Multiple spatial pooling for visual object recognition

Yongzhen Huang*, Zifeng Wu, Liang Wang, Chunfeng Song
National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing 100190, China

ARTICLE INFO

Article history:
Received 15 May 2013
Received in revised form 17 July 2013
Accepted 15 September 2013
Communicated by D. Tao
Available online 9 November 2013

Keywords:
Object classification
Spatial modeling
Multiple pooling

ABSTRACT

Global spatial structure is an important factor for visual object recognition but has not attracted sufficient attention in recent studies. Especially, the problems of features' ambiguity and sensitivity to location change in the image space are not yet well solved. In this paper, we propose multiple spatial pooling (MSP) to address these problems. MSP models global spatial structure with multiple Gaussian distributions and then pools features according to the relations between features and Gaussian distributions. Such a process is further generalized into a unified framework, which formulates multiple pooling using matrix operation with structured sparsity. Experiments in terms of scene classification and object categorization demonstrate that MSP can enhance traditional algorithms with small extra computational cost.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Visual object recognition is one of the most fundamental problems in computer vision and pattern recognition, and plays an important role in many vision tasks, e.g., scene classification [1] and object categorization and retrieval [2]. It attracts much attention from different fields [3–6]. One of the main research streams in this field is local feature-based model, i.e., firstly extracting a set of local features from each image, and then describing them with various feature coding strategies. This model achieves the state-of-the-art performance on some databases and competitions on object recognition. With this model, various algorithms on feature extraction and feature coding have been developed, most of which focus on objects' appearance description. However, in this model, objects' global spatial structure in the image space is usually undervalued. A recent psychological study on recognizing jumbled images [7] argues that subjects' recognition capability will drop dramatically without global spatial information. In particular, this study experimentally shows that if an image is divided into 9, 16, and 64 blocks with disorganization, as illustrated in Fig. 1, the recognition accuracy will decrease from 80% to about 60%, 50% and 20% respectively.

In the real world, objects are organized with specific spatial rules, and humans can hardly perceive them without taking their spatial structure into account. If objects' spatial information is disturbed, the final visual perception will conflict with our experience, which firmly justifies the importance of global spatial structure for visual object recognition.

In this paper, inspired by a recent technique of multi-way pooling [8] (to be introduced in the next section), we propose a unified mathematical form which models multiple pooling as matrix operation with structured sparsity. Based on this generalized form, we develop multiple spatial pooling (MSP) to take advantage of global spatial structure. This process is described as the following four steps: (1) Firstly extract objects' local features as well as their spatial location information, and then apply the multiple Gaussian distributions to model the objects' spatial structure in the image space. After that, generate a codebook and encode extracted local features, and finally pool over features' coding responses according to the relations (e.g., the distance) between features and Gaussian distributions.

The proposed MSP scheme is simple but effective and efficient to model features' ambiguity in the image space and alleviate features' sensitivity to location changes induced by object shifting. Experimental analysis demonstrates that MSP can enhance traditional algorithms in various conditions with nearly ignorable computational cost.

The major contributions of this paper are summarized as follows.

- We propose multiple spatial pooling to model the ambiguity of features location and alleviate the sensitivity to object shifting.
- Based on the proposed model, we further discuss other pooling models, and develop a unified mathematical form of multiple pooling using matrix operation with structured sparsity.
- Evaluate different pooling algorithms. Via Extensive experimental comparison, some meaningful findings are obtained, which is useful for practical applications.

* Corresponding author.
E-mail address: yzhuang@nlpr.ia.ac.cn (Y. Huang).

0925-2312/S – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.09.037
The remainder of this paper is organized as follows: Section 2 introduces related work, including the algorithmic platform and recent studies on discovering spatial information. Section 3 proposes the unified form of multiple pooling, derives our scheme for spatial pooling, and discusses the relations between our scheme and related methods. Section 4 reports and analyzes experimental results, and Section 5 concludes this paper.

2. Related work

The BoF (bag-of-features) model [9] is chosen as the algorithmic platform in our work, because it is intuitive and easy to embed multiple pooling. Besides, the BoF model is one of the most prevalent and successful methods for object recognition. For example, this model and its extensions have dominated the PASCAL VOC competition [10] on image classification for years.

Generally, the BoF model consists of three major components:

1. **Feature extraction.** Image patches are sampled from each image with a fixed grid or by detectors. Then local features are extracted via calculating patches’ descriptors. Widely used descriptors include gradient based descriptors [11,12], texture based descriptors [13,14], shape based descriptors [15–19], and biological descriptors [20–24].

2. **Feature coding.** In this step, local features are encoded by a codebook (a set of codewords) to generate a coding matrix. The codebook is generated by clustering or dictionary learning over extracted local features. Each feature is described by one or multiple codewords, depending on the used coding strategy, e.g., voting-based coding [9,25], Saliency based coding [26,27], Fisher coding [28], and reconstruction-based coding [29].

3. **Feature pooling.** Traditional pooling in BoF is defined as integrating the responses on a codeword into one value. Widely used traditional pooling methods are average pooling and MAX pooling, and an in-depth analysis about pooling can be found in [30].

The original BoF model and many of its extensions ignore the role of features’ spatial information. To embed spatial information, some studies are developed, generally falling into two categorizes: local and global methods, introduced as follows.

The local spatial modeling mainly focuses on the feature and the codeword levels. For example, Boureau et al. [30] propose the macro-features, where local spatial information is captured via concatenating the representation of the sub-areas of image patches. Morikoka and Satoh [31] generate features by linking spatially neighboring patches’ descriptors in a pair-wise manner, and learn codewords by clustering over these features. Further, this work is combined with the proximity distribution kernel [32] to obtain compact representation and scale invariance [33]. Generally, local spatial modeling needs large computational cost, and is limited to describe spatial structure information, e.g., the spatial organization of objects’ parts.

The most widely used global spatial model is possibly the pyramid match kernels (PMK) [34]. In this model, an image is firstly partitioned into several regular blocks in a pyramid manner, and then local features are matched with different weights according to the pyramid. Further, PMK is used with equal weights for all levels in the pyramid, which is equivalent to applying BoF in each block respectively and then concatenating all BoF representation vectors. This strategy is called spatial pyramid matching (SPM) [35], which is simple, fast and effective, and has strong generalization ability over different databases.

Some new global spatial modeling techniques have been developed since the proposal of SPM. Harada et al. [36] optimized the weights of blocks to train a discriminative spatial pyramid. Wang et al. [37] developed a new strategy for blocks partition, which is similar to the one in the shape context descriptor [15]. Yang et al. [38] proposed a co-occurrence kernel in the matching step. These models improve SPM to some degree but need extra demands on databases. Krapac et al. [39] employed GMM to model the distribution of features belonging to each codeword. This work is combined with Fisher coding for saving memory cost, but takes very limited enhancement in accuracy. Jia et al. [40] constructed over-complete spatial blocks followed by block selection. This strategy adaptively learns the discriminative blocks and achieves good performance. However, its computational cost is huge.

Recently, Boureau et al. [8] propose a technique called multi-way local pooling, which is closely related with our method. Simply speaking, the feature space is divided into a number of groups via clustering over features’ representation. Then features’ coding responses are pooled in different groups separately. We call such a process original multiple pooling (OMP). It is beneficial for representing similar features with the same bases, which helps discover the global structure of the feature space. In the next section, we will provide an in-depth analysis on OMP within our proposed unified framework of multiple pooling.

3. Our method

In this section, we firstly propose a unified framework of multiple pooling, then describe MSP in detail, and finally compare MSP and other related methods.

3.1. Unified form of multiple pooling

The unified form of multiple pooling (MP) is embedded into the BoF model for explanation. As shown in Fig. 2, for each input image, feature extraction is employed to generate a feature matrix \( F = [f_1, f_2, \ldots, f_M] \in \mathbb{R}^{D \times N} \) with \( M \) \( D \)-dimensional features. Afterward, each feature is encoded by a codebook \( B = [b_1, b_2, \ldots, b_N] \in \mathbb{R}^{D \times N} \) with \( N \) \( D \)-dimensional codewords. After feature coding,
a coding matrix $\mathbf{Z} = [z_1, z_2, ..., z_M] \in \mathbb{R}^{N \times M}$ is produced. The coding matrix is sent to the component of multiple pooling to generate the final representation $\mathbf{V}$, which is used for classification. It should be noted that, the proposed framework of multiple pooling is not limited to the BoF model but suitable for any local feature-based models where local features are described by a set of bases.

Before introducing multiple pooling, we firstly define some variables. Let $\mathbf{E} = [e_1, e_2, ..., e_M] \in \mathbb{R}^{L \times M}$, where $e_m = (x_m, y_m)$ denotes the spatial location of the $m$th feature in the image space. $x_m$ and $y_m$ are normalized into $[0, 1]$ according to the width and the height of images. Therefore, a feature $f_m$ can be described as a structure $(z_m, e_m)$ with respect to its representation in the feature space and the image space. For clarity, we elaborate the meanings of important notations in Table 1.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meanings</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbf{B} = [b_1, b_2, ..., b_N]$</td>
<td>Codebook consisting of $N$ codewords</td>
</tr>
<tr>
<td>$C$</td>
<td>Number of pooling channels</td>
</tr>
<tr>
<td>$D$</td>
<td>Dimensionality of a local feature</td>
</tr>
<tr>
<td>$\mathbf{E} = [e_1, e_2, ..., e_M]$</td>
<td>Spatial locations of $M$ local features</td>
</tr>
<tr>
<td>$\mathbf{F} = [f_1, f_2, ..., f_M]$</td>
<td>Feature matrix consisting of $M$ local features</td>
</tr>
<tr>
<td>$M$</td>
<td>Number of local features</td>
</tr>
<tr>
<td>$N$</td>
<td>Codebook size</td>
</tr>
<tr>
<td>$\mathbf{P}$</td>
<td>Pooling matrix</td>
</tr>
<tr>
<td>$\mathbf{U}$</td>
<td>Matrix after multiple pooling</td>
</tr>
<tr>
<td>$\mathbf{V}$</td>
<td>Final vector representation</td>
</tr>
<tr>
<td>$\mathbf{Z}$</td>
<td>Coding matrix</td>
</tr>
</tbody>
</table>

The core idea behind multiple pooling is to find a rule to group features. Since features in the same group are similar, pooling features in each group separately is beneficial to representing similar features with the same bases. We propose that different kinds of multiple pooling can be implemented by multiplying the coding matrix with a pooling matrix $\mathbf{P} \in \mathbb{R}^{M \times C}$ with structured sparsity, where $C$ is the number of pooling channels, i.e., the times of using pooling. This process is formulated as

$$\mathbf{U} = \mathbf{Z} \cdot \mathbf{P}. \tag{1}$$

$$\mathbf{P} = \phi(\mathbf{Z}, \mathbf{E}, \theta), \tag{2}$$

where $\phi$ is a function of generating the pooling matrix, with a parameter $\theta$ reflecting the rule of how to group features.

The final vector-based representation of an image can be represented as

$$\mathbf{V} = [\mathbf{U}(.; 1)^T, \mathbf{U}(.; 2)^T, ..., \mathbf{U}(.; C)^T]^T. \tag{3}$$

The pooling matrix determines the difference of various techniques of multiple pooling. For example, OMP [8] divides the feature space into a number of sub-areas via applying K-means clustering over features’ representation. The rule here corresponds to the K-means clustering, and $\theta$ denotes the cluster centers. As OMP only involves the feature space, the corresponding pooling matrix can be re-written as

$$\mathbf{P} = \phi(\mathbf{Z}, \theta). \tag{4}$$

Further, it can be formulated as

$$\phi(z_m, \theta_c) = \begin{cases} 1 & \text{if } z_m \in \text{cluster } \theta_c; \\ 0 & \text{otherwise} \end{cases}. \tag{5}$$

where $\theta = (\theta_1, \theta_2, ..., \theta_C)$ here denotes the cluster centers of features’ representation in the feature space. Fig. 3 shows a toy example of the pooling matrix with structured sparsity.

### 3.2. Multiple spatial pooling

Compared with OMP, our proposed MSP has two major differences. Firstly, the task of MSP and OMP is different. OMP is designed for exploiting the structure of the feature space, while MSP is proposed for modeling global spatial structure. Therefore, we conduct multiple pooling in the image space instead of the feature space. The corresponding pooling matrix in MSP can be described as

$$\mathbf{P} = \phi(\mathbf{E}, \theta). \tag{6}$$

Secondly, the ways of grouping features are different. In MSP, we employ multiple Gaussian distributions rather than the K-means clustering used in OMP. In particular, each feature is pooled multiple times with different weights, depending on the relations between the feature and Gaussian distributions. The pooling matrix in MSP can be further formulated as

$$\phi(e_m, e_n) = \begin{cases} \mathcal{G}(d(e_m, e_n)) & \text{if } d(e_m, e_n) \leq T; \\ 0 & \text{otherwise} \end{cases}, \tag{7}$$

$$d(e_m, e_n) = \|e_m - e_n\|_2. \tag{8}$$

$$\mathcal{G}(d) = \exp(-d^2/\sigma), \tag{9}$$

where $\mathcal{G}$ denotes the Gaussian function, $\sigma$ is set to 0.1 here according to cross validation, and $e_n$ here is the spatial location of the $n$th Gaussian center. For the simplest case, $\theta$ can be set as being equally distributed in the image space (see the crosses in Fig. 4). Intuitively, this is a good choice of the Gaussian centers, which is adopted in our experiments. In Section 5, we will discuss other possible strategies of solving $\theta$. $T$ is a threshold, defined as the distance between two centers. The threshold also determines the valid regions used to filter features which are too far away from $\theta_i$. For example, in Fig. 4, the valid regions on different levels are depicted by the dashed curve/circles.
The pooling matrices in OMP and MSP are of structured sparsity. Their difference lies in the representation of structure in the pooling matrices. On the one hand, the adopted multiple Gaussian distributions describe the structure more accurately than K-means clustering. On the other hand, compared with the pooling matrix in OMP (see Eq. (5)), the Gaussian function-based pooling matrix in MSP (see Eq. (7)) contains richer information, e.g., the similarity between features and Gaussian centers.

### 3.3. Relations with other methods

It is not difficult to identify that the direct extension of OMP, from the feature space to the image space, is the famous SPM [35]. It divides an image into a number of blocks on different levels, and then pooling is performed in each block respectively, setting the centers of blocks as the pooling clusters. With the proposed unified framework, it is easy to derive the mathematic formulation of SPM as

\[
\phi(x_m, y_m, \theta_c) = \begin{cases} 
1 & \text{if } d_{m,x} \leq \frac{1}{2L} \text{ and } d_{m,y} \leq \frac{1}{2L} \\
0 & \text{otherwise}
\end{cases}
\] (10)

\[
d_{m,x} = |x_m - x_{\theta_c}|, \quad d_{m,y} = |y_m - y_{\theta_c}|,
\] (11)

where \(L\) is illustrated in Fig. 5, and \(\theta_c\) is the \(c\)th block center.

MSP is superior to SPM in two aspects. Firstly, it is difficult for SPM to model the ambiguity of features’ spatial information. For example, if a feature locates in the middle of two blocks, it is difficult to judge which block it belongs to. Secondly, when the object shifts a little, the ownership of some features tends to change from one bin to another, inducing big change in the final representation after pooling. For example, it is easy for features around the boundary of two blocks to shift from one block to another.

MSP can well address such problems if we use the block centers in SPM as the Gaussian centers in MSP. Accordingly, the relations between features and groups are calculated by the Gaussian function, which describes the ownership of features so as to well model the ambiguity of features’ locations. Meanwhile, each feature in MSP can be pooled multiple times in each level, because the valid regions of Gaussian distributions overlap each other in the image space (see Fig. 4). As a result, the sensitivity to object shifting is alleviated in MSP. Taking Fig. 6 as an example. It is ambiguous to assign \(f_1\) to block 2 or block 3. With the Gaussian function, the weights of \(f_1\) in both block 2 and block 3 become small, which indicates that \(f_1\) is an unstable feature in representing the global spatial structure. In addition, when \(f_2\) shifts from block 5 to block 3, it still contributes to the pooling process in block 5, which avoids strong influence in the final representation.

### 4. Experimental results

To evaluate our proposed method, an empirical study is conducted in this section. Firstly, we introduce experimental setup and databases, then provide basic results and discussion, and finally compare our method with related methods, followed by efficiency analysis.

#### 4.1. Experimental setup and databases

The experimental setup is introduced in accordance with the framework illustrated in Fig. 2. For each image, 128 dimensional SIFT descriptors [11] are densely extracted on a grid with a step of three pixels on three scales: 16 x 16, 24 x 24 and 32 x 32. Codebook is obtained by the K-means clustering. In feature coding, hard voting (HV) [9] and super-vector coding (SVC) [41] are chosen. The parameters of multiple pooling will be detailed in each experiment. For normalization, we \(l_2\)-normalizes the square root of the responses. The final image representation is fed to linear SVM for training and testing wherein the penalty coefficient is determined via cross validation.

All experimental comparisons are based on the same platform implemented by ourselves. Such comparison is fair because it does not bias any methods with different implementation details. The small discrepancy between our results and those reported by the original literature is possibly induced by different technical tricks, e.g., codebook generation, SVM parameters and image preprocessing.
We choose the 15Scenes dataset for scene classification and the Caltech101 database for object classification. These two databases are the ones used in Boureau et al.’s work [8] to evaluate the effectiveness of OMP. In the evaluation process, we follow the experimental settings proposed in [35]. That is, on 15Scenes, from each category, 100 images are randomly picked out for training, and the remaining images are used for testing. On Caltech101, from each category, 30 images are randomly chosen for training, and at most 50 images randomly chosen from the rest are used for testing. For both the two databases, we repeat each experiment 10 times and report the average classification accuracy and the standard deviation.

As analyzed in Section 3.3, the direct extension of OMP from the feature space to the image space is the SPM algorithm. Therefore, we mainly compare our method with SPM in this section. Usually, SPM is implemented with three levels, i.e., 1/\sqrt{2}, 2/\sqrt{4}, and 4/\sqrt{16}. More implementation details can be found in [35]. The difference between MSP and SPM is only the pooling component (Eq. (10) vs. Eq. (7)).

4.2. Accuracy analysis

In this section, we analyze the experimental results of our MSP scheme introduced in Section 3.2.

As shown in Eq. (7), the main parameter of our scheme is \( T \), i.e., the distance between two Gaussian centers, which is determined by the number of Gaussian distributions under the assumption of uniform distribution (see Fig. 4). Therefore, we test the influence of the number of Gaussian distributions (\( C \)) and report the experimental results in Fig. 7.

It should be noted that the baseline algorithm here is almost the same as our method except that the component of MSP is removed. From the experimental results in Fig. 7, we can draw the following conclusions:

- The proposed MSP can enhance the baseline algorithm under all conditions in terms of the number of Gaussian distributions, the codebook size and the databases, which firmly justifies the effectiveness of MSP.
- The performance is generally better when using more Gaussian distributions. This is consistent with our intuition that more cluster centers can model more complex spatial structure.
- The superiority of MSP over the baseline is larger when adopting a small codebook size. This is caused by two possible reasons. On the one hand, the spatial distribution of codewords is more obvious with a small number of codewords. In this case, each codeword will on average occupy a larger area in the image space. On the other hand, the feature space performs relatively worse with a small codebook size. In this case, it is easier to enhance the baseline algorithm via taking advantage of the global spatial structure.
- The enhancement by MSP is more obvious on Caltech101. For example, when \( C = 16 \), with 128 codewords, the enhancement is 15.6% on 15Scenes and 22.8% on Caltech101. With 8192 codewords, the enhancement is 2.8% on 15Scenes and 13.3% on Caltech101. This is probably because that the images on the Caltech101 database are dealt with alignment, and thus the spatial structure information is more stable.

Further, we compare our method with other related ones. We firstly show the experimental difference between SPM and MSP in detail, i.e., in each block and each level, and then compare their final results.

Fig. 8 provides the performance comparison of each block. In this experiment, we use 21 block centers of SPM (see Fig. 5) as the centers of the multiple Gaussian distributions in MSP, and show

![Fig. 6. An illustration showing the limitations of SPM.](image)

![Fig. 8. Comparison between SPM and MSP in each block respectively.](image)
the classification accuracy by each block separately. From Fig. 8, it is not difficult to obtain the following conclusions.

- In most blocks, MSP greatly outperforms SPM, from 6.5% (block 6) to 13.3% (block 16). Moreover, it is a little surprising that some blocks from level 3 in MSP (e.g., block 15 and block 16) even outperform the blocks from level 2 in SPM.
- The performance of SPM and MSP is of little difference in block 1. This is because the first column of their pooling matrices (corresponding to block 1) is almost the same. The slight difference is caused by randomly choosing images for training.
- In the third level, blocks 11, 12, 15 and 16 perform better than other blocks, whether in MSP or SPM. These four blocks correspond to the central areas of an image (see Fig. 5). This result indicates that the central region of images is more informative. This is probably because most photos are taken with a prior that objects of interest are generally in the center of photos.

It may be argued that the superiority of MSP over SPM for each block results from the use of more features in pooling. To eliminate the influence of this factor, we design additional experiment which compares the performance of SPM and MSP on each level. The number of features in each level is the same for both MSP and SPM. As illustrated in Fig. 9, MSP also outperforms SPM in each level and the cases of level combination.

As the only difference of SPM and MSP lies in the pooling matrix, these two experiments demonstrate that the pooling matrix in MSP is designed more reasonably than that in SPM.

Finally, in Table 2, we list the overall results of the original BoF, SPM and MSP on 15Scenes and Caltech101. HV and SVC denote hard voting and super-vector coding [41]. It can be seen from Table 2 that SPM and MSP both perform much better than the original BOF model, which verifies the importance of spatial information and the effectiveness of multiple pooling. In addition, MSP always outperforms SPM under all tested conditions. This improvement is more obvious on the Caltech101 databases. The reason has been explained in Section 4.2.

Although our method does not beat the state-of-the-art ones using other techniques, the experimental results on these two databases conform to our expectation that MSP should enhance SPM due to more reasonable exploration of objects’ global spatial structure. We believe that after combining with other advanced techniques, there is space for MSP to enhance the performance further, e.g., using the improved Fisher coding (IFK) [28]. In addition, it should be emphasized that the enhancement of performance by MSP is achieved with very little extra computational cost, to be analyzed in the next subsection.

4.3. Efficiency analysis

In this section, we analyze the computational complexity for multiple pooling in SPM and MSP. For fair comparison, we assume that SPM and MSP use the same number of groups, i.e., C. That is, the number of blocks in SPM is equal to the number of Gaussian distributions in MSP.

The computational cost of pooling for each feature in SPM is \(O(C) + O(C)\). The first term, corresponding to Eq. (11), is the cost of calculating distance in the 2-D image space. The second term is in accordance with the comparison operation in Eq. (10).

The cost of pooling for each feature in MSP is \(O(C) + O(C) + O(C)\). The first term, corresponding to Eq. (8), is similar to that in SPM. The second term is for the comparison operation in Eq. (7). The third term is the computational cost of calculating the Gaussian function in Eq. (7).

It should be noted that the computational cost for multiple pooling in SPM and MSP is much less than that of feature coding. Take HV as an example. The main operations of encoding a feature with HV are: (1) calculating the distance between this feature and \(N\) codewords in the \(D\) dimensional feature space, and (2) ranking the distance. The computational cost corresponding to these two parts are \(O(ND)\) and \(O(N\log N)\) respectively. Usually \(N\) and \(D\) is much larger than \(C\), and thus the increased time induced by multiple pooling is almost ignorable. For example, when \(C = 21\) and \(N = 1024\), the time cost of feature coding for an image is about 0.02 s, and the component of MSP costs less than 0.001 s.

5. Conclusion and future work

In this paper, we have proposed MSP to capture objects’ global spatial structure. In particular, we have constructed the pooling matrix by employing multiple Gaussian distributions, which helps model the ambiguity of features’ location and alleviate the sensitivity to object shifting. Another contribution of this paper is the analysis of the unified mathematical form of multiple pooling. Due to its flexibility and generalization ability, OMP and SPM can be considered as special cases under this form, and also MSP can be easily designed to solve the two problems mentioned above.
The open problem is the rule of grouping features, i.e., the decision of $\varphi$ and $\theta$ in Eq. (7). Currently, we simply adopt the uniform distribution to generate the Gaussian centers (see Fig. 4). It is potential to study more discriminate schemes to learn the rule, e.g., considering $\theta$ as a latent variable for iteratively optimization, or employing the theory of structured sparsity learning for an in-depth structure discovery in the pooling matrix.

Acknowledgement

This work is jointly supported by the National Natural Science Foundation of China (61175003, 61135002, 61203252), Hundred Talents Program of CAS, and Tsinghua National Laboratory for Information Science and Technology Cross-discipline Foundation.

References

[38] Y. Yang, S. Newsam, Spatial pyramid co-occurrence for image classification, 2011.