Image Classification by Hierarchical Spatial Pooling with Partial Least Squares Analysis

Jun Zhu[1]
junnyzhu@sjtu.edu.cn
Weijia Zou[1]
zouweijia@sjtu.edu.cn
Xiaokang Yang[1]
xkyang@sjtu.edu.cn
Rui Zhang[1]
zhang_rui@sjtu.edu.cn
Quan Zhou[2]
qzhou.lhi@gmail.com
Wenjun Zhang[1]
zhangwenjun@sjtu.edu.cn

[1] Institute of Image Communication and
Information Processing
Shanghai Jiao Tong University
Shanghai, China

[2] Department of Electronics and
Information Engineering
Huazhong University of Science and
Technology
Wuhan, China

Abstract

Recent coding-based image classification systems generally adopt a key step of spatial pooling operation, which characterizes the statistics of patch-level local feature codes over the regions of interest (ROI), to form the image-level representation for classification. In this paper, we present a hierarchical ROI dictionary for spatial pooling, to beyond the widely used spatial pyramid in image classification literature. By utilizing the compositionality among ROIs, it captures rich spatial statistical information via an efficient pooling algorithm in deep hierarchy. On this basis, we further employ partial least squares analysis to learn a more compact and discriminative image representation. The experimental results demonstrate superiority of the proposed hierarchical pooling method relative to spatial pyramid, on three benchmark datasets for image classification.

1 Introduction

In recent image classification systems, spatial pooling is a key step to form the image-level representation from the patch-level local features. It captures meaningful statistical information of local feature codes over different ROIs, and achieves certain spatial invariance property for facilitating classification. On the spatial representation model, which defines the dictionary of ROIs in spatial pooling, the spatial pyramid is predominately used in image classification literature [2, 3, 12, 13, 20, 22]. As shown in Fig. 2(a), it partitions the image lattice into regular cells with increasing granularity (e.g., $1\times1+2\times2+4\times4$ grids), to characterize spatial statistics over various scales and locations. In spite of the success of spatial pyramid in practical applications, its rigid structure may limit the resultant image representation from exploring richer spatial statistical information further.

http://dx.doi.org/10.5244/C.26.102

Based on the tangram model [23], which learns flexible and adaptive configurations for scene representation, we construct a hierarchical ROI dictionary (called by HRD in this paper for short) for spatial pooling. Compared to rigid spatial pyramid model, it assembles the ROIs with more shape types, locations and scales, and is capable of retaining richer spatial statistical information. Besides, by taking advantage of mutual compositionality among ROIs, HRD can be inherently organized into a directed acyclic graph, and this derives an efficient hierarchical algorithm to facilitate spatial pooling.

Although the pooled features can be directly used for classification, it has two drawbacks: (1) Caused from the over-completeness and compositionality of ROIs in HRD, it tends to be highly correlated and redundant between the variables of pooled features; (2) For a HRD with large number of ROIs, it produces a huge number of variables in this raw feature representation and may obstruct large-scale image classification. Motivated by the success of partial least squares (PLS) in computer vision literature [17, 18], we further employ the PLS analysis for dimensionality reduction on the pooled features. It can capture the statistical relationship between pooled features and class labels for different visual words, and learn a more compact and discriminative feature representation for classification.

The contributions of this paper are summarized as follows:

- Based on the tangram model [23], an over-complete HRD is constructed for spatial pooling, to supply richer spatial statistics than the spatial pyramid.
- An efficient algorithm is proposed for spatial pooling in deep hierarchy, by utilizing the compositionality among the ROIs in HRD.
- By employing the PLS analysis, we can learn a compact and discriminative image-level representation for classification.

The remainder of this paper is organized as follows: Sec. 2 outlines the coding-based image classification framework adopted in this paper, with discussion on related works. In Sec. 3, we elaborate the method of constructing HRD as well as the algorithm for hierarchical spatial pooling. Then, Sec. 4 introduces the PLS analysis to learn final image representation for classification. In Sec. 5, we demonstrate experimental results of the proposed method on three benchmark datasets for image classification, and conclude this paper in Sec. 6.

2 The Coding-based Image Classification Framework

In this paper, we adopt the coding-based image classification framework. As illustrated in Fig. 1, it mainly involves the following three steps: *feature extraction*, *coding* and *pooling*.

(a) **Feature Extraction:** For an image $\mathbf{I}$, we first collect P local image patches through densely sampling in a regular grid or using an interest point detector. After that, each patch is represented by a low-level feature descriptor $\mathbf{a} \in \mathbb{R}^D$ (e.g., SIFT [14]), and a set of local features $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_P] \in \mathbb{R}^{D \times P}$ is extracted from $\mathbf{I}$.

(b) **Coding:** Given a codebook (denoted by $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_K] \in \mathbb{R}^{D \times K}$) with K visual words, each local feature $\mathbf{a}_t$ is encoded into a code vector $\mathbf{c}_t = [c_1^{(t)}, c_2^{(t)}, \cdots, c_K^{(t)}]^{\mathrm{T}}$, through solving a generalized least squares problem defined in Eq. (1).

$$\mathbf{c}_t = \arg\min_{\mathbf{c}} \left|\left| \mathbf{a}_t - \mathbf{B}\mathbf{c} \right|\right|_2^2 + \mathcal{M}(\mathbf{c}), \quad s.t.\ \ \mathbf{c} \in \mathcal{R}, \tag{1}$$

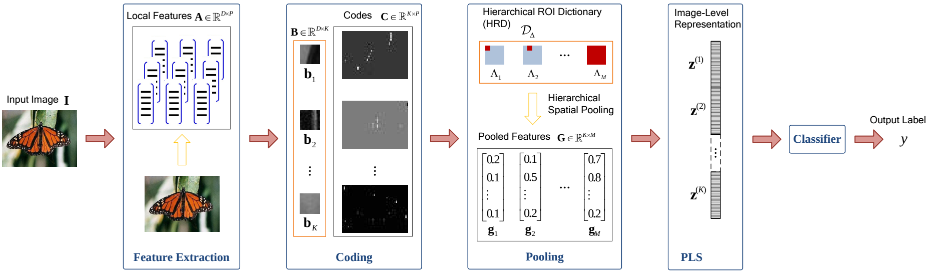

Input Image $\mathbf{I}$
Local Features $\mathbf{A} \in \mathbb{R}^{D \times P}$
Feature Extraction
Codes $\mathbf{C} \in \mathbb{R}^{K \times P}$
$\mathbf{B} \in \mathbb{R}^{D \times K}$
$\mathbf{b}_1$
$\mathbf{b}_2$
$\mathbf{b}_K$
Coding
Hierarchical ROI Dictionary (HRD) $\mathcal{D}_\Delta$
Λ_1
Λ_2
Λ_M
Hierarchical Spatial Pooling
Pooled Features $\mathbf{G} \in \mathbb{R}^{K \times M}$
0.2 0.1 0.1
0.1 0.5 0.2
0.7 0.8 0.2
$\mathbf{g}_1$
$\mathbf{g}_2$
$\mathbf{g}_M$
Pooling
Image-Level Representation
$\mathbf{z}^{(1)}$
$\mathbf{z}^{(2)}$
$\mathbf{z}^{(K)}$
PLS
Classifier
Output Label
y

Figure 1: Overview of our framework on image classification. Best viewed in color.

where $\mathcal{M}$ and $\mathcal{R}$ refer to the regularization term and feasible region on $\mathbf{c}$ respectively. The choice of $\mathcal{M}$ and $\mathcal{R}$ derives different coding schemes in literature. E.g., $\mathcal{R} = \{\mathbf{c} \mid ||\mathbf{c}||_0 = 1, ||\mathbf{c}||_1 = 1 \text{ and } \mathbf{c} \geq 0\}$ deduces vector quantization (VQ) used in the 'bag of words' model [5]. $\mathcal{M}(\mathbf{c}) = \lambda ||\mathbf{c}||_1$ realizes the sparse coding in [22]. $\mathcal{M}(\mathbf{c}) = \lambda \sum_{k=1}^{K} [\exp(\frac{||\mathbf{a}_t - \mathbf{b}_k||_2}{\sigma}) \cdot c_k]^2$ and $\mathcal{R} = \{\mathbf{c} \mid \sum_{k=1}^{K} c_k = 1\}$ corresponds to the locality-constrained linear coding (LLC) [20]. After the coding step, $\mathbf{I}$ is represented by a set of codes $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_P] \in \mathbb{R}^{K \times P}$.

(c) **Pooling:** The resultant codes $\mathbf{C}$ are still patch-level representation and highly redundant to represent $\mathbf{I}$. For preserving diagnostic information and achieving certain invariance (e.g., transformation invariance) on $\mathbf{C}$, a pooling step is adopted to form the image-level representation $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_M] \in \mathbb{R}^{K \times M}$, via summarizing the statistics over the local feature codes in M different pre-defined ROIs.

Let Λ_m denote the m^{th} ROI for spatial pooling. For each visual word $\mathbf{b}_k$, we map its codes of local image patches belonging to Λ_m into a statistical signature $g_m^{(k)}$ by the p-norm function

$$g_m^{(k)} = f_p(\mathbf{C}; \Lambda_m, \mathbf{b}_k) = [\sum_{\rho_t \in \Lambda_m} (c_k^{(t)})^p]^{\frac{1}{p}}, \tag{2}$$

where ρ_t denotes location of the t^{th} patch. $\mathbf{g}_m = [g_m^{(1)}, g_m^{(2)}, \cdots, g_m^{(K)}]^{\mathrm{T}}$ represents the pooled feature in Λ_m. Eq. (2) coincides with two widely used pooling operations: e.g., $p = 1$ corresponds to *sum pooling* (i.e. $f_{sum}(\cdot) = f_1(\cdot)$), while $p = \infty$ corresponds to *max pooling* (i.e. $f_{max}(\cdot) = f_\infty(\cdot)$). To go beyond the spatial pyramid, there are some latest works [7, 10, 11, 19, 23] proposed in literature, by learning more flexible spatial layout adaptive to the statistics for different image categories.

3 Hierarchical Spatial Pooling

In this section, we first construct the HRD based on tangram model [23]. After that, an efficient algorithm is presented for spatial pooling with HRD.

3.1 Building Hierarchical ROI Dictionary Based on Tangram Model

As in [23], a layered dictionary of shape primitives (called *tans*[1]) is constructed to quantize the spatial configuration space. Each tan is defined as a kind of connected polygon shape

[1] We inherit the terminologies and notations in [23].

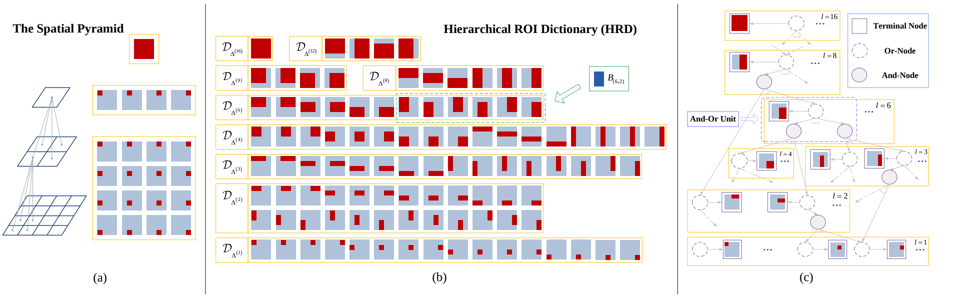

The Spatial Pyramid
(a)
Hierarchical ROI Dictionary (HRD)
$\mathcal{D}_{\Delta^{(16)}}$
$\mathcal{D}_{\Delta^{(12)}}$
$\mathcal{D}_{\Delta^{(9)}}$
$\mathcal{D}_{\Delta^{(8)}}$
$B_{(6,2)}$
$\mathcal{D}_{\Delta^{(6)}}$
$\mathcal{D}_{\Delta^{(4)}}$
$\mathcal{D}_{\Delta^{(3)}}$
$\mathcal{D}_{\Delta^{(2)}}$
$\mathcal{D}_{\Delta^{(1)}}$
(b)
$l=16$
$l=8$
$l=6$
$l=4$
$l=3$
$l=2$
$l=1$
And-Or Unit
Terminal Node
Or-Node
And-Node
(c)

Figure 2: (a) Illustration on the 3-layer spatial pyramid and its ROI dictionary. (b) Illustration on the 16-layer *Squ-HRD*. (Note that there is no tan available at the layers of $l \in \{5,7,10,11,13,14,15\}$). (c) Illustration of the associated AOG for HRD (Only a portion of graph is shown for clarity). Best viewed in color with magnification.

composed of several non-overlapping primitives in a grid of image lattice Λ. In this paper, besides the four types of triangles used in [23], we consider using the square shape as primitive. It can create a HRD with only rectangular ROIs, to be a more natural counterpart w.r.t. the spatial pyramid. For short, we call the triangle-based HRD and the square-based one by *Tri-HRD* and *Squ-HRD*, respectively.

Formally, the tan dictionary $\Delta = \bigcup_{l=1}^{L} \Delta^{(l)}$ is an union of L subsets. $\Delta^{(l)} = \{B_{(l,i)} \mid i = 1,2,\cdots,N_l\}$ denotes a set of tans for the l^{th} layer, where $B_{(l,i)}$ refers to the i^{th} tan. Given the type of shape primitives at the first layer and predefined compositional rules [23], Δ can be automatically created through a bottom-up process of recursive shape composition. Moreover, as in [23], to describe the compositionality among tans, an associated And-Or graph (AOG) Υ_Δ is accordingly built for organizing the tan dictionary Δ in a deep hierarchy. In Υ_Δ, the And-node represents that a tan can be composed by two smaller ones in layers below, while the Or-node implies that it can be generated in alternative ways of shape composition.

When placing each tan onto different locations in the image lattice, one tan $B_{(l,i)}$ may produce a set of $J_{(l,i)}$ different instances $\{\Lambda_{(l,i,j)} \mid j = 1,2,\cdots,J_{(l,i)}\}$, which are the ROIs used for spatial pooling in our framework. Moreover, Because of the mirrored correspondence between a tan and its instances, $\Lambda_{(l,i,j)}$ inherits all the And-Or compositionality from $B_{(l,i)}$, and there is also an isomorphic AOG Υ'_Δ built to organize the tan instances for Δ. More details on the tangram model can be found in [23]. Thus, in this paper, we build the HRD, denoted by $\mathcal{D}_\Delta$, via a layered collection of ROIs instantiated from the tans in Δ. That is

$$\begin{aligned} \mathcal{D}_\Delta &= \textstyle\bigcup_{l=1}^{L} \mathcal{D}_{\Delta^{(l)}}, \\ \forall\, l, \quad \mathcal{D}_{\Delta^{(l)}} &= \{\Lambda_{(l,i,j)} \mid i = 1,2,\cdots,N_l \text{ and } j = 1,2,\cdots,J_{(l,i)}\}. \end{aligned} \tag{3}$$

Similar as [23], we also define an *And-Or unit* $V_{(l,i,j)} = \{v^{T}_{(l,i,j)}, v^{Or}_{(l,i,j)}, \{v^{And}_{(l,i,j),o}\}_{o=1}^{O_{(l,i)}}\}$ for each ROI $\Lambda_{(l,i,j)}$, where $v^{T}_{(l,i,j)}$, $v^{Or}_{(l,i,j)}$ and $v^{And}_{(l,i,j),o}$ refer to the terminal node, Or-node and And-node respectively. The And-Or units are the elements to constitute Υ'_Δ. Specifically, the terminal node $v^{T}_{(l,i,j)}$ corresponds to the ROI $\Lambda_{(l,i,j)}$ itself. The And-node $v^{And}_{(l,i,j),o}$ represents that $\Lambda_{(l,i,j)}$ can be partitioned into two ones in layers below, while the Or-node $v^{Or}_{(l,i,j)}$ implies that it can either terminate into the terminal node or alternatively decompose into its child And-nodes in one of $O_{(l,i)}$ different ways. For example, Fig. 2 (b) illustrates a 16-layer *Squ-HRD* for 4×4 grid, with the associated And-Or graph shown in Fig. 2 (c).

Actually, the ROIs defined in our HRD are equivalent to the overcomplete receptive fields adopted in [10]. In addition to this, the HRD is constructed based on mutual compositionality among the ROIs, and can be inherently organized into a deep hierarchy via associated AOG, which leads to an efficient hierarchical pooling algorithm in Sec. 3.2.

3.2 Efficient Spatial Pooling in Deep Hierarchy

Based on the HRD built in Sec. 3.1, we can perform spatial pooling operation over the ROIs. Due to the over-completeness and increasing degree of freedom induced by recursive shape composition, the cardinality of HRD (i.e., the number of ROIs involved) grows drastically with its granularity level. E.g., there are totally 100 ROIs in a 16-layer *Squ-HRD* with 4×4 grid, while the cardinality of a 64-layer *Squ-HRD* with 8×8 grid rises sharply to 1296. Hence, direct spatial pooling operation on HRD is computationally demanding.

However, the over-completeness and compositionality of the ROIs result in that each ROI in the HRD can be exactly composed by its child ones in the layers below. This implies that most computational cost can be saved by taking advantage of recursive compositionality among the ROIs. Considering the directed acyclic structure [23] of associated AOG Υ'_Δ with deep hierarchy, we present an efficient algorithm for spatial pooling on HRD $\mathcal{D}_\Delta$.

For a ROI $\Lambda_{(l,i,j)}$, we denote its pooled feature by a K-dimensional vector $\mathbf{g}_{(l,i,j)}$, where the value of its k^{th} element (denoted by $g^{(k)}_{(l,i,j)}$ [2]) refers to the pooled signature for visual word $\mathbf{b}_k$. Given a HRD $\mathcal{D}_\Delta$ as well as associated AOG Υ'_Δ, the proposed pooling algorithm can be divided into two steps: I. For each ROI at the first layer, we directly compute its pooled feature by Eq. (2); II. For the other layers above, the pooled feature for each ROI is bottom-up propagated from its child nodes.

For a node v in Υ'_Δ, let $\mathbf{g}_v$ and $Ch(v)$ denote its pooled feature vector and the set of child nodes, respectively. Besides, we use the functions of $MAX(\cdot)$ (i.e., The max operation of $MAX(\mathbf{g}_v)$ is element-wise such that $\forall k$, $g^{(k)}_v = \max_{v' \in Ch(v)} g^{(k)}_{v'}$) and $SUM(\cdot)$ (i.e., $SUM(\mathbf{g}_v) = \sum_{v' \in Ch(v)} \mathbf{g}_{v'}$) to denote the element-wise max and sum operations, respectively. Thus, our hierarchical spatial pooling algorithm is summarized in Alg. 1 [3].

In step I, we observe that the direct pooling manipulations on codes $\mathbf{C}$ are reduced, corresponding to the number of the-1st-layer ROIs (e.g., an 8×8 *Squ-HRD* with the cardinality of 1296 has only 64 ROIs at the 1st layer). In step II, for each ROI at layers above, the recursive pooling manipulation just requires element-wise max or sum operation over the pooled features of its child nodes, which subjects to much less computational cost than direct pooling operation from codes. By Alg. 1, we obtain the pooled features $\mathbf{G}$, which is a $K \times M$ matrix introduced in Sec. 2. M is equal to the total number of ROIs in $\mathcal{D}_\Delta$ such that $M = \sum_{l=1}^{L} \sum_{i=1}^{N_l} J_{(l,i)}$.

4 Learning Image Representation with PLS Analysis

The partial least squares analysis is a classical statistical method for modeling relations between two sets of observed variables. The underlying assumption of PLS is that the observed data is generated from a small number of latent variables [15]. In this section, we introduce

[2] For a ROI in HRD, the trituple subscript (l,i,j) is interchangeably used with the linear index of $g^{(k)}_m$ in Eq. (2).

[3] For simplicity, we use max pooling as an example, and the case of sum pooling just requires slight modification of displacing $MAX(\cdot)$ and $f_{max}(\cdot)$ by $SUM(\cdot)$ and $f_{sum}(\cdot)$ respectively.

Algorithm 1: The Algorithm on Hierarchical Spatial Spooling

```
Input: HRD $\mathcal{D}_\Delta$, AOG $\Upsilon'_\Delta$, the codes $\mathbf{C}$ w.r.t. a visual dictionary $\mathbf{B}$
Output: the pooled features $\mathbf{G}$ on $\mathcal{D}_\Delta$
Initialize $\mathbf{G} = \{\emptyset\}$ and $M = 0$;
Step I: computing the pooled features for the ROIs at the $1^{st}$ layer in $\mathcal{D}_\Delta$
foreach ROI $\Lambda_{(1,i,j)}$ with $i = 1$ to $N_1$ and $j = 1$ to $J_{(1,i)}$ do
    foreach visual word $\mathbf{b}_k$ with $k = 1$ to $K$ do
        $g^{(k)}_{v^T_{(1,i,j)}} = f_{max}(C; \Lambda_{(1,i,j)}, \mathbf{b}_k)$;
    end
    Let $M = M + 1$ and $\mathbf{G} = [\mathbf{G}, \mathbf{g}_{(1,i,j)}]$, where $g^{(k)}_{(1,i,j)} = g^{(k)}_{v^T_{(1,i,j)}}, \forall k = 1, 2, \cdots, K$;
end
Step II: bottom-up computing the pooled features for the ROIs in other layers above
foreach level $l = 2$ to $L$ do
    foreach ROI $\Lambda_{(l,i,j)}$ with Or-node $v^{Or}_{(l,i,j)}$, $i = 1$ to $N_l$ and $j = 1$ to $J_{(l,i)}$ do
        foreach And-node $v^{And}_{(l,i,j),o}$ with $o = 1$ to $O_{(l,i)}$ do
            $\mathbf{g}_{v^{And}_{(l,i,j),o}} = MAX(\mathbf{g}_{v^{And}_{(l,i,j),o}})$;
        end
        $\mathbf{g}_{(l,i,j)} = \mathbf{g}_{v^{Or}_{(l,i,j)}} = \frac{1}{O_{(l,i)}} \sum_{o=1}^{O_{(l,i)}} \mathbf{g}_{v^{And}_{(l,i,j),o}}$; Let $M = M + 1$ and $\mathbf{G} = [\mathbf{G}, \mathbf{g}_{(l,i,j)}]$.
    end
end
```

a method of learning a compact and discriminative image-level representation with PLS analysis. Moreover, since different visual words generally have distinct spatial statistics for different image categories, we learn the PLS model for each visual word individually, to preserve class-specific discriminative information in the extracted representation.

Let $\Omega = \{(\mathbf{I}_n, y_n)\}_{n=1}^{N}$ denotes a collection of N training images, where $y_n \in \{1, 2, \cdots, \mathcal{C}\}$ refers to the class label. As described in Sec. 3, for each image $\mathbf{I}_n$, we obtain its pooled feature matrix $\mathbf{G}_n = [\mathbf{x}_n^{(1)}; \mathbf{x}_n^{(2)}; \cdots; \mathbf{x}_n^{(K)}]$, where the row vector $\mathbf{x}_n^{(k)} = [g_{n,1}^{(k)}, g_{n,2}^{(k)}, \cdots, g_{n,M}^{(k)}]$ assembles the pooled features over M ROIs for the k^{th} visual word. For notation simplicity, we drop the visual word's index k for $\mathbf{x}_n^{(k)}$ in the following discussion.

For each visual word, we collect a set of pooled features over all training images, and denote it by a $N \times M$ matrix $\mathbf{X} = [\mathbf{x}_1; \mathbf{x}_2; \cdots; \mathbf{x}_N]$. As in [16], we define a $N \times (\mathcal{C} - 1)$ indicator matrix $\mathbf{Y} = [\mathbf{y}_1; \mathbf{y}_2; \cdots; \mathbf{y}_N]$ for multi-class discrimination. Meanwhile, $\mathbf{y}_n$ is a binary row vector indicating class membership for the n^{th} training image. If $y_n \in \{1, 2, \cdots, \mathcal{C} - 1\}$, the y_n^{th} element of $\mathbf{y}_n$ is 1 and the others are 0; otherwise, it is an all-zero vector when $y_n = \mathcal{C}$. By using the nonlinear iterative partial least squares (NIPALS) algorithm [21], PLS iteratively pursues the weight vectors (i.e. projection directions) $\mathbf{w}_q$ and $\mathbf{s}_q$ such that

$$[cov(\mathbf{t}_q, \mathbf{u}_q)]^2 = \max_{\|\mathbf{w}_q\| = \|\mathbf{s}_q\| = 1} [cov(\mathbf{X}\mathbf{w}_q, \mathbf{Y}\mathbf{s}_q)]^2 = \max_{\|\mathbf{w}_q\| = \|\mathbf{s}_q\| = 1} var(\mathbf{X}\mathbf{w}_q)[corr(\mathbf{X}\mathbf{w}_q, \mathbf{Y}\mathbf{s}_q)]^2 var(\mathbf{Y}\mathbf{s}_q), \tag{4}$$

where $\mathbf{t}_q$ and $\mathbf{u}_q$ are the score vectors (i.e., latent variables) extracted in the q^{th} iteration. Meanwhile, $cov(\cdot, \cdot)$ and $corr(\cdot, \cdot)$ respectively denote the sample covariance and the sample correlation between two vectors, and $var(\cdot)$ denotes the sample variance function [15]. After that, the matrices $\mathbf{X}$ and $\mathbf{Y}$ are respectively deflated through subtracting their rank-one approximations based on $\mathbf{t}_q$ and $\mathbf{u}_q$. This process iterates until the norm of data is smaller than a predefined threshold or a desired number of score vectors are extracted. Besides, according to the suggestion of removing the Y-space penalty $var(\mathbf{Y}\mathbf{s}_q)$ in PLS discrimination [1, 16], it requires a simple modification on the indicator matrix (i.e., $\hat{\mathbf{Y}} = \mathbf{Y}(\mathbf{Y}^{\mathrm{T}}\mathbf{Y})^{-\frac{1}{2}}$).

As a result, we learn a set of weight matrices $\{\mathbf{W}^{(k)}\}_{k=1}^{K}$, where $\mathbf{W}^{(k)} = [\mathbf{w}_1^{(k)}, \mathbf{w}_2^{(k)}, \cdots, \mathbf{w}_Q^{(k)}]$ represents the learned linear subspace model for k^{th} visual word. Then, we perform dimension reduction on the pooled features via projecting $\mathbf{G}_n$ onto each $\mathbf{W}^{(k)}$ individually, obtaining a Q-dimensional vector $\mathbf{z}_n^{(k)}$. Thus, a new image-level representation is constructed by concatenating $\mathbf{z}_n^{(k)}$ for all the K visual words, which is used for classification in our framework.

Compared with widely used subspace learning method of principle component analysis (PCA), the optimization criterion of PLS considers not only the sample variance $var(\mathbf{X}\mathbf{w}_q)$ but also the correlation between $\mathbf{X}$ and $\mathbf{Y}$ (i.e., $corr(\mathbf{X}\mathbf{w}_q, \mathbf{Y}\mathbf{s}_q)$), leading to more discriminative projection directions learned. Similar to PLS, the Fisher linear discriminant analysis (FDA) also utilizes the class label information to facilitate discrimination, by maximizing between-class seperation relative to within-class variance. However, FDA has the limitation that there can be up to $\mathcal{C}-1$ non-trivial directions learned for dimension reduction. Besides, the PLS algorithm is immune to the singularity difficulty in PCA and FDA when $M > N$.

5 Experiments

In this section, we demonstrate the superiority of our method through a series of experiments on three benchmark datasets (i.e. Caltech-101 [6], Caltech-256 [9] and Scene-15 [12]) in image classification literature.

For feature extraction, we use a single type of SIFT feature throughout all the experiments. The SIFT features are extracted from densely sampled 16×16 pixel patches, on a grid with the step size of 6 pixels. All the images are converted into gray scale, with the maximum size of height and width no larger than 300 pixels. In the coding step, we construct a codebook with 1024 visual words via standard K-means clustering. The LLC coding is adopted, with the same parameter settings used in [20]. As in [3, 20, 22], the operation of max pooling is adopted in the pooling step. For multi-class discrimination, a linear SVM classifier [4] is trained by "one-vs-rest" manner. Following common settings in literature, we run 10 time experiments with different random splits of training and testing images. The performance is measured by the average of per-class classification accuracy as in [12, 20, 22].

For comparison, a baseline method is implemented by spatial pooling with the 3-layer spatial pyramid. In our method, we use 7×7 *Squ-HRD*, and the number of extracted score vectors in PLS for each visual word is set by 21, which results in an image-level representation with the same dimension produced by the baseline. It is noted that [20] also reports results for the baseline method (i.e., LLC coding with 3-layer spatial pyramid pooling). The goal of using our own implementation on the baseline is to provide more fair and precise comparison, through sharing common experimental settings (e.g., the implementation of local feature, training/testing splits, etc.). Besides, we also test the case of extracting SIFT features from the patches with three scales (i.e., 16×16, 25×25 and 31×31), which is adopted in [20] for Caltech-101 and Caltech-256.

Dataset		Caltech-101	Caltech-256	Scene-15
Method	Patch Scales			
Spatial Pyramid	16	70.9 ± 0.8	34.8 ± 0.2	81.4 ± 0.4
Our Method	16	74.2 ± 0.6	38.1 ± 0.2	81.5 ± 0.7
Spatial Pyramid	16, 25, 31	74.2 ± 0.9	38.3 ± 0.2	81.5 ± 0.5
Our Method	16, 25, 31	**77.7** ± 0.7	**41.4** ± 0.3	**82.4** ± 0.7
LLC [20]		73.44	41.19	-
ScSPM [22]		73.2 ± 0.54	34.02 ± 0.35	80.28 ± 0.93
KSPM [12]		64.6 ± 0.8	-	81.4 ± 0.5
LSAQ [13]		74.21 ± 0.81	-	82.70 ± 0.39
Boureau *et al.* [2]		75.7 ± 1.1	-	85.6 ± 0.2
Boureau *et al.* [3]		77.3 ± 0.6	41.7 ± 0.8	-
Jia *et al.* [10]		75.3 ± 0.70	-	-
Feng *et al.* [7]		82.60	43.17	83.20
Sharma *et al.* [19]		-	-	80.1 ± 0.6
Gemert *et al.* [8]		64.14 ± 1.18	27.17 ± 0.46	76.67 ± 0.39
Griffin *et al.* [9]		67.6	34.1	-

Table 1: Comparison on Classification Performance (%)

5.1 Results on Caltech-101

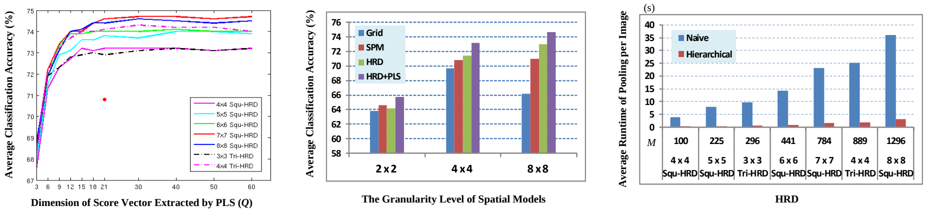
The Caltech-101 dataset [6] consists of 102 different categories (including 101 object categories and 1 additional background category) for object categorization, and contains 9144 images in total, with a varying number of images from 31 to 800 per class. For each run of experiment, we use 30 images per category for training and the rest for testing[4].

Our experimental results are shown in the second column of Table 1. In the case of one-scale local features extracted, our method achieves the average classification rate of 74.2% relative to 70.9% for the baseline method. When coupled with three-scale local features, the performance of our method increases to 77.7%, which is still beyond 74.2% obtained by the baseline. Besides, detailed comparisons with existing methods are also listed in Table 1. We observe that our method can achieve better performance than most of other ones in image classification literatures. E.g., compared to the result reported by the LLC paper [20], our method obtains the performance gain with a margin of 4.3%. In addition, our method outperforms the performance in [10] by a margin of 2.4%, which also employs an over-complete ROI dictionary for spatial pooling and is the most related work to ours. As Jia *et al.* [10] said, it is noted that [7] has reported the best performance on this dataset so far, by simultaneously learning the parameter p in p-norm function and the spatial weight map with a much larger visual codebook (i.e., $K = 4096$). Actually, it also benefits from taking advantage of richer spatial statistics than spatial pyramid.

5.2 Results on Caltech-256

The Caltech-256 dataset [9], which is an extension of Caltech-101 with much higher intra-class variation and wilder object location distribution, involves 257 categories (256 object

[4] The protocal on testing image is different from [20], which uses at most 50 images per class in testing. However, we find that this influence on performance evaluation is neglectable in our experiments.



(a) (b) (c)

Figure 3: (a) Performance with HRD type and Q. The red dot represents the baseline method of 3-layer spatial pyramid. (b) Performance comparison on different spatial representation models. (c) Comparison on runtime of spatial pooling algorithm. See Sec. 5.4 for details.

categories plus a background one). It contains 30607 images in total, with at least 80 images per class. Similar as Sec. 5.1, we also use 30 samples from each class for training and rest ones for testing. As shown in the third column of Table 1, our method consistently outperforms the baseline. Concretely, the performance improvements are 3.3% and 3.1% for the cases of extracting local features in one scale and three scales, respectively. Moreover, our method obtains comparable result (41.4%) relative to the one reported in [20] (41.19%), which uses a larger visual codebook ($K = 4096$) than ours.

5.3 Results on Scene-15

The Scene-15 dataset [12] is a widely used dataset in scene classification literatures [7, 8, 22], which contains 15 different scene categories involving outdoor natural scenes (e.g., coast, mountain and street) and indoor ones (e.g., bedroom, office room). It is composed of totally 4485 images, varying from 200 to 400 images for each category. Following [12], we use 100 images per class for training and the rest for testing. As shown in the fourth column of Table 1, our method can achieve better performance than the baseline method of spatial pyramid.

5.4 Analysis and Discussion

In this subsection, we provide more comprehensive analysis and discussion on our method. At first, we evaluate the classification performance w.r.t. different parameter settings (e.g., the type and granularity of HRD, the number of score vectors extracted in PLS) on Caltech-101 dataset, and illustrate the results in Fig. 3(a). Generally, a HRD with higher granularity, which uses more ROIs for spatial pooling, makes for classification performance. However, we can see that the performance improvement tends to be saturated until the 7×7*Squ-HRD*, and continuously increasing of granularity appears not to boost performance further. On the other side, the impact of performance w.r.t. parameter Q in PLS presents similar tendency: as Q becomes larger, the performance continuously increases until about 20 score vectors (projection directions) are extracted. It implies that the PLS analysis in our method can effectively learn a low dimensional representation with salient discriminative information preserved. This is also supported by Fig. 3(b), which demonstrates consistent superiority of the learned feature representation by PLS in various granularity levels of HRDs.

Moreover, we provides a comprehensive comparison on the effect of different ROI dictionaries with 2×2, 4×4 and 8×8 grids respectively. As shown in Fig. 3(b), it demonstrates that a hierarchical spatial representation (HRD or spatial pyramid) always outperforms its flat grid counterpart, and utilizing richer spatial statistics as the HRD does makes for classification. Finally, as shown in Fig. 3(c), the per-image average runtime[5] of hierarchical pooling algorithm in Sec. 3.2 is evaluated and compared with the naive way of directly pooling over all the ROIs from codes. We find that the proposed algorithm can speed up the naive one by at least 10 times, and its runtime grows much slower with the complexity of HRD.

6 Conclusion

This paper presents a hierarchical spatial pooling method based on HRD for image classification. Compared with spatial pyramid, the HRD employs more flexible ROIs to utilize richer spatial statistics. An efficient pooling algorithm is proposed based on compositionality of ROIs. We also adopt PLS analysis to learn a more compact and discriminative image representation. Experimental results validate the superiority of our method w.r.t. spatial pyramid.

Acknowledgements: This work was supported in part by NSFC (61025005, 61129001, 61071155), 973 Program (2010CB731401, 2010CB731406), and the 111 Project (B07022).

References

[1] M. Barker and W. S. Rayens. Partial least squares for discrimination. *Journal of Chemometrics*, 17:166–173, 2003.

[2] Y. L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *Proc. CVPR*, 2010.

[3] Y. L. Boureau, N. L. Roux, F. Bach, J. Ponce, and Y. LeCun. Ask the locals: multi-way local pooling for image recognition. In *Proc. ICCV*, 2011.

[4] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[5] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proc. CVPR*, 2005.

[6] 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. In *CVPR Workshop on Generative-Model Based Vision*, 2004.

[7] J. Feng, B. Ni, Q. Tian, and S. Yan. Geometric ℓ_p-norm feature pooling for image classification. In *Proc. CVPR*, 2011.

[8] J. C. Gemert, J. M. Geusebroek, C. Veenman, and A. Smeulders. Kernel codebooks for scene categorization. In *Proc. ECCV*, 2008.

[9] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. *Technical Report 7694, California Institute of Technology*, 2007.

[5] We conduct this experiment on a Dell PowerEdge server with 3.0 Ghz Quad Core CPU and 16 GB memory.

[10] Y. Jia, C. Huang, and T. Darrell. Beyond spatial pyramids: Receptive field learning for pooled image features. In *Proc. CVPR*, 2012.

[11] J. Krapac, J. Verbeek, and F. Jurie. Modeling spatial layout with fisher vectors for image categorization. In *Proc. ICCV*, 2011.

[12] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. CVPR*, 2006.

[13] L. Liu, L. Wang, and X. Liu. In defense of soft-assignment coding. In *Proc. ICCV*, 2011.

[14] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[15] R. Rosipal and N. Krämer. Overview and recent advances in partial least squares. *Subspace, Latent Structure and Feature Selection*, pages 34–51, 2006.

[16] R. Rosipal, L. J. Trejo, and B. Matthews. Kernel pls-svc for linear and nonlinear classification. In *Proc. ICML*, 2003.

[17] W. R. Schwartz and L. S. Davis. Learning discriminative appearance-based models using partial least squares. In *Proc. Computer Graphics and Image Processing*, 2009.

[18] W. R. Schwartz, A. Kembhavi, D. Harwood, and L. S. Davis. Human detection using partial least squares analysis. In *Proc. ICCV*, 2009.

[19] G. Sharma and F. Jurie. Learning discriminative spatial representation for image classification. In *Proc. BMVC*, 2011.

[20] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Proc. CVPR*, 2010.

[21] H. Wold. Soft modeling by latent variables; the nonlinear iterative partial least squares ap- proach. *Perspectives in Probability and Statistics*, pages 520–540, 1975.

[22] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Proc. CVPR*, 2009.

[23] J. Zhu, T. Wu, S. C. Zhu, X. Yang, and W. Zhang. Learning reconfigurable scene representation by tangram model. In *Proc. WACV*, 2012.