Simple Techniques Make Sense: Feature Pooling and Normalization for Image Classification

Lingxi Xie, Qi Tian, *Senior Member, IEEE*, and Bo Zhang

***Abstract*— Image classification is a fundamental task in computer vision, implying a wide range of challenging problems, such as object recognition, scene understanding, and image tagging. One of the most popular approaches to image classification, the bag-of-features (BoF) model, represents an image with a long feature vector and adopts machine learning algorithms for training and testing. Owing to its simplicity and scalability, the BoF model is widely used in both academic research studies and industrial applications. This paper discusses the feature summarization stage, including pooling and normalization, in the BoF model. We show that these two modules, although devalued sometimes, have important impacts on image classification performance. We present two algorithms, i.e., generalized regular spatial pooling for constructing a better group of spatial bins and hierarchical feature normalization for assigning proper weights for regional feature normalization. Both algorithms are independent of the descriptor extraction and feature encoding stages, and therefore, they could be freely transplanted onto many other classification frameworks based on local feature statistics. We further provide insightful discussions for the nature of designing efficient image classification models. Experiments verify that the proposed algorithm achieves state-of-the-art results on a wide range of image classification data sets.**

***Index Terms*— Bag-of-features (BoF) model, experiments, feature normalization, feature pooling, image classification.**

I. Introduction

IMAGE classification is a long-lasting battle in computer vision. It is a basic task toward image understanding and implies a wide range of applications, including object recognition, scene understanding, image tagging and recommendation, and so on. Recent years have also witnessed the emersion of fine-grained and/or large-scale image classification data sets, introducing new challenges into this traditional research field.

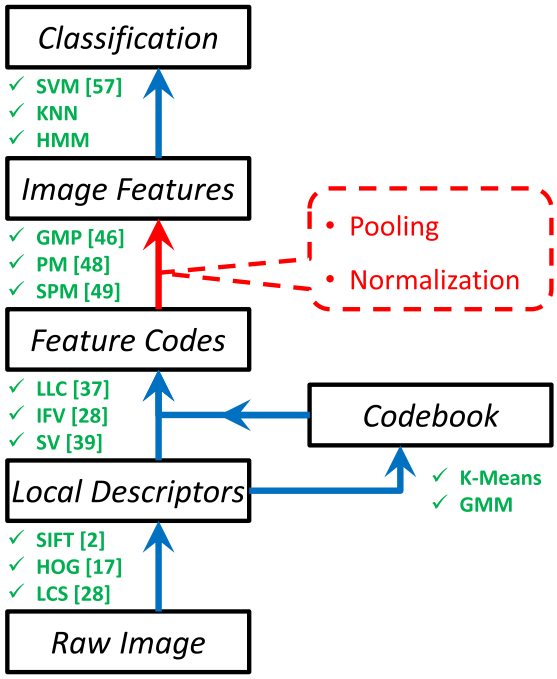


Fig. 1. Basic flowchart of the BoF model. The dashed rectangle indicates the feature summarization stage which would be studied in this paper.

The bag-of-features (BoF) model [1] is one of the most popular algorithms for image representation. It is a statistics-based model aimed at providing a compact image representation. The basic flowchart of the BoF model is shown in Fig. 1. Due to the limited descriptive power of raw pixels, local descriptors such as scale-invariant feature transform (SIFT) [2] are extracted. A visual vocabulary or codebook is then built to capture the data distribution in the feature space. Descriptors are thereafter quantized onto the codebook as compact feature vectors, and the local features are summarized as an image-level feature vector, which is the output of the BoF model. The high-dimensional vector could be used for various purposes, such as image classification [1] and image retrieval [3].

This paper focuses on the feature summarization stage (marked with red arrow in Fig. 1). In essential, feature summarization algorithms are aimed at capturing the invariance of an image. For example, slight spatial translation of objects could be formulated with regional feature pooling, and feature normalization techniques cancel out the impact

Manuscript received December 16, 2014; revised March 23, 2015, May 6, 2015, and May 31, 2015; accepted July 20, 2015. Date of publication July 29, 2015; date of current version July 7, 2016. This work was supported in part by the National Basic Research Program (973 Program) of China under Grant 2013CB329403 and Grant 2012CB316301; in part by the National Natural Science Foundation of China under Grant 61332007, Grant 61273023, Grant 91120011, and Grant 61128007; in part by the Beijing Natural Science Foundation under Grant 4132046; and in part by the Tsinghua University Initiative Scientific Research Program under Grant 20121088071. The work of Q. Tian was supported in part by the Army Research Office under Grant W911NF-12-1-0057, in part by the Faculty Research Awards by NEC Laboratories of America, and in part by the 2012 University of Texas at San Antonio START-R Research Award. This paper was recommended by Associate Editor S. Gong.

L. Xie and B. Zhang are with the State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China, and also with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: 198808xc@gmail.com; dcszb@mail.tsinghua.edu.cn).

Q. Tian is with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: qitian@cs.utsa.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2015.2461978

of different numeric ranges on different feature dimensions. Feature summarization is crucial for improving image classification performance, but is often devalued as a less interesting trick. We defend the importance of feature summarization by proposing two novel algorithms, i.e., generalized regular spatial pooling (GRSP) for spatial pooling and hierarchical feature normalization (HFN) for feature vector normalization. Both algorithms are easy to implement (with only few lines of codes) yet efficient to carry out. Moreover, the algorithms are designed independently of the previous feature encoding stage, and therefore could be freely transplanted onto many classification systems based on the BoF model. In experiments, we observe consistent accuracy gain on every single classification task with the combined (GRSP with HFN) algorithm.

It is highlighted here that we do not aim at designing novel algorithms for feature pooling and normalization. Both techniques (GRSP and HFN) discussed in this paper are very simple, but they do reveal some important principles of generating descriptive, discriminative, and robust image representation. Our contribution mainly lies in that we defend the importance of pooling and normalization, which are indispensable modules but devalued for a long time. We find that both well-designed pooling (GRSP) and normalization (HFN) algorithms consistently boost classification accuracy. Sometimes, the difference between using proper and improper parameters is so significant (e.g., on the Flower-102 dataset, as shown in Table IV, as many as 5% gain is obtained when hierarchical ℓ_2-normalization is applied) that the benefit is larger than designing sophisticated algorithms such as feature encoding. We feel it necessary to record these results and present them in a formal paper.

Preliminary literatures of this paper appeared as [4] and [5]. In this paper, we not only combine the separate modules into a generalized one, but also give insightful discussions on the principles of improving image classification performance.

The remainder of this paper is organized as follows. A detailed overview of the BoF model is outlined in Section II. Then, Sections III and IV present the algorithms for local feature summarization, i.e., GRSP and HFN, respectively. After extensive experimental results are shown in Section V, we draw the conclusion in Section VI.

II. Bag-of-Features Model

Image classification is a challenging task in computer vision [6]–[9]. There are also related topics attracting lots of attentions [10]–[12]. This section provides a detailed overview of the BoF model, one of the most popular pipelines for image classification.

A. Local Descriptor Extraction

The BoF model starts from an image, which is a $W \times H$ matrix $\mathbf{I} = (a_{ij})_{W \times H}$. Here, a_{ij} is the intensity value for a grayscale image, or a 3D vector for an RGB image.

Due to the limited representative power of raw pixels, handcrafted descriptors are often extracted from small patches named interest points on an image. For patch detection, gradient-based operators try to find local maxima that may correspond to well-defined interest points. Typical examples include differential of Gaussian [2], Hessian/Harris affine [13], maximally stable extremal region [14] operators, and dense interest points [15]. In particular, in image classification, it is also suggested to densely extract descriptors from a regular grid on the image [16]. For patch description, popular cases include scale-invariant feature transform (SIFT) [2], and histogram of oriented gradients (HOG) [17]. Other variants, such as gradient location and orientation histogram (GLOH) [18], Speed Up Robust Features (SURF) [19], binary robust independent elementary features (BRIEF) [20], DAISY descriptor [21], and oriented FAST and rotated BRIEF (ORB) [22], are also verified to be efficient and robust in image classification/retrieval tasks. Some minor modifications on local descriptors are also useful [23], [24].

Gradient-based descriptors are sensitive to texture information of an image [25], [26]. Besides texture, additional features such as color and shape could also be extracted for complementariness. A typical idea of capturing color information is to compute texture descriptors from individual color channels of the image. RGB-SIFT, C-SIFT, and Opponent-SIFT are all such cases [27], which differ from each other in the way of calculating color channels. Other kinds of descriptors such as local color statistics (LCS) [28] extract color statistics on local patches, which are verified to well cooperate with grayscale texture patch descriptors. There are also various kinds of shape descriptors, such as Shape Context (SC) [29], Inner Distance Shape Context (IDSC) [30], and Edge-SIFT [26] that extracts SIFT features on the edge responses of original images. Multiple sets of local descriptors could also be fused [25], [26] for richer image description. Recent years, the fast development of deep learning and convolutional neural networks also inspires us to adopt deep conv-net features for image classification [31], [32].

Either combination of patch detection and description algorithms yields a set $\mathcal{D}$ of local descriptors

$$\mathcal{D} = \{(\mathbf{d}_1, \mathbf{l}_1), (\mathbf{d}_2, \mathbf{l}_2), \ldots, (\mathbf{d}_M, \mathbf{l}_M)\} \tag{1}$$

where $\mathbf{d}_M$ and $\mathbf{l}_M$ denote the D-dimensional description vector and the geometric location of the mth descriptor, respectively. M is the total number of dense descriptors. There might be more than one descriptor set for an image in the cases of using multiple local descriptors.

B. Codebook Training

After descriptor extraction, a visual vocabulary, or codebook, is often trained to capture the distribution of the feature space. One of the most popular approaches to data-dependent estimation is to use the kernel density model, which constructs K vectors with D dimensions

$$\mathcal{B} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K\}. \tag{2}$$

The element $\mathbf{c}_K$, $k = 1, 2, \ldots, K$, is named a codeword, and each descriptor is then related to its nearest codeword(s) by the Euclidean distance in the feature space. The K-means clustering algorithm is used to optimize the codebook iteratively.

As accurate K-means is computationally expensive with large codebooks, some accelerating solutions are also suggested, such as hierarchical [3] or approximate [33] versions of K-means. Small codebooks with additional embedded information are also verified to be efficient [34] in real applications.

Other probabilistic models, such as the Gaussian mixture model (GMM), are trained to capture richer geometric contexts in the feature space. It describes the feature space with a mixture of K multivariant Gaussian distributions

$$\mathcal{M} = \{(\pi_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \ldots, (\pi_K, \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K)\}. \tag{3}$$

Parameters π_K, $\boldsymbol{\mu}_K$, and $\boldsymbol{\Sigma}_K$ denote the prior, mean value, and covariance of the kth Gaussian component, respectively, for $k = 1, 2, \ldots, K$. GMM could be solved with Expectation Maximization (EM) iteration.

C. Feature Encoding

The feature encoding stage is aimed at quantizing each of the descriptors into a compact representation.

If the codebook is trained with K-means clustering, i.e., composed of a set of codewords, then a descriptor could be encoded according to its distances to the codewords in the feature space. Hard quantization uses the nearest codeword to quantize a descriptor, resulting in a large quantization error. As an alternative solution, soft quantization allows a descriptor to be reconstructed by a small number of codewords. Sparse coding [35] is a special case of soft quantization, which is verified to be very efficient in image classification [36], [37]. After encoding, each descriptor $\mathbf{d}_m$ is represented as a K-dimensional sparse feature vector $\mathbf{w}_m$, i.e., in which only one or few of the dimensions are nonzero.

If the codebook is trained with a GMM, i.e., geometric context information is preserved, richer discriminative features could be captured by computing the Fisher vectors [28]. It works by decomposing the Fisher information matrix toward maximal discrimination [38]. In this case, both the first- and second-order statistics are encoded, resulting in a much longer ($2DK$ dimensions) and denser (around 50% dimensions are nonzero) feature vector. Consequently, the time and memory costs are much more expensive than using K-means-based encoding. Similar ideas are also used in other high-dimensional features, such as super vector encoding [39] and oriented SIFT/HOG encoding [40].

After the encoding stage, the set of local descriptors is transformed into a set of feature vectors

$$\mathcal{W} = \{(\mathbf{w}_1, \mathbf{l}_1), (\mathbf{w}_2, \mathbf{l}_2), \ldots, (\mathbf{w}_M, \mathbf{l}_M)\} \tag{4}$$

in which $\mathbf{d}_M$ in (1) is replaced by $\mathbf{w}_M$, for $m = 1, 2, \ldots, M$. Besides, there are also learning-based feature encoding [41] algorithms. Various methods have been proposed to encode richer information into image features, including constructing feature groups [42] (visual phrases [26]), assigning weights on visual words [43], or embedding additional information [44].

D. Feature Pooling

After feature encoding, the pooling stage follows to summarize features as a global image representation. This stage is crucial in the BoF model, not only for it summarizes different numbers of local features into a vector of the same length but also for its effect of canceling out the translation variance, allowing an object to appear on different positions of an image.

A natural way of feature pooling is to calculate a global statistics based on all the quantized vectors. Max pooling and average pooling are probably the most widely adopted approaches. The max-pooling strategy calculates the maximal response on each codeword: $\mathbf{f} = \max_{1 \le m \le M} \mathbf{w}_m$, while the average-pooling strategy calculates the average response: $\mathbf{f} = (1/M) \sum_{m=1}^{M} \mathbf{w}_m$. Here, the notations $\max_m$ and $\sum_m$ denote dimension-wise maximization and summation, respectively. Researchers have discussed the choice of max pooling versus average pooling [45], showing that max pooling gives more discriminative representation under soft quantization strategies, while average pooling fits hard quantization better. Generalized max pooling [46] discusses the relationship between max pooling and Fisher vectors.

Both max pooling and average pooling are special cases of ℓ_p-norm pooling, which calculates the ℓ_p-norm: $\mathbf{f} = [\sum_{1 \le m \le M} \mathbf{w}_m^p]^{1/p}$, where $\mathbf{w}_m^p$ denotes the dimension-wise pth power of the feature vector $\mathbf{w}_m$. When $p \to +\infty$ and $p = 1$, ℓ_p-norm pooling degenerates to max pooling and average pooling, respectively. Rather than adjusting norm p manually, the geometric ℓ_p-norm pooling algorithm [47] uses a complex objective function to find out an optimal p for each image individually.

Global pooling algorithms ignore rich spatial information that could be very useful for image understanding. Based on the kernel matching theory, spatial pooling algorithms, such as Pyramid Matching (PM) [48] or Spatial Pyramid Matching (SPM) [49], are proposed, partitioning images into smaller regions for spatial context modeling. Explicitly, let $\mathcal{J} = \{1, 2, \ldots, M\}$ be the index set of the feature set $\mathcal{W}$. The spatial pooling algorithm defines S subsets of $\mathcal{J}$, denoted by $\{\mathcal{J}_1, \mathcal{J}_2, \ldots, \mathcal{J}_S\}$, and summarizes the feature vectors in each subset individually, obtaining S pooled vectors $\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_S\}$. Efforts are also made to improve naive spatial matching methods. To capture more flexible spatial contexts [50]–[52], it is proposed to design a larger set of pooling bins and perform a wise optimization to choose some of them for feature summarization. For fine-grained recognition, it is also suggested to design the pooling bins according to the semantic parts of the objects [53]–[55].

The output of the pooling stage is a set of S individual vectors $\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_S\}$. These vectors can be of equal [49] or unequal [56] lengths. Most often, they are concatenated as an image-level vector $\mathbf{F}$.

In Section III, we will study feature pooling in depth.

E. Feature Normalization

Feature normalization, or feature scaling, is a crucial data preprocessing stage aimed at avoiding attributes in greater numeric ranges dominating those in smaller numeric ranges and controlling the similarity measure between feature vectors.

One of the most popular feature normalization methods is the ℓ_p-norm normalization, in which we divide each feature

vector by its length in the ℓ_p space: $\widetilde{\mathbf{F}} = \mathbf{F}/\|\mathbf{F}\|_p$, so that all the vectors become ℓ_p unit length. The selection of the norm p might significantly impact the performance of generalized classifiers. For instance, it is demonstrated that in support vector machine (SVM), ℓ_2 normalized vectors have the minimal structural risk [57].

Besides the naive normalization method, researchers have proposed various techniques to fit different machine learning models, such as SVM [58]–[60], Naive Bayes classifier [61], hidden Markov model estimation [62], kernel Fisher discriminant analysis [63], and even the inverted index structure [64]. It is also important to consider the proper order of feature normalization and concatenation, which might heavily impact the discriminative power of feature vectors, especially in the scenarios of part-based classification [53].

We will provide a detailed discussion on feature normalization in Section IV.

F. Classification

The output of the BoF model is usually a set of very long feature vectors. Since the number of training samples is relatively smaller, the SVM is often adopted to avoid overfitting.

Recent years, as the number of image categories grows from hundreds to tens of thousands [65], scalability has become more and more important for practical classification systems. In general, training one-versus-one classifiers is much more expensive than training one-versus-rest classifiers, and therefore the latter strategy is often adopted when the number of categories is large [66]. Besides the flat classifiers, hierarchical techniques [67], [68] have also been proposed for training large-scale classifiers, but the obtained accuracy is often lower than that of flat classifiers [66]. Moreover, image-to-class distance [69] is often more robust in classification.

It is also crucial to select a proper kernel function for SVM training. Although nonlinear kernels such as the χ^2 [70] or Hellinger's kernel [28] often produce higher accuracy for linear nonseparable cases, the linear kernel is verified to be more efficient in training large-scale classifiers [71].

The detailed discussion of classifiers goes out of the goal of this paper.

III. Generalized Regular Spatial Pooling

In this section, we present the GRSP algorithm. It is considered as a generalization of SPM [49].

In essential, spatial pooling algorithms are aimed at constructing a group of index subsets $\{\mathcal{J}_1, \mathcal{J}_2, \ldots, \mathcal{J}_S\}$ of the full index set $\mathcal{J} = 1, 2, \ldots, M$, and then performing a specific pooling method on each subset. Following this basic rule, we define the subsets in both SPM and GRSP.

A. Spatial Pyramid Matching

SPM defines the number of layers L for spatial matching and partitions the image plane into equal-sized regular grids in each layer. Mathematically, let $\mathcal{P}$ be the set of pixels in image $\mathbf{I}$: $\mathcal{P} = \{\mathbf{l} = (x, y) \mid 0 < x \leqslant W, 0 < y \leqslant H\}$. At the lth layer, $l = 0, 1, \ldots, L-1$, the image is partitioned into $A_l \times B_l$ pooling bins. Therefore, the size of each bin at the lth layer is $(W/A_l) \times (H/B_l)$, and the (a, b)th bin, $a = 0, 1, \ldots, A_l - 1$, $b = 0, 1, \ldots, B_l - 1$, is defined as

$$\mathcal{P}_{l,a,b} = \left\{ \mathbf{l} = (x, y) \middle| \frac{aW}{A_l} < x \leqslant \frac{(a+1)W}{A_l} \wedge \frac{bW}{B_l} < y \leqslant \frac{(b+1)H}{B_l} \right\}. \quad (5)$$

In the original SPM model [49], we have $A_l = B_l = 2^l$ for $l = 0, 1, \ldots, L-1$, implying that the image is divided into 2×2 subregions at the first layer, and then each of the subregion is recursively divided into 2×2 smaller subregions at the next layer. Recently, it has been verified that horizontal stripes are also efficient pooling bins [50], especially in the case of using Fisher vectors [66], [72], [73].

We define the index sets directly using the pooling bins

$$\mathcal{J}_{l,a,b} = \{m \mid 1 \leqslant m \leqslant M \wedge \mathbf{l}_m \in \mathcal{P}_{l,a,b}\}. \quad (6)$$

The number of index sets equals to the number of pooling bins, i.e., $S = \sum_{l=0}^{L-1} A_l \times B_l$, in an L-layer model.

B. Generalized Regular Spatial Pooling

As a generalization to SPM, GRSP also uses rectangular pooling bins, but allows changing the number of the bins in each layer for either denser or sparser spatial context modeling.

Let us still assume that the size of bins at the lth layer is $(W/A_l) \times (H/B_l)$, i.e., using the same setting as in the SPM model. Then, we define another sequence $(A'_0, B'_0), (A'_1, B'_1), \ldots, (A'_{L-1}, B'_{L-1})$, which means that there are $A'_l \times B'_l$ equal-sized pooling bins in the lth layer. We then place a $(W/A_l) \times (H/B_l)$ rectangle at the top-left corner of the image and move the bin along both x and y axes of the image, from top-left to bottom-right corner, making sure that the spatial strides at each move, either horizontal or vertical, are equal. Mathematically, it is easy to derive that the spatial strides in the lth layer are $s_{l,x} = (W - (W/A_l)/(A'_l - 1)$ and $s_{l,y} = (H - (H/B_l)/B'_l - 1)$, respectively. Following (5), the (a, b)th bin at this layer, $a = 0, 1, \ldots, A'_l - 1$ and $b = 0, 1, \ldots, B'_l - 1$, is defined as

$$\mathcal{P}_{l,a,b} = \left\{ \mathbf{l} = (x, y) \mid a \times s_{l,x} < x \leqslant a \times s_{l,x} + \frac{W}{A_l} \wedge b \times s_{l,y} < y \leqslant b \times s_{l,y} + \frac{H}{B_l} \right\}. \quad (7)$$

Obviously, if we have $A'_l = A_l$ and $B'_l = B_l$ for any l, $l = 1, 2, \ldots, L-1$, GRSP degenerates to SPM. Otherwise, the pooling bins at the lth layer might become either denser ($A'_l > A_l$ and $B'_l > B_l$) or sparser ($A'_l < A_l$ and $B'_l < B_l$). Fig. 2 illustrates the denser spatial pooling on the original 2×2 layer: $A'_1 = B'_1 = 3$, and Fig. 3 shows the sparser spatial pooling on the original 4×4 layer: $A'_2 = B'_2 = 3$. When a denser pooling is performed at one layer, some local features could be covered by more than one bin, while a sparser pooling might ignore a fraction of local features (not covered by any of the bins). One might certainly adopt a larger pooling bin to fill up the whole image in the latter case; for simplicity, we just use a straightforward solution to preserve the same bin size as in SPM.

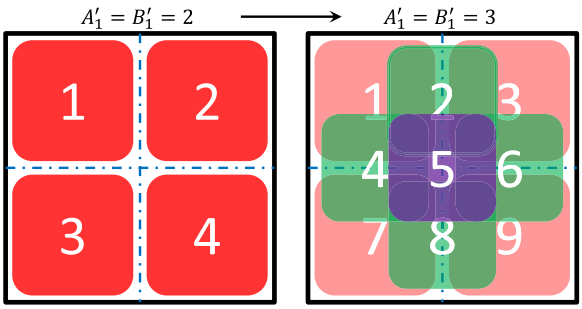


Fig. 2. Example of original (left) and denser (right) spatial pooling in the first layer ($A_1 = B_1 = 2$ and $A'_1 = B'_1 = 3$; please note that the definition starts with the zeroth layer). Each pooling bin shares half of its pixels with its neighboring bins.

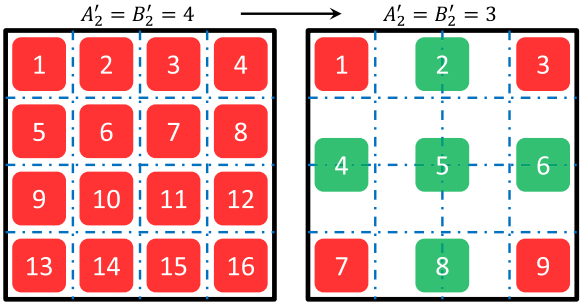


Fig. 3. Example of original (left) and sparser (right) spatial pooling in the second layer ($A_2 = B_2 = 4$ and $A'_2 = B'_2 = 3$; please note that the definition starts with the zeroth layer). Some regions on the image plane are not occupied by any of the pooling bins.

The definition of index sets in GRSP simply follows (6). The number of index sets is $S = \sum_{l=0}^{L-1} A'_l \times B'_l$.

C. Comparison With Previous Works

There are many works aimed at providing more reasonable ways of spatial pooling beyond SPM [49]. Liu *et al.* [56] propose computing smaller codebooks for feature encoding in the lower levels (smaller bins), while [36] suggests combining sparse coding algorithms with spatial pyramids toward better image representation. When the encoded feature vectors are very long and dense, such as in the case of Fisher vector encoding [28], [73], it is suggested to use a smaller number of pooling bins to reduce the time and memory complexity [72].

Maybe the most relevant works to our algorithm are [50] and [51]. In [50], a larger number of receptive field candidates are extracted on the image plane, and a classifier is trained with structured sparsity to use only a subset of all the features. In [51], a hierarchical ROI dictionary is trained for spatial pooling, and partial least-square analysis is employed to learn a compact image representation. These methods often produce larger improvement on object recognition than on scene recognition [51], for the reason that scene images are somewhat regular, thus naive pooling strategy works very well. For example, the accuracy gain on the Caltech101 data set by [50] and [51] are 1.9% and 3.1%, respectively, but the gain on the Scene-15 data set by [51] is merely 1.1%. Our simple solution gives ~1% gain on both data sets (see Section V-B). In comparison, the proposed GRSP algorithm is extremely simple and generalizable: one needs only few lines of codes to implement the algorithm, and it produces consistent improvement in a wide range of classification tasks.

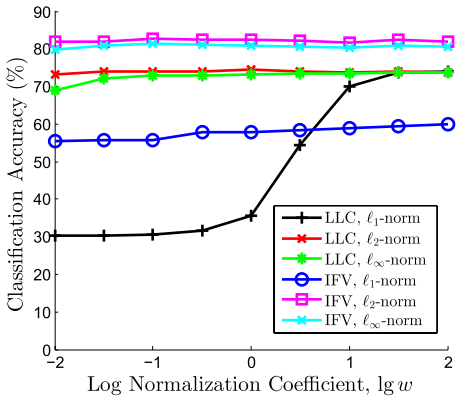


Fig. 4. Classification results based on LLC [37] and IFV [28] on the Caltech101 data set. Different normalization parameters (power p and coefficient w) are used. w is a dominant factor in LLC encoding with ℓ_1-normalization.

IV. Hierarchical Feature Normalization

In this section, we present the HFN algorithm. It is previously described in [4] as an optimization of conventional ℓ_p-norm normalization. We first show that tuning parameters in normalization help to achieve good performance, and then generalize from global normalization to weighted part-wise normalization based on two assumptions, i.e., equal/hierarchical contribution assumption.

A. Conventional Normalization: Power and Coefficient

One of the simplest and most widely adopted normalization techniques is the ℓ_p-normalization, in which a feature vector is projected onto the ℓ_p-norm unit hypersphere

$$\widetilde{\mathbf{F}} = \frac{\mathbf{F}}{\|\mathbf{F}\|_p} \tag{8}$$

where $\|\mathbf{F}\|_p$ is the ℓ_p-norm: $\|\mathbf{F}\|_p = (\sum_i F_i^p)^{1/p}$ and p is named the normalization power. When $p \to +\infty$, $\|\mathbf{F}\|_p = \max_i F_i$. In most cases, the normalized feature vectors are fed into SVM, which is quite sensitive to the numerical ranges of input data [74], and therefore it is reasonable to choose a proper normalization coefficient w and modify (8) as

$$\widetilde{\mathbf{F}} = w \times \frac{\mathbf{F}}{\|\mathbf{F}\|_p}. \tag{9}$$

We observe the impact of normalization power and coefficient on the Caltech101 data set [75]. Detailed experiment settings could be found in Section V-A. The classification results using different combinations of power p and coefficient w are shown in Fig. 4. One can observe the importance of choosing proper normalization parameters. Especially, when $p = 1$, i.e., ℓ_1-norm normalization is adopted, it is instructive to use a larger w to prevent the SVM classifier from being disturbed by small feature values. When a relatively larger w is used, there are less numerical stability issues observed, and consequently, ℓ_1-norm produces comparable performance than that of ℓ_2-norm in the case of LLC encoding. This provides a different opinion from that ℓ_1-normalization would cause classification accuracy drop dramatically [37]. Although the performance of ℓ_1-norm with improved Fisher vector (IFV) encoding is always much worse

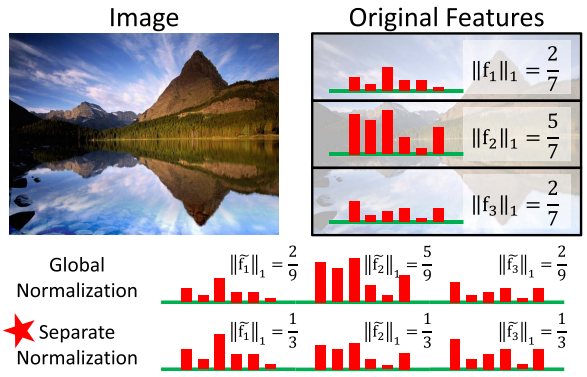


Fig. 5. Global normalization versus SFN. We use ℓ_1-norm (sum norm) as an example. Different spatial regions are set with equal weights to preserve nearly the same amount of information.

than that of ℓ_2-norm, adopting a larger coefficient w still helps to boost classification accuracy. Therefore, in the later experiments, we fix $w = 100$ in every single case.

B. Separate and Hierarchical Feature Normalization

The above normalization method simply considers each feature vector as a whole. However, a feature vector generated by the BoF model usually contains several parts. For example, if there are S spatial pooling bins, the image-level feature vector comprises of S originally individual regional feature vectors. It is not instructive to ignore its intrinsic structure.

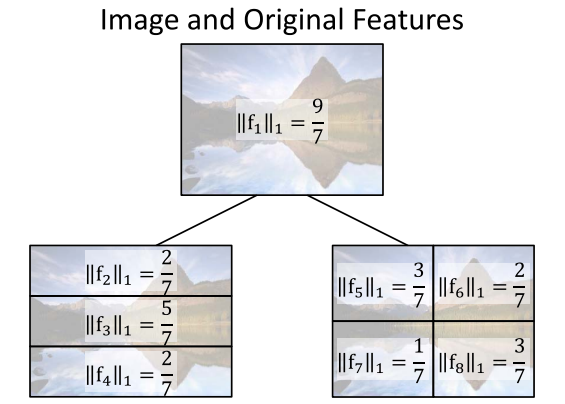
A straightforward modification of the global pooling formula (9) starts from the equal contribution assumption, i.e., each part of $\mathbf{F}$ equally contributes to recognition. Denote $\mathbf{F} = [\mathbf{f}_1; \mathbf{f}_2; \ldots; \mathbf{f}_S]$, where $\mathbf{f}_s$ is the feature vector of the sth pooling bin, for $s = 1, 2, \ldots, S$. Instead of normalizing $\mathbf{F}$ in a global manner, we perform a separate normalization technique, which normalizes $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_S$ individually

$$\widetilde{\mathbf{f}}_s = \frac{w}{S^{1/p}} \times \frac{\mathbf{f}_s}{\|\mathbf{f}_s\|_p}, \quad s = 1, 2, \ldots S \tag{10}$$

and concatenates $\widetilde{\mathbf{f}}_1, \widetilde{\mathbf{f}}_2, \ldots, \widetilde{\mathbf{f}}_S$ as $\widetilde{\mathbf{F}}$: $\widetilde{\mathbf{F}} = [\widetilde{\mathbf{f}}_1; \widetilde{\mathbf{f}}_2; \ldots; \widetilde{\mathbf{f}}_S]$. The modified normalization coefficient $w/S^{1/p}$ in (10) confirms $\|\widetilde{\mathbf{F}}\|_p = w$ as it is in (9). Equation (10) defines the separate feature normalization (SFN) algorithm.

SFN is illustrated with a real part-based model in Fig. 5. In this case, the scene image is partitioned into several regions (pooling bins) with various saliencies on the image, and therefore pooled feature vectors in different parts might have different lengths in the feature space. Under the assumption that these parts equally contribute to image classification, we shall normalize the feature vectors separately in order to prevent small parts being dominated by the large ones. It is also worth noting that SFN is the default strategy used in the IFV encoding [28], in which it is claimed that SFN helps to provide more discriminative feature vectors.

However, it is not always true that each part contributes equally. Most often, larger pooling bins consist of more basic regions and are consequently more robust and discriminative. Therefore, we slightly modify the equal contribution assumption into the hierarchical contribution assumption, i.e., the contribution of a pooling bin is proportional to its



	$\|f_1\|_1$	$\|f_2\|_1$	$\|f_3\|_1$	$\|f_4\|_1$	$\|f_5\|_1$	$\|f_6\|_1$	$\|f_7\|_1$	$\|f_8\|_1$
Global Normalization	$\frac{1}{3}$	$\frac{2}{27}$	$\frac{5}{27}$	$\frac{2}{27}$	$\frac{1}{9}$	$\frac{2}{27}$	$\frac{1}{27}$	$\frac{1}{9}$
Separate Normalization	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$
Hierarchical Normalization	$\frac{1}{3}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{12}$
Number of Contained Pixels	$k_1 : k_{2,3,4} : k_{5,6,7,8} = 1 : \frac{1}{3} : \frac{1}{4}$							

Fig. 6. Comparison among global normalization, SFN, and HFN. We use ℓ_1-norm (sum norm) as an example. HFN provides more reasonable results for a multilayer spatial pooling structure.

area (number of pixels) on the image plane. Let us denote k_s by the number of pixels contained in sth pooling bin. Appending the fixed constraint $\|\mathbf{F}\|_p = w$ obtains

$$\begin{cases} w_s^p \propto k_s \\ \sum_{s=1}^{S} w_s^p = w^p. \end{cases} \tag{11}$$

Solving (11) gives a group of new coefficients for part-wise normalization, in which we enhance the spatial weights on larger pooling bins to emphasize global information. We name (11) the HFN algorithm.

HFN is illustrated in Fig. 6 using a three-layer spatial pooling model with $\{1 \times 1, 2 \times 2, 1 \times 3\}$ pooling bins. It is also convenient to add more layers of pooling bins (e.g., 4×4) into the model. With hierarchical normalization, features generated by smaller regions are assigned lower weights, implying that features extracted on smaller regions are more likely to be unstable and less trustworthy.

C. Comparison With Previous Works

Although the feature normalization stage is often considered a less interesting issue, previous literatures have verified that classification accuracy is greatly impacted by different normalization strategies. In [37], it is claimed that ℓ_1-normalization results in dramatic accuracy drop compared with ℓ_2-normalization. We make a strong defense for ℓ_1-norm by noting that it works very well with a large normalization coefficient. In [28], the Fisher vector is improved with power normalization, which implicitly introduces the Hellinger's kernel to calculate the distance in the feature space. Reference [72] also suggests to choose the feature normalization strategy carefully so as to achieve higher classification accuracy.

Our work is also related to several feature pooling algorithms that do not discuss the feature normalization strategies explicitly. In [56], it is suggested to use smaller codebooks to quantize the local descriptors on smaller pooling bins, resulting in extracting lower dimensional feature vectors on the smaller pooling bins. In this paper, we preserve the same feature length for large and small pooling bins, but adopt a hierarchically decayed weighting scheme for similar effects.

V. Experiments

A. Datasets and Settings

We test our approach on three types of image classification tasks, i.e., scene recognition, generic object recognition, and fine-grained object recognition.

For scene recognition, the following data sets are used.

1) The UIUC Sport-8 data set [76] contains 8 sporting scenes and 1579 images. A total of 70 images per category are randomly selected for training.
2) The Scene-15 data set [49] contains 15 scenes and 4485 images. A total of 100 images per category are randomly selected for training. It is one of the most widely used data sets for scene understanding tasks.
3) The LandUse-21 data set [77] contains 21 land-use scenes with 100 images for each class. A total of 80 images per category are randomly selected for training.
4) The MIT Indoor-67 data set [78] contains 67 indoor scenes and 15 620 images. A total of 80 images per category are randomly selected for training.
5) The SUN-397 data set [79] contains 397 indoor/outdoor scenes and 108 792 images. A total of 50 images per category are randomly selected for training.

For generic object recognition, the following image corpora are evaluated.

1) The Bird-6 data set [80] contains 6 *bird* categories and 100 images per class. A total of 50 images per category are randomly selected for training.
2) The Butterfly-7 data set [81] contains 619 *butterfly* images from 7 species. A total of 26 images per category are randomly selected for training.
3) The Flower-17 data set [82] contains 17 *flower* categories with 80 images per class. A total of 60 images per category are randomly selected for training.
4) The Caltech101 data set [75] contains 9144 images of 102 classes. A total of 30 images per category are randomly selected for training.
5) The Caltech256 data set [83] contains 30 607 images of 257 classes. A total of 60 images per category are randomly selected for training.
6) The PascalVOC-2007 data set [84] is one of the most popular and challenging cases for multilabel concept learning and object retrieval. This data set, with around 10 000 images, provides a fixed training/testing split. Performance is evaluated by the mean average precision score of each query image.

Although bird and flower data sets also appear in the fine-grained data sets, we emphasize that generic object recognition often deals with very small numbers of fine-grained concepts. There could be a large difference when classifying increasing numbers of fine-grained concepts, such as the following fine-grained object recognition cases.

1) The Oxford Pet-37 data set [85] contains 37 *cat* or *dog* breeds and 7349 images. A total of 100 images per category are randomly selected for training.
2) The Aircraft-100 data set [86] contains 100 *aircraft* models and 100 images for each model. A total of 67 images per category are randomly selected for training.
3) The Oxford Flower-102 data set [87] contains 8189 *flower* images from 102 categories. A total of 20 images per category are randomly selected for training.
4) The Stanford Dog-120 data set [88] contains 20 580 *dog* images of 120 breeds. A total of 100 images per category are randomly selected for training.
5) The Caltech-UCSD Bird-200-2011 data set [89] contains 11 788 *bird* images of 200 different species. A total of 30 images per category are randomly selected for training.

The basic experimental settings used in the later sections follow the recent proposed BoF models, i.e., based on weak [locality-constrained linear coding (LLC) [37]] and strong IFVs [28] feature encoding algorithms.

1) *Image Rescale:* Images are scaled, with the aspect ratios preserved, so that the larger axis is 300 pixels for LLC and 600 pixels for IFV. When a bounding box is available, we use only the region within the box.
2) *Local Descriptors:* We use the VLFeat [90] library to extract dense RootSIFT [24] descriptors. The spatial stride and window size of dense sampling are 6 and 12 for LLC, while 8 and 16 for IFV, respectively. On the same patches, 96D LCS descriptors [28] are also extracted. The dimensions of both descriptors are reduced to 64 using PCA in the case of IFV encoding.
3) *Codebook Training:* We then cluster the descriptors with K-means clustering ($K = 2048$) and GMM ($K = 256$), respectively, for the LLC [37] and IFV [28] encoding methods. The number of descriptors collected for clustering does not exceed two million.
4) *Feature Encoding and Pooling:* We use LLC and IFV algorithms for feature encoding. After LLC encoding, we use max pooling with a $\{1 \times 1, 2 \times 2, 4 \times 4\}$ spatial pyramid, while after IFV encoding, we use sum pooling with a $\{1 \times 1, 2 \times 2, 1 \times 3\}$ spatial pyramid.
5) *Classification:* We use SGD [91], a scalable SVM implementation for training and testing. The training and testing split follows the original setting of each data set. The average accuracy over all the tested categories is calculated. We repeat the random selection 10 times and report the averaged results.

B. Generalized Regular Spatial Pooling

We observe the impact of using different numbers of pooling bins in the GRSP algorithm. Here, we fix the bin size in the lth layer as $(W/2^l) \times (H/2^l)$. The classification results on the

TABLE I

CLASSIFICATION RESULTS OF DIFFERENT POOLING PARAMETERS ON FOUR DATA SETS WITH FEWER TRAINING SAMPLES

Case No.	Encoding	$A'_l \times B'_l$ 0th	1st	2nd	Feature Dims	Bird-6 Acc. (%)	Butterfly-7 Acc. (%)	Sport-8 Acc. (%)	Scene-15 Acc. (%)
1	LLC	1×1	2×2	3×3	28K	89.97	86.87	88.13	83.09
2	LLC	1×1	2×2	4×4	42K	90.10	87.23	88.25	83.44
3	LLC	1×1	2×2	6×6	82K	90.13	87.43	88.22	83.59
4	LLC	1×1	3×3	3×3	38K	90.67	87.71	88.75	83.73
5	LLC	1×1	3×3	4×4	52K	90.73	87.89	88.78	83.77
6	LLC	1×1	3×3	6×6	92K	90.67	87.91	88.62	83.80
7	LLC	1×1	4×4	3×3	52K	90.27	87.62	88.48	83.67
8	LLC	1×1	4×4	4×4	66K	90.47	87.77	88.44	83.76
9	LLC	1×1	4×4	6×6	106K	90.03	87.51	87.92	83.68
10	IFV	1×1	2×2	1×2	224K	93.67	91.05	91.56	88.96
11	IFV	1×1	2×2	1×3	256K	93.90	91.47	91.89	89.70
12	IFV	1×1	2×2	1×4	288K	94.07	91.58	92.04	90.21
13	IFV	1×1	3×3	1×2	384K	94.17	91.98	92.25	89.94
14	IFV	1×1	3×3	1×3	416K	94.27	92.20	92.38	90.24
15	IFV	1×1	3×3	1×4	448K	94.30	92.22	92.44	90.35
16	IFV	1×1	4×4	1×2	608K	94.07	91.72	92.05	89.97
17	IFV	1×1	4×4	1×3	640K	94.03	91.87	92.08	90.15
18	IFV	1×1	4×4	1×4	672K	93.83	91.56	91.98	90.21

TABLE II

CLASSIFICATION RESULTS OF DIFFERENT POOLING PARAMETERS ON FOUR DATA SETS WITH MORE TRAINING SAMPLES

Case No.	Encoding	$A'_l \times B'_l$ 0th	1st	2nd	Feature Dims	SUN-397 Acc. (%)	Caltech256 Acc. (%)	Bird-200 Acc. (%)	Dog-120 Acc. (%)
19	LLC	1×1	2×2	3×3	28K	37.98	48.78	34.61	29.29
20	LLC	1×1	2×2	4×4	42K	38.79	49.52	35.57	30.13
21	LLC	1×1	2×2	6×6	82K	39.56	50.07	36.37	30.87
22	LLC	1×1	3×3	3×3	38K	39.96	49.98	35.91	30.35
23	LLC	1×1	3×3	4×4	52K	40.43	50.47	36.76	31.07
24	LLC	1×1	3×3	6×6	92K	40.56	50.81	37.19	31.53
25	LLC	1×1	4×4	3×3	52K	40.23	50.25	36.77	31.18
26	LLC	1×1	4×4	4×4	66K	40.58	50.89	37.31	31.65
27	LLC	1×1	4×4	6×6	106K	40.65	51.13	37.44	31.89
28	IFV	1×1	2×2	1×2	224K	47.13	58.01	45.56	40.24
29	IFV	1×1	2×2	1×3	256K	48.35	58.77	46.61	41.12
30	IFV	1×1	2×2	1×4	288K	49.10	59.23	47.34	41.78
31	IFV	1×1	3×3	1×2	384K	48.76	59.03	46.97	41.27
32	IFV	1×1	3×3	1×3	416K	49.37	59.67	47.61	41.87
33	IFV	1×1	3×3	1×4	448K	49.79	59.87	47.98	42.06
34	IFV	1×1	4×4	1×2	608K	49.23	59.81	47.77	41.75
35	IFV	1×1	4×4	1×3	640K	49.89	60.21	48.16	42.32
36	IFV	1×1	4×4	1×4	672K	50.21	60.33	48.25	42.41

data sets with smaller and larger numbers of training samples are summarized in Tables I and II, respectively.

First, we compare the classification results on the data sets with fewer training samples (see Table I). With LLC encoding [37] (cases 1–9), one can observe significant accuracy gain as the number of pooling bins increases from 2×2 to 3×3 on the first layer, [see case pairs $(1, 4)$, $(2, 5)$, and $(3, 6)$]. However, when the number is further increased from 3×3 to 4×4, one can observe only a limited accuracy gain or even accuracy drop [see case pairs $(4, 7)$, $(5, 8)$, and $(6, 9)$]. This suggests that denser spatial pooling bins do provide extra information into image representation, but using too many bins could also introduce considerable redundance, which actually harms the classification accuracy. On the second layer, things become different: the best classification accuracy is obtained with 3×3 pooling bins, and increasing the number to 4×4 or 6×6 causes slight accuracy drop [see case groups $(1, 2, 3)$, $(4, 5, 6)$, and $(7, 8, 9)$]. A similar discipline is also summarized from the results using IFV encoding [28] (cases 10–18). When the originally used 2×2 grid is replaced by a 3×3 grid, the classification accuracy is improved significantly, whereas the even denser 4×4 grid does not help much to provide complementary information in image representation. When using horizontal stripes for spatial pooling, the best choice is to use original 1×3 bins.

However, quite different results are observed when there are more training samples for classification (see Table II). With LLC encoding [37] (cases 19–27), the best classification accuracy is obtained with the largest number of pooling bins, i.e., 4×4 on the first layer and 6×6 on the second layer. With IFV encoding, it is also instructive to introduce more pooling bins to improve the representative power of features, and the best performance is obtained with 3×3 and 1×4 bins on the first and second layers, respectively. We have also tested the algorithm with even larger numbers of pooling bins, e.g., 5×5 on the first layer and 1×8 on the second and third layers

TABLE III
CLASSIFICATION RESULTS OF DIFFERENT FEATURE NORMALIZATION STRATEGIES ON SCENE RECOGNITION DATA SETS

Case No.	Encoding	Normalization Algorithm	Sport-8 Acc. (%)	Scene-15 Acc. (%)	LandUse-21 Acc. (%)	Indoor-67 Acc. (%)	SUN-397 Acc. (%)
1	LLC	No Normalization	86.51	81.63	87.50	45.38	37.20
2	LLC	Global-ℓ_1	87.67	82.80	88.50	46.34	38.23
3	LLC	Global-ℓ_2	88.25	83.44	89.17	46.91	38.79
4	LLC	Global-ℓ_∞	87.11	82.17	88.07	45.84	38.05
5	LLC	Separate-ℓ_1	86.12	81.19	87.10	44.83	36.39
6	LLC	Separate-ℓ_2	87.69	83.02	88.81	46.10	38.55
7	LLC	Separate-ℓ_∞	87.03	82.19	87.98	45.92	37.70
8	LLC	Hierarchical-ℓ_1	86.81	81.61	87.86	45.04	36.71
9	LLC	Hierarchical-ℓ_2	88.35	83.65	89.52	47.45	39.46
10	LLC	Hierarchical-ℓ_∞	87.56	82.39	88.31	46.08	38.61
11	IFV	No Normalization	90.48	88.18	92.43	60.31	46.88
12	IFV	Global-ℓ_1	73.91	70.17	76.86	45.01	33.18
13	IFV	Global-ℓ_2	91.02	88.91	92.52	60.38	47.06
14	IFV	Global-ℓ_∞	90.42	88.48	92.12	59.57	46.11
15	IFV	Separate-ℓ_1	61.27	60.21	66.19	34.28	27.69
16	IFV	Separate-ℓ_2	91.89	89.70	93.64	61.87	48.35
17	IFV	Separate-ℓ_∞	90.79	88.93	92.67	60.69	47.21
18	IFV	Hierarchical-ℓ_1	65.14	63.49	70.24	38.51	28.12
19	IFV	Hierarchical-ℓ_2	92.01	90.14	94.07	62.41	48.81
20	IFV	Hierarchical-ℓ_∞	88.49	86.88	90.21	57.80	45.71

for IFV, respectively. The accuracy improvement is relatively smaller.

As assistant experiments, we also perform cross validation using training samples only. On each data set, the training subset is equally partitioned into five equal parts. Every time, four of them are used for training and the remaining one is left for testing. In these experiments, we observe the same results, i.e., increasing the number of pooling bins works better on larger training sets, which implies that the above parameter selection process could be performed automatically.

The different disciplines observed in the cases with smaller and larger numbers of training samples could be explained with overfitting. In the case of fewer training samples, classification models in a high-dimensional feature space may not be well trained. As the number of training samples grows, machine learning algorithms become more confident in fitting classification models into a high-dimensional space.

In conclusion, we will use different settings of pooling bins according to the number of training samples in the data sets. For those data sets with less or equal than 1000 training samples, we use $\{1 \times 1, 3 \times 3, 4 \times 4\}$ pooling bins for LLC encoding and $\{1 \times 1, 3 \times 3, 1 \times 3\}$ pooling bins for IFV encoding. For those data sets with more than 1000 training samples, we use $\{1 \times 1, 4 \times 4, 6 \times 6\}$ pooling bins for LLC encoding and $\{1 \times 1, 4 \times 4, 1 \times 4\}$ pooling bins for IFV encoding. Although it is not perfect to discriminate different data sets merely using the number of training samples, our model provides a simple solution toward extracting features with varying descriptive power in different cases.

In the cases of smaller data sets such as those presented in Table I, the accuracy gain is relatively small, i.e., most often less than 1% beyond standard SPM. However, the gain could be as large as 2% in larger data sets (SUN-397 and Caltech256 in Table II). Moreover, we point out that GRSP provides consistent accuracy gain, which verifies our motivation, i.e., a properly constructed spatial pooling set helps image classification, which is just the goal of this paper.

The time and memory complexity of SVM classification is linear to the total number of pooling bins S, which is the same as previous algorithms [50], [51]. Thanks to the simplicity of GRSP, computational costs on the feature pooling stage are almost ignorable.

C. Hierarchical Feature Normalization

We evaluate different feature normalization models, i.e., global, separate, and hierarchical normalization, on a wide range of image data sets for scene recognition and generic/fine-grained object recognition. We choose three most widely adopted norms, i.e., ℓ_1-norm, ℓ_2-norm, and ℓ_∞-norm (max norm), and fix the normalization coefficient $w = 100$. The classification results are summarized in Tables III and IV, respectively.

One can observe that, in most cases, classification accuracy is improved by adopting the SFN and HFN algorithms. This indicates the benefit from normalizing the feature vectors according to their intrinsic structure. Moreover, HFN always works better than SFN, validating that larger pooling bins indeed provide more trustworthy information.

It is also instructive to observe the difference among scene recognition, generic object recognition, and fine-grained object recognition. In general, the separate/hierarchical contribution assumption would be better satisfied when the image is partitioned into semantic parts or regions in spatial context modeling. In the case of fine-grained recognition, pose variation between different samples is relatively small, and therefore the accuracy gain using HFN is significantly larger than in the cases of scene or generic object recognition. In [4], we also perform HFN on a part-based classification model on the Bird-200 data set [89] and observe even more significant accuracy gain (relatively more than 10%) over the

TABLE IV
CLASSIFICATION RESULTS OF DIFFERENT FEATURE NORMALIZATION STRATEGIES ON OBJECT RECOGNITION DATA SET

Case No.	Encoding	Normalization Algorithm	Caltech101 Acc. (%)	Caltech256 Acc. (%)	Flower-102 Acc. (%)	Dog-120 Acc. (%)	Bird-200 Acc. (%)
1	LLC	No Normalization	75.17	48.26	71.78	28.71	35.57
2	LLC	Global-ℓ_1	76.06	48.96	72.58	29.81	35.07
3	LLC	Global-ℓ_2	76.61	49.52	73.05	30.13	35.57
4	LLC	Global-ℓ_∞	75.49	48.59	72.16	29.35	34.39
5	LLC	Separate-ℓ_1	74.27	47.00	70.51	27.35	33.32
6	LLC	Separate-ℓ_2	76.25	49.27	73.21	31.35	35.89
7	LLC	Separate-ℓ_∞	75.46	48.23	72.97	30.20	34.67
8	LLC	Hierarchical-ℓ_1	74.93	47.47	71.35	28.90	34.07
9	LLC	Hierarchical-ℓ_2	76.98	49.90	74.38	32.36	37.01
10	LLC	Hierarchical-ℓ_∞	75.67	48.57	72.88	30.11	34.81
11	IFV	No Normalization	78.43	55.65	78.74	38.29	43.39
12	IFV	Global-ℓ_1	67.18	46.98	66.93	33.91	35.92
13	IFV	Global-ℓ_2	80.73	57.71	81.31	40.24	45.75
14	IFV	Global-ℓ_∞	79.59	56.42	80.75	39.39	44.58
15	IFV	Separate-ℓ_1	55.23	35.01	54.37	27.34	30.01
16	IFV	Separate-ℓ_2	81.07	58.77	82.43	41.12	46.61
17	IFV	Separate-ℓ_∞	80.31	57.09	81.72	40.62	45.83
18	IFV	Hierarchical-ℓ_1	63.12	39.42	60.46	30.53	31.36
19	IFV	Hierarchical-ℓ_2	81.52	59.37	83.71	42.51	47.98
20	IFV	Hierarchical-ℓ_∞	78.36	55.41	80.05	39.19	44.52

TABLE V
COMPARISON OF OUR CLASSIFICATION RESULTS WITH PREVIOUS WORKS ON SCENE RECOGNITION

Algorithm	UIUC Sport-8	Scene-15	LandUse-21	MIT Indoor-67	SUN-397
Lazebnik *et.al.* [49]	–	81.4	–	–	–
Li *et.al.* [76]	73.4	–	–	–	–
Quattoni *et.al.* [78]	–	–	–	26.1	–
Yang *et.al.* [36]	–	80.4	–	–	–
Boureau *et.al.* [92]	–	84.3	–	–	–
Xiao *et.al.* [79]	–	88.1	–	–	38.0
Yang *et.al.* [77]	–	–	81.19	–	–
Xie *et.al.* [26]	88.17	83.77	–	46.38	–
Kobayashi *et.al.* [40]	90.42	85.63	–	58.91	–
Wang *et.al.* [93]	91.0	88.7	–	–	–
Lin *et.al.* [94]	92.08	91.06	–	68.50	–
Xie *et.al.* [95]	–	–	–	63.48	45.91
Ours (LLC [37])	88.25 ± 0.82	83.44 ± 0.36	89.17 ± 1.13	46.91 ± 0.63	38.79 ± 0.31
Ours (LLC + GRSP)	88.78 ± 0.88	83.77 ± 0.43	89.86 ± 1.06	47.94 ± 0.57	40.65 ± 0.38
Ours (LLC + HFN)	88.35 ± 0.69	83.65 ± 0.38	89.52 ± 0.98	47.45 ± 0.64	39.46 ± 0.26
Ours (LLC + both)	$\mathbf{88.81} \pm 0.78$	$\mathbf{83.96} \pm 0.48$	$\mathbf{90.00} \pm 1.02$	$\mathbf{48.36} \pm 0.58$	$\mathbf{41.09} \pm 0.36$
Ours (IFV [28])	91.89 ± 0.92	89.70 ± 0.58	93.64 ± 0.95	61.87 ± 0.65	48.35 ± 0.37
Ours (IFV + GRSP)	92.38 ± 1.00	90.41 ± 0.48	94.26 ± 1.03	63.01 ± 0.58	50.21 ± 0.28
Ours (IFV + HFN)	92.01 ± 0.93	90.14 ± 0.64	94.07 ± 0.92	62.41 ± 0.49	48.81 ± 0.27
Ours (IFV + both)	$\mathbf{92.39} \pm 1.02$	$\mathbf{90.67} \pm 0.56$	$\mathbf{94.36} \pm 0.99$	$\mathbf{63.71} \pm 0.61$	$\mathbf{50.51} \pm 0.30$

global normalization method. This is not strange since our normalization strategy treats each section of a feature vector as a single part. It works better when the BoF model fits the part-based assumption better, i.e., parts are more semantically meaningful.

D. Comparison With the State of the Art

We compare the results produced by GRSP and HFN with the state-of-the-art algorithms.

1) Scene Recognition: The classification results on scene recognition data sets are shown in Table V. Since spatial context plays an important role in scene image understanding, regular spatial division models such as SPM [49] work well on these tasks. The proposed GRSP algorithm follows the idea of SPM and generalizes it onto a more flexible set of pooling bins. It improves SPM and works even better than algorithms using complicated pooling techniques such as [95]. Although GRSP produces slightly lower accuracy than [94] in which important spatial pooling regions (ISPRs) are learned for semantic pooling, we point out that GRSP is more generalized than ISPR, which is only applied to scene recognition tasks.

2) Generic Object Recognition: We next report generic object recognition results in Table VI. Here, GRSP also produces accuracy gain over SPM, owing to that extra pooling bins help to represent possible objects and/or parts on the image. In [50], an automatic learning algorithm is proposed to construct pooling bins. In comparison, our algorithm is more generalized and easier to implement.

We also provide object retrieval results on the PascalVOC-2007 data set [84]. In these tasks, more similar to image retrieval, our algorithm also produces consistent accuracy gain. It is worth noting that there exist some cases

TABLE VI
COMPARISON OF OUR CLASSIFICATION RESULTS WITH PREVIOUS WORKS ON GENERIC OBJECT RECOGNITION

Algorithm	Bird-6	Butterfly-7	Flower-17	Caltech101	Caltech256	VOC07
Lazebnik *et.al.* [81]	–	90.4	–	–	–	–
Lazebnik *et.al.* [80]	91.67	–	–	–	–	–
Lazebnik *et.al.* [49]	–	–	–	64.6	–	–
Nilsback *et.al.* [82]	–	–	81.3	–	–	–
Gehler *et.al.* [96]	–	–	85.5	77.7	–	–
Larlus *et.al.* [97]	93.00	90.61	–	–	–	–
Yang *et.al.* [36]	–	–	–	73.20	40.14	–
Boureau *et.al.* [92]	–	–	–	75.7	–	–
Wang *et.al.* [37]	–	–	–	73.44	47.68	55.1
Jia *et.al.* [50]	–	–	–	75.3	–	–
Zhu *et.al.* [51]	–	–	–	77.7	41.4	–
Wang *et.al.* [93]	–	–	–	79.2	–	–
Xie *et.al.* [26]	–	90.83	91.56	82.45	50.33	53.64
Ours (LLC [37])	90.10 ± 1.65	87.23 ± 1.21	88.29 ± 0.94	76.61 ± 0.83	49.52 ± 0.36	55.57
Ours (LLC + GRSP)	90.73 ± 1.59	87.89 ± 1.07	88.83 ± 1.02	77.52 ± 0.63	51.13 ± 0.41	56.78
Ours (LLC + HFN)	90.43 ± 1.51	87.58 ± 1.23	88.50 ± 0.80	76.98 ± 0.91	49.80 ± 0.38	55.63
Ours (LLC + both)	$\mathbf{91.07} \pm 1.57$	$\mathbf{88.10} \pm 1.10$	$\mathbf{89.04} \pm 0.93$	$\mathbf{77.69} \pm 0.73$	$\mathbf{51.29} \pm 0.41$	**56.89**
Ours (IFV [28])	93.90 ± 1.45	91.47 ± 0.93	92.29 ± 0.73	81.07 ± 0.78	58.77 ± 0.32	62.18
Ours (IFV + GRSP)	94.27 ± 1.38	92.20 ± 0.90	92.83 ± 0.97	82.11 ± 0.86	60.33 ± 0.37	62.95
Ours (IFV + HFN)	93.93 ± 1.57	91.74 ± 1.02	92.58 ± 0.88	81.19 ± 0.68	59.37 ± 0.45	62.57
Ours (IFV + both)	$\mathbf{94.33} \pm 1.42$	$\mathbf{92.41} \pm 0.96$	$\mathbf{93.04} \pm 0.78$	$\mathbf{82.16} \pm 0.81$	$\mathbf{60.64} \pm 0.31$	**63.07**

TABLE VII
COMPARISON OF OUR CLASSIFICATION RESULTS WITH PREVIOUS WORKS ON FINE-GRAINED OBJECT RECOGNITION

Algorithm	Pet-37	Aircraft-100	Flower-102	Dog-120	Bird-200
Nilsback *et.al.* [87]	–	–	72.8	–	–
Khosla *et.al.* [88]	–	–	–	21.9	–
Wah *et.al.* [89]	–	–	–	–	10.7
Chai *et.al.* [98]	–	–	85.2	26.9	–
Parkhi *et.al.* [85]	59.21	–	–	–	–
Maji *et.al.* [86]	–	48.69	–	–	–
Berg *et.al.* [99]	–	–	–	–	56.78
Chai *et.al.* [55]	–	–	–	45.6	59.4
Gavves *et.al.* [54]	–	–	–	50.1	62.7
Zhang *et.al.* [100]	–	–	–	–	50.98
Murray *et.al.* [46]	56.8	–	84.6	–	33.3
Pu *et.al.* [101]	–	–	–	39.3	44.2
Ours (LLC [37])	52.65 ± 0.61	59.26 ± 0.71	73.05 ± 0.44	30.13 ± 0.87	35.57 ± 0.58
Ours (LLC + GRSP)	53.61 ± 0.57	60.55 ± 0.59	74.21 ± 0.45	31.89 ± 0.79	37.44 ± 0.49
Ours (LLC + HFN)	53.29 ± 0.50	60.10 ± 0.67	74.38 ± 0.61	32.36 ± 0.76	37.01 ± 0.53
Ours (LLC + both)	$\mathbf{53.95} \pm 0.59$	$\mathbf{61.11} \pm 0.58$	$\mathbf{75.09} \pm 0.39$	$\mathbf{32.78} \pm 0.91$	$\mathbf{38.14} \pm 0.55$
Ours (IFV [28])	59.24 ± 0.66	70.12 ± 0.67	82.43 ± 0.51	41.12 ± 0.93	46.61 ± 0.48
Ours (IFV + GRSP)	59.91 ± 0.59	71.47 ± 0.66	83.50 ± 0.44	42.41 ± 0.82	48.25 ± 0.45
Ours (IFV + HFN)	59.78 ± 0.71	71.02 ± 0.59	83.71 ± 0.61	42.51 ± 0.96	47.98 ± 0.55
Ours (IFV + both)	$\mathbf{60.36} \pm 0.60$	$\mathbf{72.18} \pm 0.70$	$\mathbf{84.02} \pm 0.40$	$\mathbf{43.15} \pm 0.87$	$\mathbf{48.78} \pm 0.52$

with small objects in PascalVOC-2007. The use of GRSP significantly increases the possibility of detecting semantic contents in these images (please see Fig. 7 for examples).

3) Fine-Grained Object Recognition: Finally, the fine-grained object recognition results are summarized in Table VII. Although GRSP still works better than SPM, the results are poor compared with those using semantic parts as pooling regions, such as [53]–[55] and [100]. The reason that our algorithm is trailed by a large margin by the above competitors is that we do not use detected object parts that are verified crucial for fine-grained recognition tasks. We leverage the part detection results in [53] on the Bird-200 data set [89]. With detected parts, our algorithm reports 58.09% and 65.41% accuracy, using LLC and IFV encoding, respectively. After adopting the hierarchical structure learning (HSL) algorithm [53], the results are boosted to 59.86% and 66.87%, which are competitive among those reported in Table VII. Since HSL could also be considered as an alternative solution of increasing pooling regions, this experiment once again verifies our statement: a well-designed pooling algorithm is crucial for image classification.

The impact of HFN on fine-grained object recognition is also worth emphasizing. In both scene and generic object recognition tasks, HFN improves the accuracy less significantly, since it is not likely to partition the image into semantic parts with simple rectangular pooling bins. Exceptions come from the fine-grained object recognition, in which the objects are better described by the underlying semantic parts. Although regular grids are not perfect part detectors, it does capture useful information since the pose variation in fine-grained data sets is much smaller.

Fig. 7. Example images on which classification algorithms with (above the dashed line) and without (below) using GRSP. Red frames indicate pooling regions that are additionally generated by GRSP, which better capture visual concepts. On the other hand, yellow frames indicate those less meaningful regions generated by GRSP, which confuse classifiers.

Therefore, normalizing each region individually produces more discriminative feature vectors. Similar results on detected parts are reported in [4].

E. Discussion

Some necessary discussions are made here.

1) Significance of Accuracy Gain: We perform student's t-test to verify that the accuracy gain of our algorithm is statistically significant. On each data set, we compute the likelihood of null hypothesis (p-value) in each of the 10 fixed-split testing rounds. The p-value is smaller than 0.01 in all the small data sets such as Bird-6 and smaller than 10^{-4} in the data sets with more than 100 categories such as SUN-397.

2) Qualitative Analysis: We provide some sample images in Fig. 7 as qualitative analysis. One might observe that GRSP helps to improve classification performance on some images with small objects. However, GRSP also produces wrong results on some images that are correctly classified by SPM. Overall, we find more positive cases than negative cases in every single data set.

3) Weakness of Our Algorithm: The main weakness of our algorithm lies in the GRSP module. Although GRSP could be easily implemented, the increasing number of pooling bins brings in heavier computational overheads on the classification stage. In the future, we will leverage feature selection algorithms such as [50] to alleviate the extra costs.

4) Contribution: The main contribution of this paper lies in that we provide solid evidences, showing that feature pooling and normalization are crucial modules in image classification.

Finally, we shall emphasize that the proposed normalization techniques, i.e., SFN and HFN, are extremely simple and efficient. One needs only few lines of codes to implement them, and there are almost no extra costs on both time and memory. Therefore, we suggest adopting these two algorithms in every part-based image classification model.

VI. Conclusion

In this paper, we propose two simple algorithms, i.e., GRSP and HFN, for summarizing encoded features in the BoF model. Although feature pooling and normalization stages are considered less interesting compared with other modules, we can still obtain consistent accuracy gain with intuitive analysis on the nature of image representation. The proposed algorithms are extremely easy to implement yet very efficient to carry out, and could be freely transplanted onto various types of classification tasks based on the BoF model. The experimental results have revealed that the combined model (GRSP with HFN) helps to improve the classification accuracy on every single case, using either LLC or IFV encoding. We achieve state-of-the-art performance on a wide range of image classification tasks.

References

[1] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proc. Workshop Statist. Learn. Comput. Vis., Eur. Conf. Comput. Vis.*, 2004, pp. 1–2.
[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[3] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 2161–2168.
[4] L. Xie, Q. Tian, and B. Zhang, "Feature normalization for part-based image classification," in *Proc. 20th IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 2607–2611.
[5] L. Xie, Q. Tian, and B. Zhang, "Generalized regular spatial pooling for image classification," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2015, pp. 1374–1378.
[6] E. Martinez-Enriquez, A. Jimenez-Moreno, M. Angel-Pellon, and F. Diaz-de-Maria, "A two-level classification-based approach to inter mode decision in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 11, pp. 1719–1732, Nov. 2011.
[7] M. A. Hasan, M. Xu, X. He, and C. Xu, "CAMHID: Camera motion histogram descriptor and its application to cinematographic shot classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 10, pp. 1682–1695, Oct. 2014.
[8] J.-S. Pan, Q. Feng, L. Yan, and J.-F. Yang, "Neighborhood feature line segment for image classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 3, pp. 387–398, Mar. 2015.
[9] J. Xu, Q. Wu, J. Zhang, F. Shen, and Z. Tang, "Boosting separability in semisupervised learning for object classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 7, pp. 1197–1208, Jul. 2014.
[10] L. Liu, P. W. Fieguth, D. Hu, Y. Wei, and G. Kuang, "Fusing sorted random projections for robust texture and material classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 3, pp. 482–496, Mar. 2014.
[11] V. D. Nguyen, D. D. Nguyen, T. T. Nguyen, V. Q. Dinh, and J. W. Jeon, "Support local pattern and its application to disparity improvement and texture classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 263–276, Feb. 2014.
[12] X. Wen, L. Shao, W. Fang, and Y. Xue, "Efficient feature selection and classification for vehicle detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 3, pp. 508–517, Mar. 2014.
[13] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *Int. J. Comput. Vis.*, vol. 60, no. 1, pp. 63–86, 2004.
[14] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image Vis. Comput.*, vol. 22, no. 10, pp. 761–767, 2004.
[15] T. Tuytelaars, "Dense interest points," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2281–2288.
[16] A. Bosch, A. Zisserman, and X. Muñoz, "Scene classification via pLSA," in *Proc. 9th Eur. Conf. Comput. Vis.*, 2006, pp. 517–530.

[17] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 886–893.

[18] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[19] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.

[20] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "BRIEF: Computing a local binary descriptor very fast," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1281–1298, Jul. 2012.

[21] E. Tola, V. Lepetit, and P. Fua, "DAISY: An efficient dense descriptor applied to wide-baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 815–830, May 2010.

[22] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.

[23] L. Xie, Q. Tian, J. Wang, and B. Zhang, "Image classification with max-SIFT descriptors," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2015, pp. 1369–1373.

[24] R. Arandjelovic and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2911–2918.

[25] A. Bosch, A. Zisserman, and X. Muñoz, "Image classification using random forests and ferns," in *Proc. IEEE 11th Conf. Comput. Vis. Pattern Recognit.*, Oct. 2007, pp. 1–8.

[26] L. Xie, Q. Tian, M. Wang, and B. Zhang, "Spatial pooling of heterogeneous features for image classification," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 1994–2008, May 2014.

[27] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1582–1596, Sep. 2010.

[28] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 143–156.

[29] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.

[30] H. Ling and D. W. Jacobs, "Shape classification using the inner-distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 286–299, Feb. 2007.

[31] J. Donahue *et al.*, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 647–655.

[32] L. Xie, R. Hong, B. Zhang, and Q. Tian, "Image classification and retrieval are ONE," in *Proc. Int. Conf. Multimedia Retr.*, 2015, pp. 3–10.

[33] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[34] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. 10th Eur. Conf. Comput. Vis.*, 2008, pp. 304–317.

[35] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1–8.

[36] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1794–1801.

[37] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3360–3367.

[38] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Proc. Adv. Neural Inf. Process. Syst.*, 1999, pp. 487–493.

[39] X. Zhou, K. Yu, T. Zhang, and T. S. Huang, "Image classification using super-vector coding of local image descriptors," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 141–154.

[40] T. Kobayashi, "BoF meets HOG: Feature extraction based on histograms of oriented p.d.f. gradients for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 747–754.

[41] L. Shao, L. Liu, and X. Li, "Feature learning for image classification via multiobjective genetic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 7, pp. 1359–1371, Jul. 2014.

[42] T. de Campos, G. Csurka, and F. Perronnin, "Images as sets of locally weighted features," *Comput. Vis. Image Understand.*, vol. 116, no. 1, pp. 68–85, 2012.

[43] M. San Biagio, L. Bazzani, M. Cristani, and V. Murino, "Weighted bag of visual words for object recognition," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2014, pp. 2734–2738.

[44] J. Sánchez, F. Perronnin, and T. de Campos, "Modeling the spatial layout of images beyond spatial pyramids," *Pattern Recognit. Lett.*, vol. 33, no. 16, pp. 2216–2223, 2012.

[45] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 111–118.

[46] N. Murray and F. Perronnin, "Generalized max pooling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2473–2480.

[47] J. Feng, B. Ni, Q. Tian, and S. Yan, "Geometric ℓ_p-norm feature pooling for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 2609–2704.

[48] K. Grauman and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, Oct. 2005, pp. 1458–1465.

[49] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 2169–2178.

[50] Y. Jia, C. Huang, and T. Darrell, "Beyond spatial pyramids: Receptive field learning for pooled image features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3370–3377.

[51] J. Zhu, W. Zou, X. Yang, R. Zhang, Q. Zhou, and W. Zhang, "Image classification by hierarchical spatial pooling with partial least squares analysis," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 1–11.

[52] F. Zhu, Z. Jiang, and L. Shao, "Submodular object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2457–2464.

[53] L. Xie, Q. Tian, R. Hong, S. Yan, and B. Zhang, "Hierarchical part matching for fine-grained visual categorization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1641–1648.

[54] E. Gavves, B. Fernando, C. G. M. Snoek, A. W. M. Smeulders, and T. Tuytelaars, "Fine-grained categorization by alignments," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1713–1720.

[55] Y. Chai, V. Lempitsky, and A. Zisserman, "Symbiotic segmentation and part localization for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 321–328.

[56] X. Liu, D. Wang, J. Li, and B. Zhang, "The feature and spatial covariant kernel: Adding implicit spatial constraints to histogram," in *Proc. 6th ACM Int. Conf. Image Video Retr.*, 2007, pp. 565–572.

[57] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag, 1999.

[58] A. B. A. Graf and S. Borer, "Normalization in support vector machines," in *Pattern Recognition*. Berlin, Germany: Springer-Verlag, 2001.

[59] P. Juszczak, D. Tax, and R. P. W. Duin, "Feature scaling in support vector data description," in *Proc. Annu. Conf. Adv. School Comput. Imag.*, 2002, pp. 95–102.

[60] A. Stolcke, S. Kajarekar, and L. Ferrer, "Nonparametric feature normalization for SVM-based speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May/Apr. 2008, pp. 1577–1580.

[61] E. Youn and M. K. Jeong, "Class dependent feature scaling method using naive Bayes classifier for text datamining," *Pattern Recognit. Lett.*, vol. 30, no. 5, pp. 477–485, 2009.

[62] S. Tsakalidis, V. Doumpiotis, and W. Byrne, "Discriminative linear transforms for feature normalization and speaker adaptation in HMM estimation," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 3, pp. 367–376, May 2005.

[63] L. Bo, L. Wang, and L. Jiao, "Feature scaling for kernel Fisher discriminant analysis using leave-one-out cross validation," *Neural Comput.*, vol. 18, no. 4, pp. 961–978, 2006.

[64] S. Aksoy and R. M. Haralick, "Feature normalization and likelihood-based similarity measures for image retrieval," *Pattern Recognit. Lett.*, vol. 22, no. 5, pp. 563–582, 2001.

[65] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[66] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Good practice in large-scale learning for image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 507–520, Mar. 2013.

[67] T. Gao and D. Koller, "Discriminative learning of relaxed hierarchy for large-scale visual recognition," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2072–2079.
[68] M. Sun, W. Huang, and S. Savarese, "Find the best path: An efficient and accurate classifier for image hierarchies," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 265–272.
[69] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *Proc. IEEE Conf. CVPR*, Jun. 2008, pp. 1–8.
[70] N. Zhang, R. Farrell, and T. Darrell, "Pose pooling kernels for sub-category recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3665–3672.
[71] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, Jun. 2008.
[72] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: An evaluation of recent feature encoding methods," in *Proc. Brit. Mach. Vis. Conf.*, 2011, pp. 76.1–76.12.
[73] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the Fisher vector: Theory and practice," *Int. J. Comput. Vis.*, vol. 105, no. 3, pp. 222–245, 2013.
[74] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," Dept. Comput. Sci., Nat. Taiwan Univ., Taipei, Taiwan, Tech. Rep., 2003.
[75] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," *Comput. Vis. Image Understand.*, vol. 106, no. 1, pp. 59–70, 2007.
[76] L.-J. Li and L. Fei-Fei, "What, where and who? Classifying events by scene and object recognition," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.
[77] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, 2010, pp. 270–279.
[78] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 413–420.
[79] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3485–3492.
[80] S. Lazebnik, C. Schmid, and J. Ponce, "A maximum entropy framework for part-based texture and object recognition," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, Oct. 2005, pp. 832–838.
[81] S. Lazebnik, C. Schmid, and J. Ponce, "Semi-local affine parts for object recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2004, pp. 779–788.
[82] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 1447–1454.
[83] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. 7694, 2007.
[84] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. (2007). *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*. [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
[85] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar, "Cats and dogs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3498–3505.
[86] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi. (2013). "Fine-grained visual classification of aircraft." [Online]. Available: arXiv:1306.5151
[87] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proc. 6th Indian Conf. Comput. Vis., Graph., Image Process.*, Dec. 2008, pp. 722–729.
[88] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel dataset for fine-grained image categorization: Stanford dogs," in *Proc. 1st Workshop Fine-Grained Vis. Categorization, IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1–2.
[89] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD birds-200-2011 dataset," Dept. Comput. Neural Syst., California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.
[90] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," in *Proc. ACM Int. Conf. Multimedia*, 2010, pp. 1469–1472.
[91] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 161–168.
[92] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2559–2566.
[93] P. Wang, J. Wang, G. Zeng, W. Xu, H. Zha, and S. Li, "Supervised kernel descriptors for visual recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2858–2865.
[94] D. Lin, C. Lu, R. Liao, and J. Jia, "Learning important spatial pooling regions for scene classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3726–3733.
[95] L. Xie, J. Wang, B. Guo, B. Zhang, and Q. Tian, "Orientational pyramid matching for recognizing indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3734–3741.
[96] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 221–228.
[97] D. Larlus and F. Jurie, "Latent mixture vocabularies for object categorization and segmentation," *Image Vis. Comput.*, vol. 27, no. 5, pp. 523–534, 2009.
[98] Y. Chai, E. Rahtu, V. Lempitsky, L. Van Gool, and A. Zisserman, "TriCoS: A tri-level class-discriminative co-segmentation method for image classification," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 1–14.
[99] T. Berg and P. N. Belhumeur, "POOF: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 955–962.
[100] N. Zhang, R. Farrell, F. Iandola, and T. Darrell, "Deformable part descriptors for fine-grained recognition and attribute prediction," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 729–736.
[101] J. Pu, Y.-G. Jiang, J. Wang, and X. Xue, "Which looks like which: Exploring inter-class relationships in fine-grained visual categorization," in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, pp. 425–440.

Lingxi Xie received the B.E. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2010, and the Ph.D. degree in engineering from Tsinghua University, in 2015.

He was a Research Intern with Microsoft Research Asia, Beijing, in 2013–2015. He was a Visiting Researcher with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX, USA, in 2014. His research interests include computer vision, multimedia information retrieval, and machine learning.

Qi Tian (M'96–SM'03) received the B.E. degree in electronic engineering from Tsinghua University, Beijing, China, in 1992; the M.S. degree in electrical and computer engineering from Drexel University, Philadelphia, PA, USA, in 1996; and the Ph.D. degree in electrical and computer engineering from University of Illinois at Urbana–Champaign, Urbana, IL, USA, in 2002.

He is with University of Texas at San Antonio, San Antonio, TX, USA.

Bo Zhang received the B.E. degree from the Department of Automatic Control, Tsinghua University, Beijing, China, in 1958.

He is a Professor with the Department of Computer Science and Technology, Tsinghua University. His research interests include artificial intelligence, machine learning, pattern recognition, knowledge engineering, intelligent robotics, and intelligent control.

Mr. Zhang is a fellow of the Chinese Academy of Sciences.