeline. Instead, our work focuses on the subsequent visual words spatial aggregation step and can be seamlessly combined with any dictionary formation method.

3. Geometric ℓ_p-norm feature pooling

The pipeline of a popular image classification procedure is shown in Fig. 2. As can be seen from the figure, a multi-stage image classification architecture generally comprises four components. After local features are extracted from the input image, many methods can be used to encode the feature vectors.

The first two building blocks are feature extraction and encoding. We assume that there are n_c image classes, and the class index set is denoted as $\mathcal{C} = \{1, 2, \cdots, j, \cdots, n_c\}$. Additionally, we denote the image index set for the j-th class as $\mathcal{I}_j$ and the number of images in the j-th class is denoted as N_j. Denote the location index set as

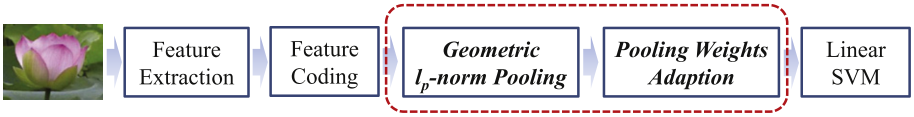

Fig. 2. Overview of the image classification flowchart. The shown architecture has proven to perform best among the methods based on a single type of features [50]. Here we replace the original max pooling building block with our proposed geometric ℓ_p-norm pooling method and image-specific pooling weights adaption module, and shall show the new pipeline is better.

$\mathcal{M} = \{1, 2, \ldots, m, \cdots, M\}$ in an image with M feature locations, *e.g.* distributed over a regular grid. For each image I, we extract a set of d-dimensional local descriptors, *e.g.* SIFT [35], from M densely arranged locations. Then each local descriptor $\mathbf{x}$ is encoded by a pre-trained visual word dictionary $D \in \mathbb{R}^{d\times K}$ (*i.e.*, $D = \{\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_K\}$, where $\mathbf{d}_k$ is a d-dimensional vector denoted as visual word prototype) into a K-dimensional code vector $\mathbf{u}$ in a pre-defined feasible region $\mathcal{F}$:

$$\begin{aligned} &\mathbf{u} = \arg\min_{\mathbf{u}} ||\mathbf{x} - D\mathbf{u}||_2, \\ &s.t. \quad \mathbf{u} \in \mathcal{F}. \end{aligned} \tag{1}$$

When $\mathcal{F}$ is constrained to the set of 0–1 vectors with only a single entry equal to 1, the encoding method is known as the hard assignment. When $\mathcal{F}$ is defined as the set of neighboring bases of the local descriptor $\mathbf{x}$, the resultant $\mathbf{u}$ could for example correspond to the recently proposed locality-constrained linear coding (LLC) [50] as follows:

$$\begin{aligned} &\mathbf{u} = \arg\min_{\mathbf{u}} ||\mathbf{x} - D\mathbf{u}||_2 \\ &s.t. \quad \mathbf{u} \in \mathcal{F}, \quad \mathcal{F} = \{\mathbf{u} | ||\mathbf{u} \circ \mathbf{e}||_2 \le \lambda, ||\mathbf{u}||_1 = 1\}, \end{aligned} \tag{2}$$

where the entries of $\mathbf{e}$ are the Euclidean distances between $\mathbf{x}$ and the bases in D. Each element u_k of the code vector $\mathbf{u}$ indicates the local descriptor's response to the k-th visual word in the dictionary D. We aggregate the local descriptors' responses across all the M image locations into an M-dimensional response vector $\mathbf{v}^{(k)}$. Namely, each element $v_m^{(k)}$ of $\mathbf{v}^{(k)}$ represents the response of the local descriptor $\mathbf{x}_m$ at the m-th location to the k-th visual word.

After feature extraction and encoding, pooling operations are performed to aggregate the encoded response vectors into a statistic vector to represent the whole image or the region of interest. Finally the pooled feature vector is fed into a classifier, and then further assigned to one of the pre-defined classes. As illustrated in Fig. 2, this work is aimed at replacing the pooling component only rather than renewing the whole classification architecture. In fact, the proposed GLP and RGLP pooling scheme can be seamlessly combined with arbitrary types of local features, encoding methods and ultimate classifiers. In the following, we will first review the limitations of the conventional feature pooling schemes and then present the proposed GLP and RGLP pooling scheme.

3.1. Pooling methods revisited

Essentially, feature pooling is to map the response vector $\mathbf{v}^{(k)}$ into a statistic value $f(\mathbf{v}^{(k)})$ via some spatial pooling operation f, where $f(\mathbf{v}^{(k)})$ is used to summarize the joint distribution of visual features over the region of interest. Here, for notational simplicity, we drop the visual word index k for $\mathbf{v}^{(k)}$ in all the following sections.

In modern visual classification models, there are two widely used pooling operations, *i.e.* the average pooling [6] and the max pooling [39]. Average pooling adopts the scaled ℓ_1-norm of the response vector $\mathbf{v}$ as the statistic value and its operation can be expressed as

$$f_a(\mathbf{v}) = \frac{1}{M}||\mathbf{v}||_1 = \frac{1}{M}\sum_{m=1}^{M} v_m. \tag{3}$$

Namely, average pooling sums up the response values throughout the entire image or the region of interest in an orderless manner. The pooling result is generally tolerant to object transformation. However, it is not selective or discriminative enough for the classification tasks [37].

Recently, inspired by the mechanism of the complex cells in the primary visual cortex, another pooling operation is proposed in [42]. The so-called max pooling operation computes the ℓ_∞-norm of the response vector,

$$f_m(\mathbf{v}) = ||\mathbf{v}||_\infty = \max_m v_m. \tag{4}$$

The max pooling only captures the most salient response over the whole image or the region of interest. Thus it is more selective than the average pooling and able to preserve invariance to object's spatial transformations [42].

However, both pooling methods discard the spatial distributions of local descriptors by either forcing the distribution to be uniform (the average pooling) or only adopting the most salient location (the max pooling). This information loss severely limits their discriminating capability and degrades the performance of the subsequent classification procedures.

In fact, each visual word may exhibit a certain geometric structure within individual classes since images for a certain classification task are often well roughly aligned or can be roughly aligned automatically by saliency or symmetry detection. These structures can contribute significantly to the discriminating capability once they are properly utilized as illustrated in Fig. 1. But once lost, this useful information could never be recovered in the subsequent process. Therefore if we can well model the spatial distribution for individual visual words, the obtained pooling results will be more discriminative than those from traditional pooling methods. In the case that the foreground regions of images are misaligned, we also provide a self-alignment module that adapts the common pooling function to individual images during both training and testing.

3.2. Geometric ℓ_p-norm pooling

As discussed, both the average and max pooling discard the geometric information of local responses and thus only maintain limited discriminating capability. To overcome this inherent issue, we propose the so-called geometric ℓ_p-norm pooling (GLP) method. GLP is aimed at utilizing the spatial distribution patterns of responses across different classes and meanwhile preserving the selective capability and robustness as traditional pooling methods do.

More specifically, GLP process is defined as

$$\begin{aligned} &f_g(\mathbf{v}; \mathbf{w}) = \sum_{m=1}^{M} w_m v_m^p = \mathbf{w}^T \mathbf{v}^p, \\ &s.t. \quad ||\mathbf{w}||_2 = 1, \quad p \ge 0, \end{aligned} \tag{5}$$

where $\mathbf{v}^p$ denotes the element-wise p-th power of the response vector $\mathbf{v}$. The geometric coefficient w_m encodes the contribution of the m-th image location for the specific visual word. Different locations are given different weights during the pooling process. A special case is when all w_ms are of the same value, *i.e.*, average pooling. The parameter p determines the selection policy for locations. Note that $\mathbf{v}$ has been normalized by its ℓ_2-norm, and all the elements of $\mathbf{v}$ are smaller than or equal to 1. Therefore, when the value of p equals 1, GLP aggregates the responses over the entire region uniformly without preference to any location (same to the average pooling). When p increases to a large value, the policy changes toward winner-take-all (same to the max pooling). Namely, the value of p tunes the pooling operation to transit from the average to the max pooling. Instead of fixing the value of p, GLP adopts a more flexible one and possesses better selective capability. Moreover, in GLP method, the values of $\mathbf{w}$ and p are visual-word-specific. This enables GLP to better capture geometric information of the descriptors based on the fact that different visual features usually follow different spatial distributions among different classes.

4. Toward discriminative geometric ℓ_p-norm feature pooling

4.1. Class separability

To determine the parameters in the GLP, we adopt the class separability as the objective function and optimize it with respect to both $\mathbf{w}$ and p. A practical choice of the class separability criterion is the marginal Fisher analysis (MFA) developed in [53]. MFA can well characterize the class separability of the data with more general distributions beyond the Gaussian distribution. More specifically, the objective function is to maximize the inter-class separability scaled by the within-class compactness of the pooled features, namely,

$$\max_{\mathbf{w},p}\{\mathcal{D}(\mathbf{w},p) := \frac{\mathbf{w}^T S_b(p)\mathbf{w}}{\mathbf{w}^T S_w(p)\mathbf{w}}\}, \tag{6}$$

where $S_b(p)$ characterizes the separability of different classes and $S_w(p)$ describes the within-class compactness [53]. These two matrices are computed as follows:

$$\begin{aligned} S_b(p) &= \sum_i \sum_{j\in N^-_{k_1}(i)} (\mathbf{v}_i^p - \mathbf{v}_j^p)(\mathbf{v}_i^p - \mathbf{v}_j^p)^T, \\ S_w(p) &= \sum_i \sum_{j\in N^+_{k_2}(i)} (\mathbf{v}_i^p - \mathbf{v}_j^p)(\mathbf{v}_i^p - \mathbf{v}_j^p)^T. \end{aligned} \tag{7}$$

Here $N^-_{k_1}(i)$ means the index set for the k_1 nearest neighbors of the response vector $\mathbf{v}_i$ from different classes with $\mathbf{v}_i$ and $N^+_{k_2}(i)$ denotes the k_2 nearest neighbors of $\mathbf{v}_i$ from the same class as $\mathbf{v}_i$.

4.2. Spatial correlation of local features

The previous analysis in [6] is based on the strict assumption that the response values from M locations in $\mathbf{v}$ are independent. However, this assumption is often invalid for real-world data as also mentioned in [6], since image features at adjacent locations are often strongly correlated. Ignoring this important fact may lead to degraded capability in describing images. Although there exists little prior knowledge about the exact form of the spatial correlation, an intuitive and simple constraint/prior is that the geometric coefficients of Eq. (5) located at adjacent image locations should exhibit similar values. To incorporate this spatial smoothness constraint, we define a spatially smooth function as follows:

$$\mathcal{S}(\mathbf{w}) = \sum_{i,j\in\mathcal{M}, i\neq j} s^a_{ij} \parallel w_i - w_j\|^2. \tag{8}$$

The value of weight s^a_{ij} is set as

$$s^a_{ij} = \exp(-\frac{\parallel \mathbf{a}_i - \mathbf{a}_j\|^2}{2\rho_a^2}), \tag{9}$$

where $\mathbf{a}_i$ denotes the spatial coordinates of the i-th feature location. Minimizing $\mathcal{S}(\mathbf{w})$ penalizes the case when adjacent elements of $\mathbf{w}$ show a large numerical gap. ρ_a is an empirical bandwidth parameter of the neighborhood and fixed as 0.5 in all our experiments.

The smooth function can be further rewritten as

$$\mathcal{S}(\mathbf{w}) = \mathbf{w}^T L\mathbf{w}, \tag{10}$$

where the Laplacian matrix L is defined as $L = D - S$. The similarity matrix S is defined as $S = [s^a_{ij}]_{M\times M}$ and the degree matrix D is a diagonal matrix with $D_{ii} = \sum_j S_{ij}$.

4.3. Discriminative geometric pooling

We combine the objectives in both Eqs. (6) and (10) into a unified function with a weighting factor λ, namely,

$$\begin{aligned} \max_{\mathbf{w},p} \mathcal{Q}(\mathbf{w},p) \ &:= \ \frac{\mathbf{w}^T S_b(p)\mathbf{w}}{\mathbf{w}^T S_w(p)\mathbf{w} + \lambda\mathbf{w}^T L\mathbf{w}} \\ &= \ \frac{\mathbf{w}^T S_b(p)\mathbf{w}}{\mathbf{w}^T \tilde{S}_w(p)\mathbf{w}}, \end{aligned} \tag{11}$$

where $\tilde{S}_w$ is the regularized within-class scatter matrix, *i.e.*, $\tilde{S}_w = S_w + \lambda L$.

Though the optimization problem in Eq. (11) is not convex overall, there exists a closed form solution for $\mathbf{w}$ when p is fixed. Thus, we solve this optimization problem iteratively by optimizing with respect to p and $\mathbf{w}$ alternatively.

Note that when optimizing for $\mathbf{w}$, this objective function has the same form as the well-known linear discriminative analysis (LDA) [11] algorithm, where $S_b(p)$ corresponds to the between-class scatter matrix and $\tilde{S}_w(p)$ corresponds to the within-class scatter matrix. Here we borrow the analytical solution from LDA to derive the optimal solution $\mathbf{w}_{opt}$ to Eq. (11) with p fixed:

$$\begin{aligned} &\mathbf{w}_{opt} = \arg\max_{\mathbf{w}} \gamma, \\ &s.t. \quad S_b\mathbf{w} = \gamma\tilde{S}_w\mathbf{w}. \end{aligned} \tag{12}$$

The solution $\mathbf{w}_{opt}$ is the eigenvector corresponding to the largest eigenvalue.

For the optimization of Eq. (11) with respect to p, there is no closed-form solution. We adopt a gradient ascent process to solve p in an iterative manner. Let y denote the pooled feature $y = \mathbf{w}^T\mathbf{v}^p$, thus the between-class and within-class scatter matrices of the pooled features can be written as

$$\begin{aligned} \hat{S}_b(p) \ &= \ \mathbf{w}^T S_b(p)\mathbf{w} \\ &= \sum_i \sum_{j\in N^-_{k_1}(i)} (y_i - y_j)^2, \\ \hat{S}_w(p) \ &= \ \mathbf{w}^T \tilde{S}_w(p)\mathbf{w} \\ &= \sum_i \sum_{j\in N^+_{k_2}(i)} (y_i - y_j)^2 + \lambda\mathbf{w}^T L\mathbf{w}. \end{aligned} \tag{13}$$

We use $\boldsymbol{\alpha}$ to denote the Hadamard product $\boldsymbol{\alpha} = ln\mathbf{v} \circ \mathbf{v}^p$. Then the derivatives of $\hat{S}_b$ and $\hat{S}_w$ with respect to p are as follows:

$$\begin{aligned} \frac{\partial}{\partial p}\hat{S}_b &= 2\sum_i \sum_{j\in N^-_{k_1}(i)} (y_i - y_j)\mathbf{w}^T(\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j), \\ \frac{\partial}{\partial p}\hat{S}_w &= 2\sum_i \sum_{j\in N^+_{k_2}(i)} (y_i - y_j)\mathbf{w}^T(\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j). \end{aligned} \tag{14}$$

The partial derivative of the objective function (Eq. (11)) with respect to p is

$$\nabla p = \frac{\partial}{\partial p}\mathcal{Q} = \frac{1}{\hat{S}_w^2}(\frac{\partial\hat{S}_w}{\partial p}\hat{S}_b - \frac{\partial\hat{S}_b}{\partial p}\hat{S}_w). \tag{15}$$

Thus we update p along the gradient direction with step size β as follows:

$$p^{(t+1)} = p^{(t)} + \beta \nabla p. \tag{16}$$

The process will stop when the change of p is less than a predefined threshold θ_p or the number of iterations exceeds the permitted number N_{iter}. λ is set optimally via three-fold cross-validation on a randomly sampled subset from the training data.

5. Robust adaptive pooling for misaligned image

It is notable that there exist variations of object position/scale in images. However, the discriminative pooling function derived above assumes roughly aligned object foreground region and is not adaptive to the change of object position/scale in testing images. It is therefore preferable to have an adaptive feature pooling scheme where the discriminative pooling function can be tailored for individual images and thus the pooled image representation is robust to misalignment of foreground. In the meantime, such an adjustment procedure should be simple and efficient. To this end, we have two key observations:

1. Spatially nearby and appearance-consistent pixels within an image local region present similar representative and discriminative information, thus the pooling weights with respect to these pixels should be similar. If we construct a pixel adjacency graph and on this graph nearby pixels which belong to the same object region are linked, the learned common discriminative pooling weights (from the training images) could be propagated (diffused) according to this graph. Thus, the incorrect pooling weights on some misaligned image foreground region will be corrected by receiving messages propagated from adjacent nodes (pixels).
2. Visual saliency map [21] provides rough information about the locations of object region-of-interest [25,36,43,47,49]. This information can guide the pooling weights to propagate toward the object region-of-interest (foreground) from background area.

Our basic idea is that for each image, we can construct a directed pixel adjacency graph based on these two types of side information, *i.e.*, pixel similarity and visual saliency. We can then perform message passing algorithm (*e.g.*, random walk) on the graph and eventually propagate the discriminative pooling information onto foreground region of an individual image, and in the meantime, nearby appearance-consistent pixels will share similar pooling weights.

5.1. Adjacency graph construction

The pixel adjacency graph is constructed based on image over-segmentation results. The assumption is that within each image segment, pixels are consistent in appearance and therefore the pooling weights can be diffused among the pixels to achieve similar values. More specifically, given an image, we segment it into super-pixels using simple linear iterative clustering (SLIC) [2]. The advantages of SLIC over-segmentation algorithm are 1) its segmented regions well preserve the object boundaries; and 2) it is very flexible to change the granularity of the segmented regions. We define the segmented regions (super-pixels) for image I as $\{\mathcal{R}_1, \cdots, \mathcal{R}_{N(I)}\}$, where $N(I)$ is the number of segmented regions. We denote by $R(i)$ the region index which contains pixel i, where $i \in \{1, \cdots, M\}$. The parameter of the average size of segments is empirically set as 20 pixels. We define an undirected pixel adjacency graph $G_U = [p^U_{ij}]_{i,j=1,\cdots,M}$, where each edge potential p^U_{ij} (between pixel i and j; the superscript U stands for undirected) is defined as

$$p^U_{ij} = \begin{cases} \exp(- \parallel \mathbf{a}_i - \mathbf{a}_j \parallel_2^2) & \text{if } \mathrm{R(i) = R(j)}, \\ 0, & \text{else}, \end{cases} \tag{17}$$

where $\mathbf{a}_i$ is the spatial 2D coordinate of pixel i.

To direct the pooling weights toward the object region-of-interest, we utilize side information which indicates the foreground region, *i.e.*, visual saliency map. To this end, we compute visual saliency maps using three state-of-the-art methods including graph based visual saliency (GBVS) [18], spectral residual approach (SR) [19], and frequency tuned method (FT) [1]. We use the mean saliency map by averaging the saliency maps computed from these three methods. In practice, combining these three methods outperforms any single method for visual saliency detection [1,18,19]. We then define a directed pixel adjacency graph $G_D = [p^D_{ij}]_{i,j=1,\cdots,M}$, where each edge potential p^D_{ij} (from pixel j to i; the superscript D stands for directed) is defined as

$$p^D_{ij} = \begin{cases} 1, & \text{if } \mathrm{A(i)} > 0, \\ 0, & \text{else}, \end{cases} \tag{18}$$

where $A(i)$ denotes the saliency value on pixel i. We note that on this graph, edges are only linked toward nodes (pixels) that have non-zero saliency values (which indicate foreground region). This indicates that graph algorithms which are based on message passing cannot propagate message from foreground regions (non-zero saliency value) to background regions. This ensures no leakage of pooling weights to the background region. Namely, when we propagate the feature pooling weights on the graph, background regions will not receive pooling weights from adjacent pixels.

The final graph for propagating pooling weights is a directed graph $G = [p_{ij}]_{i,j=1,\cdots,M}$, whose edge is a product of corresponding edge potential of G^U and G^D,

$$\widetilde{p_{ij}} = p^U_{ij} \times p^D_{ij}. \tag{19}$$

We define by $\mathbf{P} = [\widetilde{p_{ij}}]$ the row normalized adjacency matrix of $[\widetilde{p_{ij}}]$. It can also be considered as a transition matrix for transforming the feature pooling weights vector $\mathbf{w}$.

5.2. Pooling weights propagation

Using the constructed adjacency graph introduced above, we can propagate the discriminative common pooling weights by simple random walk. Assuming that the learned discriminative pooling weights vector is $\mathbf{w}$, a single step (order-1 updating) random walk to diffuse the pooling weights vector $\mathbf{w}$ for image I is defined as

$$\mathbf{w}^{(1)} = \mathbf{P}(I)\mathbf{w}. \tag{20}$$

In some cases, a single step random walk is not sufficient to update the pooling weights vector to cope with image foreground misalignment and therefore multi-step (order-n updating) random walk updating is required, which is denoted as

$$\mathbf{w}^{(n)} = \mathbf{P}^{(n)}(I) = \underbrace{\mathbf{P}(I) \cdots \mathbf{P}(I)}_{n}\mathbf{w}. \tag{21}$$

Note that the transition matrix $\mathbf{P}(I)$ is neither aperiodic nor irreducible, thus the iterative process of Eq. (21) does not converge when n approaches infinity. The image-specific pooling weights vector $\mathbf{w}^{(n)}$ is then used as in Eq. (5) to replace $\mathbf{w}$ for pooling visual features of individual images. We also note that this proposed adaptive adjustment for pooling weights is very efficient. For K visual words undergoing order-n updating, only $K \times n$ matrix multiplications are needed. In practice our Matlab implementation on a 3 GHz duo-core PC with 4 GB memory takes less than 0.1 s for a single random walk update on an image. Fig. 3 visualizes the proposed method for adapting the learned common pooling weights to individual images.

5.3. Training

The above mentioned random walk based self-alignment step is therefore utilized to adjust the pooling procedure for individual images (*e.g.*, for image I_i, and we assume order-n update) in terms of Eq. (5) as

$$f_{rg}(\mathbf{v};\mathbf{w}) = (\mathbf{P}^{(n)}(I_i)\mathbf{w})^T\mathbf{v}^p = \mathbf{w}^T(\mathbf{P}^{(n)}(I_i)^T\mathbf{v}^p),\\ s.t.||\mathbf{w}||_2 = 1, p \geq 0, \tag{22}$$

where f_{rg} denotes robust geometric pooling. Here $\mathbf{w}$ is the target discriminative pooling weights shared by all images and $\mathbf{P}^{(n)}(I_i)$ is an image specific random walk term computed from the corresponding side information (saliency map and over-segmentation) extracted for images I_i. From Eq. (22), we note that $\mathbf{w}$ is uncoupled with $\mathbf{P}^{(n)}(I_i)^T\mathbf{v}^p$. Therefore, we can denote $\tilde{\mathbf{v}}^p = \mathbf{P}^{(n)}(I_i)^T\mathbf{v}^p$ and the derived objective function for discriminative pooling is still in the format of Eq. (11), with the scatter matrices S_b and S_w changed to $\tilde{S}_b$ and $\tilde{S}_w$ as

$$\tilde{S}_b(p) = \sum_i \sum_{j \in N_{k_1}^-(i)} (\tilde{\mathbf{v}}_i^p - \tilde{\mathbf{v}}_j^p)(\tilde{\mathbf{v}}_i^p - \tilde{\mathbf{v}}_j^p)^T,\\ \tilde{S}_w(p) = \sum_i \sum_{j \in N_{k_2}^+(i)} (\tilde{\mathbf{v}}_i^p - \tilde{\mathbf{v}}_j^p)(\tilde{\mathbf{v}}_i^p - \tilde{\mathbf{v}}_j^p)^T,\\ \tilde{\mathbf{v}}_i^p = \mathbf{P}^{(n)}(I_i)^T\mathbf{v}_i^p, \quad \forall i. \tag{23}$$

Therefore we can use the same optimization scheme in Section 4.3 to optimize with respect to $\mathbf{w}$. To optimize with respect to p, the same scheme as in Section 4.3 is used with the derivatives of $\hat{S}_b$ and $\hat{S}_w$ being changed to

$$\frac{\partial}{\partial p}\tilde{S}_b = 2\sum_i \sum_{j \in N_{k_1}^-(i)} (\tilde{y}_i - \tilde{y}_j)\mathbf{w}^T(\mathbf{P}^{(n)}(I_i)^T\boldsymbol{\alpha}_i\\ -\mathbf{P}^{(n)}(I_j)^T\boldsymbol{\alpha}_j),\\ \frac{\partial}{\partial p}\tilde{S}_w = 2\sum_i \sum_{j \in N_{k_2}^+(i)} (\tilde{y}_i - \tilde{y}_j)\mathbf{w}^T(\mathbf{P}^{(n)}(I_i)^T\boldsymbol{\alpha}_i\\ -\mathbf{P}^{(n)}(I_j)^T\boldsymbol{\alpha}_j), \tag{24}$$

where we define $\tilde{y}_i$ as $\tilde{y}_i = \mathbf{w}^T\mathbf{P}^{(n)}(I_i)^T\mathbf{v}^p$.

5.4. Testing

For a testing image I_i, we first calculate its image specific random walk matrix $P^{(n)}(I_i)$ (*i.e.*, order-n) and it is then straightforward to apply the image specific pooling weights $\mathbf{P}^{(n)}(I_i)\mathbf{w}$ and the learned optimal p to form the image-level representation.

6. Experiments

In this section, we evaluate the performance of the proposed GLP method as well as its enhanced version RGLP handling misalignment and compare it with the state-of-the-art average and max pooling methods. First, we investigate the separability of the pooling results produced by GLP and the other two methods on a synthesized dataset, which possesses distinctive spatial distribution patterns for different classes. Then we evaluate GLP and RGLP along with the average and max pooling on real-world datasets for image classification and action recognition: Caltech-101 dataset [30], Caltech-256 dataset [16], 15 scene dataset [27], Indoor 67 dataset [38] and

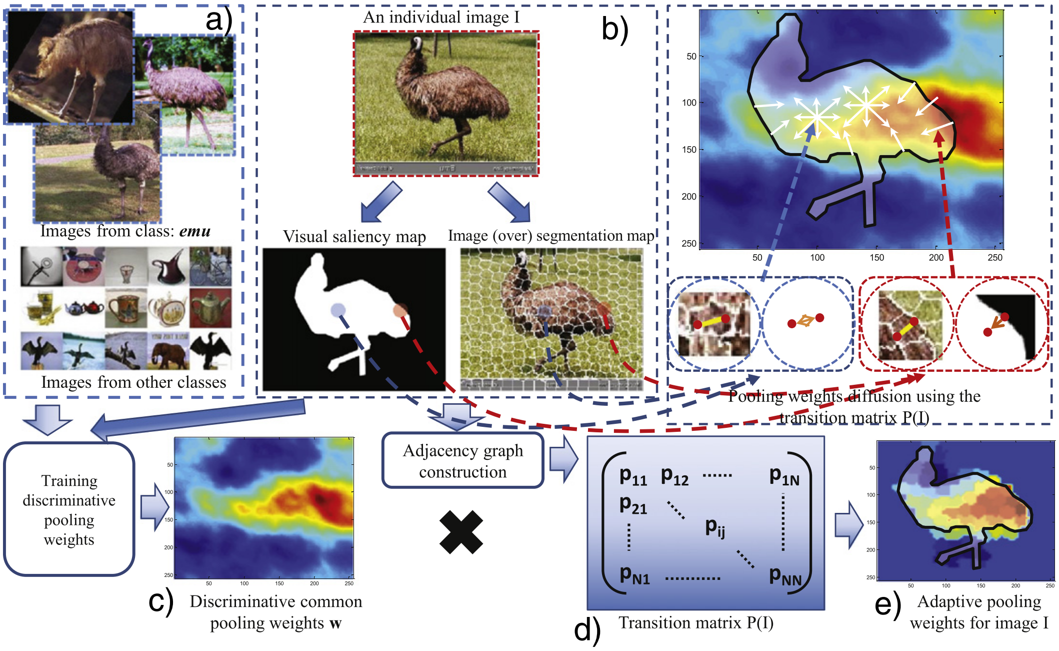
a)
Images from class: *emu*
Images from other classes
An individual image I
b)
Visual saliency map
Image (over) segmentation map
50
100
150
200
250
50
100
150
200
250
Pooling weights diffusion using the transition matrix P(I)
Training discriminative pooling weights
c)
Discriminative common pooling weights **w**
Adjacency graph construction
p11 p12 ...... p1N
p21
pij
pN1 .... pNN
d)
Transition matrix P(I)
e)
Adaptive pooling weights for image I

Fig. 3. A diagram to visualize the random walk based self-alignment method which adjusts the pooling weights for an individual image, based on the side information provided by visual saliency detection and image over-segmentation. Note that the learned common pooling weights are propagated toward the detected foreground (salient) region. (a) denotes the input images. For an individual image, we calculate its saliency map and over-segmentation map illustrated in (b) and the adjacency graph for pooling weights propagation (d) is calculated based on (b). The learned common discriminative pooling weights map (c) is input to the pooling weights propagation (transition) matrix (d) and the output is the desired image adaptive pooling weights map (e).

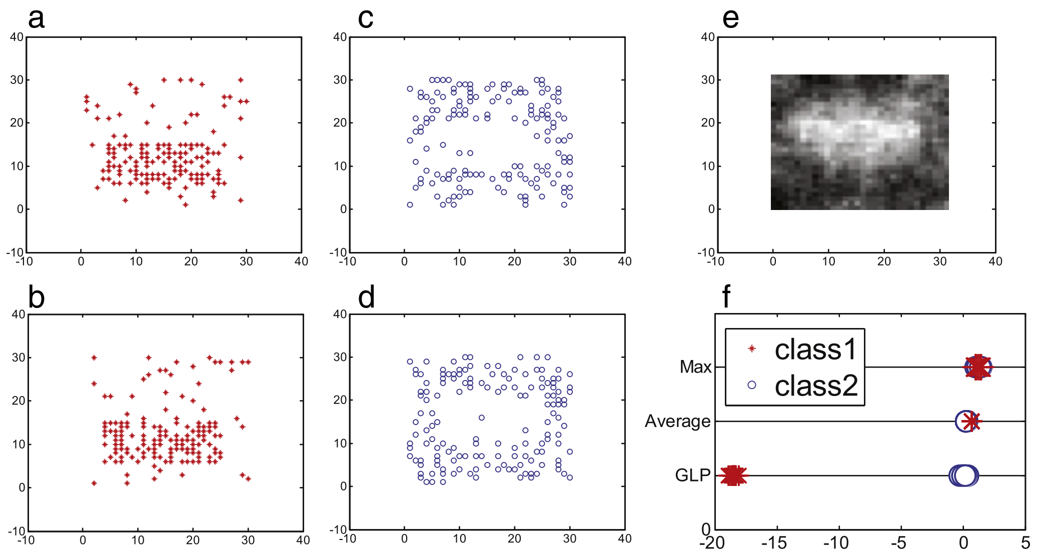

Fig. 4. Comparison of GLP and average/max pooling over the synthesized data with distinctive feature distributions for different classes. (a), (b) and (c), (d) show the exemplar data from two different classes respectively. (e) displays the optimized geometric coefficients over the region. Brighter pixels mean that the coefficients are larger at the corresponding locations. (f) shows the pooling results distribution via the average, max and GLP poolings. It can be seen that GLP can separate the data from two classes well while average pooling and max pooling cannot.

human–object interaction (HOI) activity dataset [17]. These datasets are widely used in recent years so that we can compare with state-of-the-art methods conveniently, and evaluate the proposed algorithms extensively. Note that our task is for single-label image classification, thus PASCAL VOC datasets are not used for experiment.

6.1. Experiment on synthetic data

A set of randomly generated data is used to investigate the effectiveness of feature spatial distribution for the classification purpose. The synthesized dataset comprises two classes of data, with distinctive spatial distribution per class. There are 200 data matrices for each class. The size of the matrix is fixed as 30 × 30 to simulate an image with 30 × 30 feature locations. Each element of the matrix is a binary variable to indicate the presence of a certain visual feature at the corresponding location. Random transitional noise with magnitude ranging from 1 to 20 locations is added to each datum. Fig. 4 shows two exemplar data from different classes. We perform average pooling, max pooling and GLP on this dataset and plot the distributions of the pooling results in Fig. 4. From the derived pooling-feature distribution, it can be seen that neither average pooling nor max pooling can well separate these two classes due to the loss of the spatial information, while GLP properly utilizes the features' class-specific spatial distributions and the resultant statistics are separable. GLP produces a discriminative pooling coefficients map as shown in Fig. 4.

6.2. Experiment on image classification

In this subsection, we continue the comparison on real image datasets for image classification and image-based action recognition. The purposes of the experiments are two-fold. The first one is to compare GLP directly with the other two pooling methods. The second one is to evaluate the performance of the new image classification framework which includes GLP and RGLP as a new plug-in, and compare it with the state-of-the-art methods.

The classification performances based on these three pooling methods are compared under two different feature representing schemes (if not otherwise mentioned): one is based on the hard assignment and the other is based on the combination of locality-constrained linear coding (LLC) [50] and spatial pyramid matching (SPM) [27]. For Caltech-101/256 and 15 scene datasets, we perform multiple runs (20), and both the mean and the standard deviation of the classification accuracies are reported. For Indoor 67 dataset, we follow the training/testing image set partition provided in [38].

6.2.1. Experimental configurations

These two groups of experiments follow the common experimental settings. Although various types of image features can be explored, we only use a single type of local descriptor, dense SIFT [35], throughout the experiments. The SIFT features are extracted from densely located patches centered at every 4 pixels on the images and the size of the patches is fixed as 16 × 16 pixels. We construct a visual word dictionary containing K words from the training samples via K-means clustering. The value of K depends on the number of samples and varies across different datasets. Each SIFT feature vector is encoded into a K-dimensional code vector based on the dictionary. Then the code vectors from every image are pooled into a single feature vector via different pooling methods. During the training process of the GLP (and RGLP), we use all the training samples to calculate the MFA scatter matrices and we set $k_1 = 20$ and $k_2 = 20$. The maximum number of the alternations between optimizing $\mathbf{w}$ and p is fixed as $N_{alter} = 10$. The stopping threshold of updating p and $\mathbf{w}$ is $\theta_p = 0.1$ or $\theta_{\mathbf{w}} = 0.01$, respectively. The optimal values of λ_1 and λ_2 are set by three-fold cross-validation on a randomly sampled subset of the training dataset. The pooled features are used to train a multi-class linear SVM. In our experiments, all the images are resized to 256 × 256 pixels.

In the first group of experiments, GLP is directly compared with other pooling methods, and the code vectors are generated by hard assignment. In the second group of experiments, we apply the GLP and RGLP as a new pooling component in the multi-stage image classification architecture proposed in [50]. The original architecture consists of four components: image local feature extraction, feature encoding, feature pooling and spatial pyramid matching (SPM) [27], followed by linear SVM classifier. Here, we replace the max pooling component with the GLP/RGLP method, and compare the image classification performance with the original one and other state-of-the-art methods. We follow the same experimental setting as

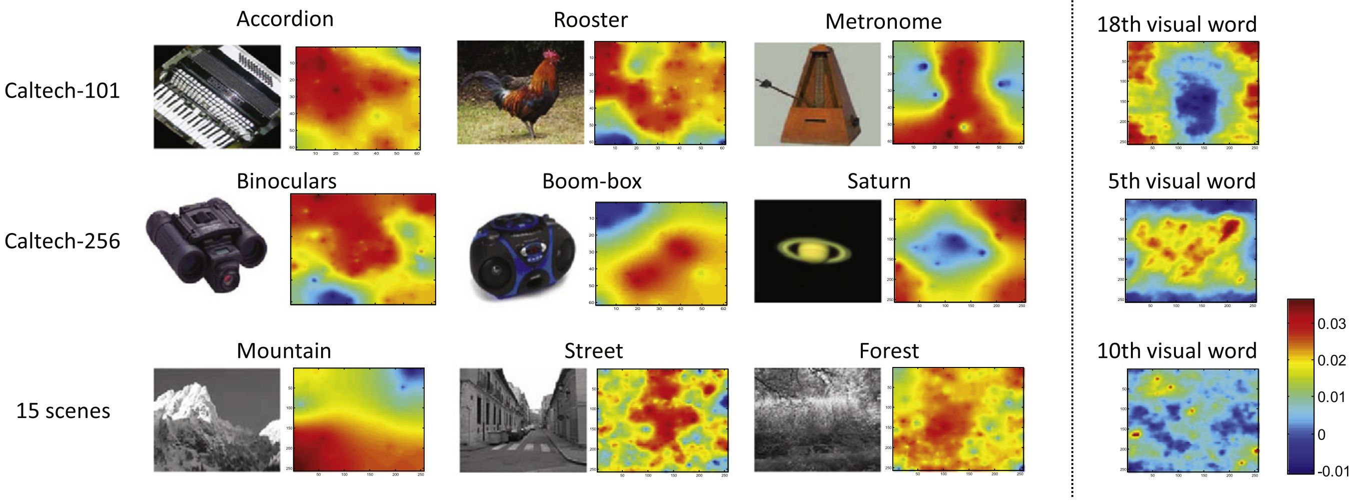

Fig. 5. Visualization of the pursued geometric coefficient maps for each specific visual word over different classes. The left 6 columns show the exemplar images from 3 classes per dataset and their corresponding geometric coefficient distribution maps. The coefficients for one specific class are computed in one-vs-all manner. The right most column shows the geometric coefficients for one specific visual word, derived from GLP over all the classes. Each row displays one dataset. For better view, please refer to the color version.

in [50]. SIFT features are encoded by locality-constrained cinear oding (LLC) [50] and the number of neighbors is fixed as 5. Images are hierarchically partitioned into 1 × 1, 2 × 2 and 4 × 4 blocks on 3 levels respectively in the SPM.

We compare the performance of our proposed method with various state-of-the-art image classification methods. The comparing methods include the following:

1. the spatial pyramid matching kernel method (KSPM) in [27];
2. the sparse coding+spatial pyramid matching method (ScSPM) [55];
3. the discriminative nearest neighbor method (SVM-KNN) [62];
4. the naive Bayesian nearest neighbor method (NBNN) [4];
5. the kernel codebooks (KC) method [15];
6. the metric learning based hashing+CORR kernel method (ML+CORR) [22];
7. the latent pyramidal region (LPR) method in [41];
8. the locally linear encoding method (LLC) in [50];
9. the linear distance coding method (LSA) in [51];
10. the smooth sparse coding method (SSC) in [3];
11. the Laplacian sparse coding method (LScSPM) in [13];
12. the multiple kernel object feature combination method (LP-β-MKL) in [14]; and
13. the dense spatial sampling and pooling method (BSPR) in [54].

Note that we directly report the published results for these state-of-the-art methods as all methods follow the same experimental settings.

Table 1
Accuracy comparison of image classification using hard assignment for three different pooling methods.

	Caltech-101	Caltech-256	15 scenes
Average	44.2 ± 0.5	22.0 ± 0.7	56.7 ± 0.6
Max	48.1 ± 0.6	18.5 ± 0.6	54.2 ± 0.5
GLP	**54.9** ± 0.5	**33.3** ± 0.4	**64.9** ± 0.5
RGLP	**56.7** ± 0.6	**38.7** ± 0.5	**66.8** ± 0.4

The results of proposed methods in this paper are shown in bold.

6.2.2. Visualization of geometric ℓ_p-norm pooling effects

We visualize the geometric coefficients in Fig. 5 learned by GLP, from which it can be seen that the coefficients derived from GLP are able to capture the visual word spatial distributions.

6.2.3. Results on Caltech-101 dataset

The Caltech-101 dataset [30] contains 9144 images in total from 102 different categories, including 101 object categories and 1 additional background category. The number of images per category ranges from 31 to 800. The resolution of most images is about 300 × 300 pixels. Following the setting in [50] and [29], we randomly select 5, 15 and 30 images respectively for training and report the classification accuracies averaged over the 102 categories. The size of visual word dictionary is set as $K = 2048$ as in [50].

The performance comparison of different pooling methods based on the hard-assignment encoding scheme is shown in the first column of Table 1. It can be seen that GLP consistently outperforms the average and max pooling by a margin of 11% and 7% respectively. The robust version of GLP (RGLP) performs better than the original GLP, as RGLP alleviates the image foreground misalignment issue. The classification accuracy of GLP combined with LLC and SPM is shown in Table 2. Based on the comparison with the original LLC [50], it can be observed that the performance improvement brought by GLP is nearly 8% when using 30 training samples. Also GLP outperforms all the single type of feature based methods. And the robust version of

Table 2
Classification accuracy (%) comparison on Caltech-101 dataset.

Algorithms	5 training	15 training	30 training
SVM-KNN [62]	46.6	59.1 ± 0.6	66.2 ± 0.5
KSPM [27]	–	56.4	64.6 ± 0.8
NBNN [4]	–	65.0 ± 1.1	70.4
ML+CORR [22]	–	61.0	69.6
KC [15]	–	–	64.1 ± 1.2
ScSPM [55]	–	67.0 ± 0.5	73.2 ± 0.5
SSC [3]	–	–	81.0 ± 1.2
LLC [50]	51.2	65.4	73.4
LSA [51]	–	–	74.6 ± 0.5
LP-β-MKL [14]	–	–	77.7 ± 0.3
GLP	**58.3** ± 0.5	**69.3** ± 0.4	**81.9** ± 0.5
RGLP	**60.6** ± 0.5	**71.8** ± 0.8	**83.7** ± 0.6

The results of proposed methods in this paper are shown in bold.

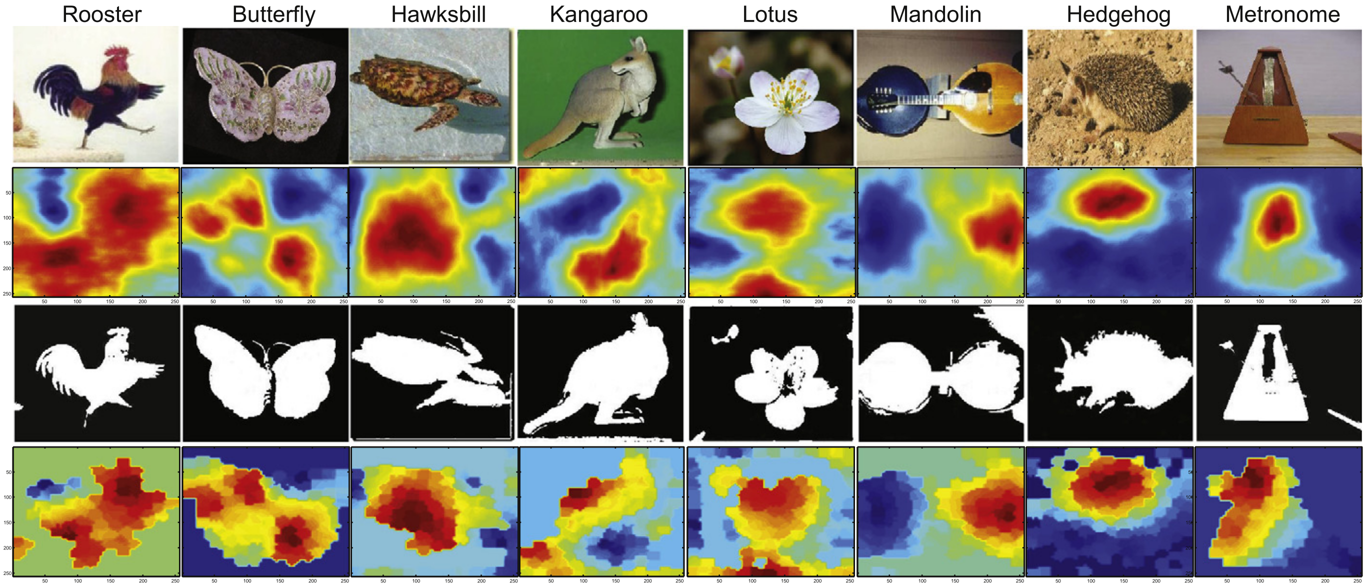

Fig. 6. Examples of the learned common discriminative pooling weights and the adapted pooling weights for individual images based on the proposed random walk based self-alignment step. From the top row to the bottom row: 1) input image; 2) the learned common discriminative pooling weights for each class; 3) the segmented saliency region; and 4) the adapted pooling weights for the input image. Color code can be found in Fig. 5. For better view, please refer to the color version.

GLP (RGLP) performs better than the original GLP due to its capability in dealing with image foreground misalignment. The performance of our proposed method has already exceeded the best one (82.3%) ever reported on the Caltech-101 dataset in [29]. This result is very encouraging as the method in [29] utilizes the groundtruth segmentation, which is not available for real applications. Also, [29] uses 8 different types of features in total. In contrast, our method only uses one single type of feature (dense SIFT) and needs no groundtruth image segmentation results to be provided. In Fig. 6 we also visualize some example geometric coefficient maps for each specific visual word over different classes, before and after order-3 self-alignment. We see that the proposed random walk based self-alignment step can effectively adjust the spatial pooling weights toward the object foreground region.

6.2.4. Results on Caltech-256 dataset

Caltech-256 [16] is an extension of the Caltech-101 dataset. It consists of 256 object categories and contains from 80 to 827 images per category. The total number of images is 30,608. This dataset possesses larger intra-class variability than the Caltech-101 and thus is more challenging. As in [50], 15, 30 and 45 images from each category are used for training respectively, and we use a 4096-D visual word dictionary as in [50]. As can be seen from the second column of Table 1, GLP also consistently leads the performance compared with other pooling methods on this dataset under hard-assignment setting. Also our method outperforms the state-of-the-art method (LLC) on this dataset, with a margin of 2% as shown in Table 3. Note that Caltech-256 has more serious image foreground misalignment issue than other datasets, therefore, we can observe that the performance gain by our random walk based self-alignment step, *i.e.*, RGLP, outperforms the original GLP significantly and it achieves the best performance among all comparing methods.

Table 3
Classification accuracy (%) comparison on Caltech-256 dataset.

Algorithms	15 training	30 training	45 training
KSPM [16]	–	34.1	–
ScSPM [55]	27.7 ± 0.5	34.0 ± 0.4	37.5 ± 0.6
LScSPM [13]	30 ± 0.1	35.7 ± 0.1	38.5 ± 0.4
LLC [50]	34.4	41.2	45.3
KC [15]	–	27.2 ± 0.5	–
LSA [51]	–	38.4 ± 0.1	–
LP-β-MKL [14]	–	45.8	–
BSPR [54]	–	46.8 ± 0.2	50.7 ± 0.2
GLP	**35.1** ± 0.3	**43.0** ± 0.4	**46.5** ± 0.4
RGLP	**38.8** ± 0.4	**47.5** ± 0.3	**51.4** ± 0.5

The results of proposed methods in this paper are shown in bold.

6.2.5. Results on 15 scene dataset

Scene-15 dataset is composed of 15 scene classes. Each class contains 200 to 400 images and there are 4485 images in total. The scene categories contain from the out-door street and industry to the in-door kitchen and living room. As in [27,55], we randomly select 100 images from each class as training samples to construct a 1024-D visual dictionary. The improvements brought by GLP (RGLP) over the average and max pooling are about 9% and 11% respectively as shown in the third column of Table 1 under hard-assignment setting. Also from Table 4 we can see that GLP under the setting with LLC and SPM can improve the classification performance further with a margin of 4% compared with LLC and outperforms KSPM by nearly 2%. We note that on this dataset, Laplacian sparse coding (LScSPM) based feature encoding method achieves very high recognition accuracy. Motivated by this, we also report the performance by our GLP and RGLP pooling frameworks using the Laplacian sparse

Table 4
Classification accuracy (%) comparison on 15 scene dataset.

Algorithms	Accuracy	Algorithms	Accuracy
KSPM [27]	81.4 ± 0.5	BSPR [54]	88.0 ± 0.6
ScSPM [55]	80.3 ± 0.9	SSC [3]	90.2 ± 2.9
LLC [50]	79.8 ± 0.4	GLP	**83.4** ± 0.5
LSA [51]	82.5 ± 0.5	RGLP	**83.8** ± 0.4
LScSPM [13]	89.8 ± 0.5	GLP (LScSPM)	**91.2** ± 0.4
LPR [41]	85.8	RGLP (LScSPM)	**91.4** ± 0.5

The results of proposed methods in this paper are shown in bold.

Table 5
Classification accuracy (%) comparison on Indoor 67 dataset.

Algorithms	Accuracy	Algorithms	Accuracy
ROI+GIST [38]	26.5	LLC [50]	46.3
Object bank [24]	37.6	LSA [51]	46.7
LPR [41]	44.8	GLP	**51.5**
MID-DP [48]	49.4	RGLP	**51.8**

The results of proposed methods in this paper are shown in bold.

coding. That is, we replace LLC coding with Laplacian sparse coding, and the results are denoted as GLP (LScSPM) and RGLP (LScSPM). We see the combination of our pooling framework with the Laplacian sparse coding method achieves the best recognition accuracy among all comparing methods. This further demonstrates that the proposed GLP and RGLP feature pooling framework can be flexibly integrated with any feature extraction and coding component in the image representation pipeline (as shown in Fig. 2) to boost the recognition accuracy. Another observation is that on this dataset, RGLP does not outperform GLP obviously. This is because the scene images do not possess strong foreground/background separation characteristics and the visual features extracted from all regions of the image are important for representing the image. Therefore, the proposed self-alignment step is not as necessary as it is in object recognition.

6.2.6. *Results on indoor 67 datasets*

Indoor 67 dataset [38] contains 67 indoor scene categories, and a total of 15,620 images. All images have a minimum resolution of 200 pixels along the smaller axis. We follow the settings of the baseline method provided in [38], *i.e.*, 80 images of each class used for training and 20 images for testing. The recognition accuracy comparisons are shown in Table 5. The current state-of-the-art accuracy is achieved by the middle-level discriminative patch method (MID-DP) [48]. Note that the proposed method significantly outperforms the state-of-the-art. We also note that RGLP does not significantly outperform GLP for this dataset. The reason is the same as that of the 15 scene dataset.

6.3. *Experiments on action recognition*

The well-known human–object interaction (HOI) activity dataset contains six activity classes [17]: cricket-defensive shot (player and cricket bat), cricket-bowling (player and cricket ball), croquet-shot (player and croquet mallet), tennis-forehand (player and tennis racket), tennis-serve (player and tennis racket), and volleyball-smash (player and volleyball). There are 50 images in each activity class. We follow the same setting as that in [17]: 30 images for training and 20 for testing, and we use a 4096-D visual word dictionary as in [9].

The recognition accuracy comparisons are shown in Table 6. We compare with previous state-of-the-art results reported in [9,17,57]. the proposed method significantly outperforms the state-of-the-art. We also note that RGLP does not significantly outperform GLP for this dataset. The reason is the same as that of the 15 scene dataset. It is observed that is that on this dataset, the proposed methods show to be effective, and RGLP outperforms GLP obviously. This is because the foreground/background features have clear different influences for activity images recognition, which is differently from that in scene images classification. The proposed self-alignment step is quite useful in human action recognition.

Table 6
Classification accuracy (%) comparison on action recognition dataset.

Algorithms	Accuracy	Algorithms	Accuracy
Gupta et al. [17]	78.7	LSVM+BOF Image [9]	85
Yao and Fei-Fei [57]	83.3	GLP	**87.9**
BOF Image [9]	85	RGLP	**89.6**

The results of proposed methods in this paper are shown in bold.

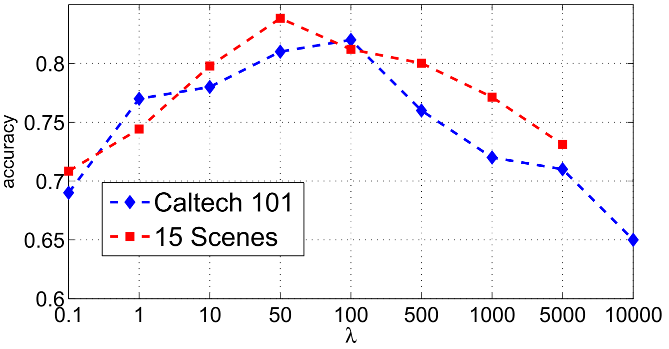

Fig. 7. The effect of smooth factor λ on the classification accuracy. The plot is based on the classification result on Caltech-101 and 15 scene dataset, using 30 and 100 training samples respectively.

6.3.1. *Discussions on parameter selection*

The spatial smooth factor λ in Eq. (11) models the strength of the correlation among adjacent local features. During the GLP pooling procedure, its value controls the homogeneous degree of the geometric coefficients w_i over the region of interest. Here we plot the classification accuracy curve with respect to different values of λ in Fig. 7. When the value of λ is very small, the features are assumed to be distributed independently and no correlations are taken into account. On the contrary, when the value of λ approaches infinity, the spatial smoothness term will dominate and enforce all the local features to follow the uniform distribution. As the accuracy curve shows, neither the independent assumption as in [6] nor the uniform distribution as the average pooling adopts is optimal. The optimal correlation model actually lies in between these two extremes.

There is no theoretical guarantee about what is the optimal order (n) for the random walk based self-alignment step. We plot in Fig. 8 the classification accuracies according to different values for n, for Caltech-101 and Caltech-256 datasets respectively. One can note that the major performance gain is from $n = 0$ (no adjustment) to $n = 1$ and further increase of the value of n does not bring significant improvement. Therefore in this work, if not otherwise stated, we choose $n = 1$ as the default experimental setting, for the trade off between performance and complexity.

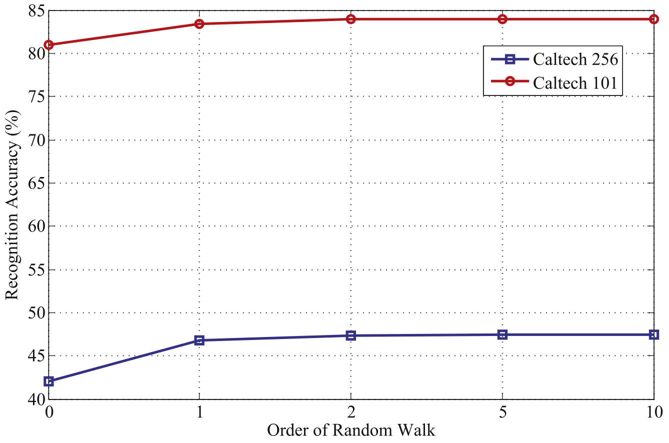

Fig. 8. The effect of the order of random walk of RGLP on the classification accuracy. The plot is based on the classification result on Caltech-101 and Caltech-256 datasets, using 30 training samples.

7. Conclusion

In this work, we first proposed a geometric ℓ_p-norm pooling (GLP) method to perform feature pooling. Different from traditional feature pooling methods, *e.g.* the average and max pooling, the GLP method can utilize the geometric information of the feature spatial distributions and thus provide more discriminative pooling results. Second, we proposed a simple yet effective random walk based image self-alignment step to alleviate the foreground misalignment issue in geometric ℓ_p-norm feature pooling, which results in an image-adaptive discriminative pooling scheme (RGLP). Comprehensive experimental results on several benchmarks have demonstrated that the proposed GLP and RGLP can serve as a highly effective building block for the image classification architecture and boost the performance to outperform the state-of-the-arts.

Acknowledgments

This work is supported by the National Natural Science Foundation (NSF) of China (No. 61572029, No. 61300056), and the Science and Technology Project of Anhui Province (No. 1501b042207).

References

[1] R. Achanta, S. Hemami, F. Estrada, S. Susstrunk, Frequency-tuned salient region detection, Computer Vision and Pattern Recognition, 2009. pp. 1597–1604.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk, SLIC Superpixels Compared to State-of-the-art Superpixel Methods, IEEE Trans. Pattern Anal. Mach. Intell. 34 (11) (2012) 2274–2282.

[3] K. Balasubramanian, K. Yu, G. Lebanon, Smooth sparse coding via marginal regression for learning sparse representations, International Conference on Machine Learning, 2013.

[4] O. Boiman, E. Shechtman, M. Irani, In defense of Nearest-Neighbor based image classification, Computer Vision and Pattern Recognition, 2008.

[5] A. Bosch, A. Zisserman, X.M.u. noz, Image Classification using Random Forests and Ferns, International Conference on Computer Vision, 2007.

[6] Y. Boureau, J. Ponce, Y. LeCun, A Theoretical Analysis of Feature Pooling in Visual Recognition, International Conference on Machine Learning, 2010.

[7] Y. Chai, V. Lempitsky, A. Zisserman, BiCoS: a bi-level co-segmentation method for image classification, IEEE International Conference on Computer Vision, 2011. pp. 2579–2586.

[8] Q. Chen, Z. Song, Y. Hua, Z. Huang, S. Yan, Hierarchical Matching with Side Information for Image Classification, Computer Vision and Pattern Recognition, 2011.

[9] V. Delaitre, J. Sivic, Recognizing human actions in still images: a study of bag-of-features and part-based representations, 21st British Machine Vision Conference, 2010.

[10] J. Feng, B. Ni, Q. Tian, S. Yan, Geometric Lp-norm feature pooling for image classification, Computer Vision and Pattern Recognition, 2011.

[11] R. Fisher, The Use of Multiple Measurements in Taxonomic Problems, Ann. Eugen. 7 (2) (1936) 179–188.

[12] K. Fukushima, S. Miyake, Neocognitron: a new algorithm for pattern recognition tolerant of deformations and shifts in position, Pattern Recogn. 15 (6) (1982) 455–469.

[13] S. Gao, I.W. Tsang, L.-T. Chia, P. Zhao, Local features are not lonely — Laplacian sparse coding for image classification, Computer Vision and Pattern Recognition, 2010.

[14] P. Gehler, S. Nowozin, On feature combination for multiclass object classification, International Conference on Computer Vision, 2009. pp. 221–228.

[15] J. Gemert, J. Geusebroek, C. Veenman, A. Smeulders, Kernel Codebooks for Scene Categorization, European Conference on Computer Vision, 2008.

[16] G. Griffin, A. Holub, P. Perona, Caltech-256 Object Category Dataset, California Institute of Technology 2007.

[17] A. Gupta, A. Kembhavi, L. Davis, Observing human–object interactions: using spatial and functional compatibility for recognition, IEEE T. Pattern Anal. 31 (10) (2009) 1775C1789

[18] J. Harel, C. Koch, P. Perona, Graph-Based Visual Saliency, Advances in Neural Information Processing Systems, 2006. pp. 545–552.

[19] X. Hou, L. Zhang, Saliency detection: a spectral residual approach, Computer Vision and Pattern Recognition, 2007.

[20] D. Hubel, T. Wiesel, Receptive fields, binocular interaction and functional architecture in the cat's visual cortex, J. Physiol. 160 (1) (1962) 106–154.

[21] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, IEEE Trans. Pattern Anal. Mach. Intell. 20 (11) (1998) 1254–1259.

[22] P. Jain, B. Kulis, K. Grauman, Fast image search for learned metrics, Computer Vision and Pattern Recognition, 2008.

[23] K. Jarrett, K. Kavukcuoglu, M. Ranzato, Y. LeCun, hat is the best multi-stage architecture for object recognition?, International Conference on Computer Vision, 2009.

[24] L. jia Li, H. Su, E.P. Xing, L. Fei-fei, Object bank: a high-level image representation for scene classification and semantic feature sparsification, Advances in Neural Information Processing Systems, 2010.

[25] C. Kanan, G. Cottrell, Robust classification of objects, faces, and flowers using natural image statistics, Computer Vision and Pattern Recognition, 2010.

[26] S. Lazebnik, M. Raginsky, Supervised Learning of Quantizer Codebooks by Information Loss Minimization, IEEE Trans. Pattern Anal. Mach. Intell. 31 (7) (2009) 1294–1309.

[27] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, Computer Vision and Pattern Recognition, 2006.

[28] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, L. Jackel, Handwritten digit recognition with a back-propagation network, Advances in Neural Information Processing Systems, 1989.

[29] F. Li, J. Carreira, C. Sminchisescu, Object recognition as ranking holistic figure–ground hypotheses, Computer Vision and Pattern Recognition, 2010.

[30] F. Li, R. Fergus, P. Perona, Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, CVPR Workshop, 2004.

[31] F. Li, P. Perona, A Bayesian hierarchical model for learning natural scene categories, Computer Vision and Pattern Recognition, 2005.

[32] Z. Li, J. Liu, J. Tang, H. Lu, Robust Structured Subspace Learning for Data Representation, IEEE Trans. Pattern Anal. Mach. Intell. 37 (10) (2015) 2085–2098.

[33] Z. Li, J. Liu, Y. Yang, X. Zhou, H. Lu, Clustering-guided sparse structural learning for unsupervised feature selection, IEEE Trans. Knowl. Data Eng. 26 (9) (2014) 1-1

[34] Z. Li, J. Tang, Unsupervised Feature Selection Via Nonnegative Spectral Analysis Redundancy Control, IEEE Trans. Image Process. 24 (12) (2015) 5343–5355.

[35] D. Lowe, Distinctive Image Features from Scale-invariant Keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.

[36] F. Moosmann, D. Larlus, F. Jurie, Learning saliency maps for object categorization, ECCV Workshop on the Representation and Use of Prior Knowledge in Vision, 2006.

[37] B. Ni, S. Yan, A. Kassim, Contextualizing histogram, Computer Vision and Pattern Recognition, 2009.

[38] A.Torralba, A. Quattoni, Recognizing indoor scenes, IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[39] M. Ranzato, Y. Boureau, Y. LeCun, Sparse feature learning for deep belief networks, Advances in Neural Information Processing Systems, 2007.

[40] M. Riesenhuber, T. Poggio, Hierarchical models of object recognition in cortex, Nat. Neurosci. 11 (1999) 1019–1025.

[41] F. Sadeghi, M.F. Tappen, Latent pyramidal regions for recognizing scenes, European Conference on Computer Vision, 2012.

[42] T. Serre, L. Wolf, T. Poggio, Object Recognition with features inspired by visual cortex, Computer Vision and Pattern Recognition, 2005.

[43] F. Shahbaz Khan, J. van de Weijer, M. Vanrell, Top-down color attention for object recognition, International Conference on Computer Vision, 2009. pp. 979–986.

[44] L. Shao, L. Liu, X. Li, Feature Learning for Image Classification Via Multiobjective Genetic Programming, IEEE Trans. Neural Netw. Learn. Syst. 25 (7) (2014) 1359–1371.

[45] L. Shao, L. Liu, M. Yu, Kernelized Multiview Projection for Robust Action Recognition, Int. J. Comput. Vis. (2015) 1–15.

[46] L. Shao, X. Zhen, D. Tao, X. Li, Spatio-temporal Laplacian pyramid coding for action recognition, IEEE Trans. Cybern. 44 (6) (2014) 817–827.

[47] G. Sharma, F. Jurie, C. Schmid, Discriminative spatial saliency for image classification, Computer Vision and Pattern Recognition, 2012. pp. 3506–3513.

[48] S. Singh, A. Gupta, A.A. Efros, Unsupervised discovery of mid-level discriminative patches, European Conference on Computer Vision, 2012.

[49] G. Wang, D. Forsyth, Joint learning of visual attributes, object classes and visual saliency, International Conference on Computer Vision, 2009.

[50] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained cinear oding for image classification, Computer Vision and Pattern Recognition, 2010.

[51] Z. Wang, J. Feng, S. Yan, H. Xi, Linear distance coding for image classification, IEEE Trans. Image Process. 22 (2) (2013) 537–548.

[52] O. Yakhnenko, J. Verbeek, Region-Based Image Classification with a Latent SVM Model, Technical report, INRIA, 2011.

[53] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: a general framework for dimensionality reduction, IEEE Trans. Pattern Anal. Mach. Intell. 29 (1) (2007) 40–51.

[54] S. Yan, X. Xu, D. Xu, S. Lin, X. Li, Beyond spatial pyramids: a new feature extraction framework with dense spatial sampling for image classification, European Conference on Computer Vision, 2012. pp. 473–487.

[55] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, Computer Vision and Pattern Recognition, 2009.

[56] L. Yang, R. Jin, R. Sukthankar, F. Jurie, Unifying discriminative visual codebook generation with classifier training for object category reorganization, Computer Vision and Pattern Recognition, 2008.

[57] B. Yao, X. Jiang, A. Khosla, A.L. Lin, L. Guibas, L. Fei-Fei, Human action recognition by learning bases of action attributes and parts, Proceedings of the 2011 International Conference on Computer Vision, ICCV '11, 2011. pp. 1331–1338.

[58] J. Yu, Y. Rui, B. Chen, Exploiting Click Constraints and Multi-view Features for Image Re-ranking, IEEE Trans. on Multimedia 16 (1) (2014) 159–168.

[59] J. Yu, Y. Rui, Y.Y. Tang, D. Tao, High-order distance-based multiview stochastic learning in image classification., IEEE Trans. Cybern. 44 (12) (2014) 2431–2442.

[60] J. Yu, Y. Rui, D. Tao, Click prediction for web image reranking using multimodal sparse coding., IEEE Trans. Image Process. 23 (5) (2014) 2019–2032.

[61] M. Yu, L. Liu, L. Shao, Structure-Preserving Binary Representations for RGB-D Action Recognition, IEEE Trans. Softw. Eng. (2015) 1-1.

[62] H. Zhang, A. Berg, M. Maire, J. Malik, SVM-KNN: discriminative nearest neighbor classification for visual category recognition, Computer Vision and Pattern Recognition, 2006.

[63] F. Zhu, L. Shao, Weakly-Supervised Cross-Domain Dictionary Learning for Visual Recognition, Int. J. Comput. Vis. 109 (109) (2014) 42–59.