LEARNING WEIGHTED GEOMETRIC POOLING FOR IMAGE CLASSIFICATION

Chaoqun Weng, Hongxing Wang, Junsong Yuan

Nanyang Technological University

ABSTRACT

Local feature extraction, coding, pooling, and image classification are the four typical steps for the state-of-the-art visual recognition systems. Unlike previous work that treats feature pooling and image classification as separated steps, we propose to jointly learn the geometric pooling and image classifier by support tensor machine. Inspired by previous work of spatial pyramid matching and receptive field learning, we also propose spatial pyramid geometric pooling, receptive field geometric pooling and random partition geometric pooling approaches to exploit the spatial pooling neighborhood structure to further boost the classification performance. Experiments on 15-scene dataset validate the advantages of our proposed algorithms.

***Index Terms*—** jointly geometric pooling, image classification, support tensor machine

1. INTRODUCTION

Modern image classification algorithms often adopt a four-step pipeline, i.e., local descriptor extraction and coding, spatial pooling of local descriptor codes and the final classifier training. As illustrated in the top row of Fig. 1, many algorithms first extract hand-crafted SIFT [1] or HOG [2] local descriptors from images, upon which a codebook is trained and the descriptors are encoded, by k-means or sparse coding [3]; after that, global image representations are formed by spatially pooling (e.g., average-pooling and max-pooling) the local descriptor codes [4]; finally, a linear [3] or non-linear [5] classifier is trained over the pooled image representations. State-of-the-art performances have been achieved by such four-step pipeline methods on several benchmark image classification datasets, such as Scene-15, Caltech-101 and Caltech-256 [3, 5–8].

Many previous methods have focused on the local descriptor coding step, such as sparse coding [3, 9], locality constrained linear coding [10], soft assignment coding [11] and the spatial locality-aware sparse coding [12], among which the target is to learn more representative local descriptor codes by exploiting the feature space of the local descriptors. Unlike the local descriptor coding step, spatial pooling step aims at obtaining more discriminative image representations for the classification task by exploiting the spatial distributions of local descriptors. The spatial pooling technique is usually an operation over the local descriptor codes from a sub-region of the image, such as the sum-pooling, average-pooling [5, 13, 14] and max-pooling [3, 8, 15]. As discussed in [16], max-pooling produces more discriminative representations if soft coding methods upon local descriptors are used, while average-pooling on the other hand, works better if hard quantization method is applied. Recently, [6] has proposed to learn a weighted ℓ_p-norm spatial pooling function tailored for the class-specific feature spatial distribution and has boosted the discriminating capability of the resultant features for image classification.

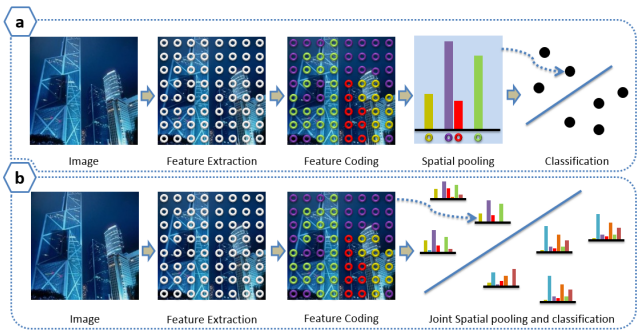

Fig. 1. Illustration of image classification procedures: (a) four-step pipeline: separated spatial pooling and classifier training; (b) three-step pipeline: jointly learning spatial pooling and image classifier (Ours).

However in spite of the great successes of previous spatial pooling work, there still exist many limitations. First, many previous methods usually apply spatial pooling techniques to reduce computational complexity or bring translation invariance, but is not necessarily an optimal choice for classification. Second, the class-specific geometric distribution information of local descriptors is often ignored during the spatial pooling process. Third, the importance of the spatial pooling neighborhood structure has not been well explored in many previous spatial pooling methods. The successes of spatial pyramid matching [5] and receptive field learning [7] have illustrated the effectiveness of using spatial pyramid structure and adaptive receptive field structure for image classification task. Therefore it is a sensible idea to utilize such spatial

structure in the spatial pooling step.

This paper contributes to addressing the above issues. First, as shown in the bottom row of Fig. 1, we propose to jointly learn the spatial pooling and the final image classifier by support tensor machine [17], to exploit the class-specific local descriptor geometric information and thus obtain more discriminative image classifier. Besides, inspired by previous work of spatial pyramid matching and receptive field learning, we also propose spatial pyramid geometric pooling, receptive field geometric pooling and random partition geometric pooling methods, to exploit the spatial pooling neighborhood structure and thus further boost the classification performance. Experiments on 15-scene dataset validate the advantages of our proposed algorithms.

2. METHODOLOGY

2.1. Image Classification Procedures

The traditional four-step pipeline image classification procedure is shown in the top row of Fig. 1. The first and second steps include local descriptors (such as dense SIFT or HOG descriptors) extraction and coding. The third step is the spatial pooling that forms the global representations of the original images. The fourth step is to train a classifier upon the spatial pooled feature vectors.

Formally, an image is represented by a set of local descriptors $\mathbf{D} = [\mathbf{d_1}, \mathbf{d_2}, \cdots, \mathbf{d_m}] \in \mathbb{R}^{d \times m}$ where m is the number of local descriptors and d is the dimension of each local descriptor. Given a codebook $\mathbf{B} \in \mathbb{R}^{d \times k}$ where k is the codebook size, the sparse coding representation $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_m}] \in \mathbb{R}^{k \times m}$ of the descriptor set $\mathbf{D}$ can be calculated as follows,

$$\arg\min_{\mathbf{X}} \|\mathbf{D} - \mathbf{B}\mathbf{X}\|_F^2 + \lambda \|\mathbf{X}\|_{\ell_1} \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix and $\|\cdot\|_{\ell_1}$ is the ℓ_1 norm.

After the coding step, we define a set of sub-regions of the image as $R = \{R_1, R_2, \cdots, R_n\}$ (e.g., R can be the whole image, or the spatial pyramid cells [5]), and then use spatial pooling techniques such as average-pooling in Eq. 2 or max-pooling in Eq. 3 to aggregate the encoded local descriptors into the feature vector $\mathbf{z_j} \in \mathbb{R}^k$ for sub-region R_j,

$$\mathbf{z_j} = \frac{1}{|R_j|} \sum_{i \in R_j} \mathbf{x_i} \tag{2}$$

$$\mathbf{z_j} = \max_{i \in R_j} (\mathbf{x_i}) \tag{3}$$

where the max operation is element-wise operation. Then we can concatenate all the sub-regions to form the global image representation $\mathbf{z} \in \mathbb{R}^{nk}$, as follows:

$$\mathbf{z} = [\mathbf{z_1^T}, \mathbf{z_2^T}, \cdots, \mathbf{z_n^T}]^T \tag{4}$$

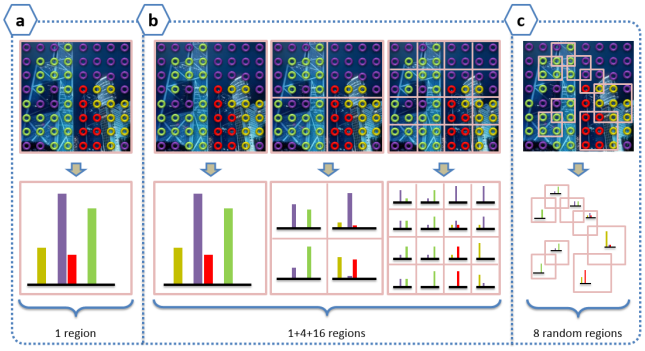

Fig. 2. Illustration of different geometric pooling methods: (a) Dense grid; (b) Spatial pyramid; (c) Receptive field.

Finally, training a linear SVM classifier is favorable due to its computation efficiency:

$$\arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_i \max(0, 1 - y_i f(\mathbf{z_{(i)}})) \tag{5}$$

$$f(\mathbf{z}) = \mathbf{w^T z} \tag{6}$$

where $\mathbf{z_{(i)}}$ is the concatenated feature vector for image i, $y_i \in \{+1, -1\}$ is the corresponding label for image i and $f(\cdot)$ is the linear SVM classifier.

In the following we will introduce our three-step pipeline image classification procedure. The local descriptor extraction and coding are the same as above. Thus we will focus on how to learn the geometric pooling and the image classifier jointly, as illustrated in the bottom row of Fig. 1.

2.2. Dense Grid Geometric Pooling

Let us first consider the dense grid setting for geometric pooling. Formally, local descriptors are extracted densely in a grid-based fashion, as illustrated in the left column of Fig. 2. After the coding step, we represent each image by the codes matrix $\mathbf{X} \in \mathbb{R}^{k \times m}$.

We propose the **weighted geometric pooling** for the dense grid local descriptors, as follows,

$$\mathbf{z} = \sum_i v_i \mathbf{x_i} = \mathbf{X}\mathbf{v} \tag{7}$$

where $\mathbf{v} = [v_1, v_2, \cdots, v_m]^T \in \mathbb{R}^m$ is the geometric pooling coefficient weight and $\mathbf{z} \in \mathbb{R}^k$ is the resultant geometric pooled feature vector for the image. Note that average-pooling is a special case of our weighted geometric pooling if $\mathbf{v} = \mathbf{1}$.

Once we have obtained $\mathbf{z}$, we are interested in training a linear SVM classifier similar to Eq. 6:

$$f(\mathbf{X}) = \mathbf{u^T z} = \mathbf{u^T X v}$$

where $\mathbf{u} = [u_1, u_2, \cdots, u_k]^T \in \mathbb{R}^k$ is the linear SVM weight.

We follow the work on support vector machines and define the empirical loss of classifier $f(\cdot)$ as the sum of hinge losses over a collection of training images:

$$\sum_i \max(0, 1 - y_i f(\mathbf{X_{(i)}}))$$

where $\mathbf{X_{(i)}}$ is the codes matrix for image i. Since focusing solely on the empirical loss may result in over-fitting, we impose an ℓ_2 regularization penalty on the matrix $\mathbf{uv^T}$, as used in *support tensor machine* [17]:

$$\arg\min_{\mathbf{u},\mathbf{v}} \frac{1}{2}\|\mathbf{uv^T}\|_F^2 + C\sum_i \max(0, 1 - y_i f(\mathbf{X_{(i)}})) \quad (8)$$

$$f(\mathbf{X}) = \mathbf{u^T X v} \quad (9)$$

It is worth noting that Eq. 9 combines the weighted geometric pooling and the linear classifier training in a unified framework. Besides, the regularization term $\|\mathbf{uv^T}\|_F^2$ in Eq. 8 is analogous to the regularization term $\|\mathbf{w}\|^2$ in Eq. 5, if we flatten the image data matrix $\mathbf{X}$ and the regularization matrix $\mathbf{uv^T}$ into vectors.

We use alternate projection method to iteratively optimize the objective function over $\mathbf{u}$ and $\mathbf{v}$: when $\mathbf{v}$ is fixed, training $\mathbf{u}$ becomes a standard SVM problem; similarly once $\mathbf{u}$ is fixed, updating $\mathbf{v}$ also reduces to a standard SVM problem. Intuitively, training $\mathbf{u}$ can be viewed as the conventional first-pooling-then-training-svm image classification process, given the feature vector $\mathbf{Xv}$ that is geometric-pooled by $\mathbf{v}$, while on the other hand updating $\mathbf{v}$ can be viewed as another SVM classifier training process, given the feature vector $\mathbf{u^T X}$ that contains sub-region scores predicted by $\mathbf{u}$. Since at each iteration we can find the global solution to the quadratic programming SVM problem (which means the objective will decrease or at least remain fixed), the objective will gradually converge to a local minima [17].

2.3. Spatial Pyramid Geometric Pooling

To incorporate the idea of spatial pyramid matching [5], we continue the previous discussion on dense grid geometric pooling and propose to explore the spatial pyramid structure for geometric pooling.

Assume that ℓ is the number of spatial pyramid levels, we then define a set of spatial pyramid sub-regions $R = \{R_1, R_2, \cdots, R_n\}$ of size $n = \sum_{i=0}^{\ell} 4^i = \frac{1}{3}\left(4^{\ell+1} - 1\right)$, as illustrated in the middle column of Fig. 2. After that, the spatial pyramid data matrix $\mathbf{X'} \in \mathbb{R}^{nk \times m}$ is composed by local descriptor codes that falls within each sub-region, as follows,

$$\mathbf{X'} = [\mathbf{XI_1}, \mathbf{XI_2}, \cdots, \mathbf{XI_n}]^T \quad (10)$$

where each $\mathbf{I_j} \in \mathbb{R}^{m \times m}$ is a diagonal indicator matrix with non-zero element indicating whether the corresponding local descriptor is located in sub-region R_j.

In order to substitute $\mathbf{X}$ by $\mathbf{X'}$ in Eq. 9, we further define the corresponding classifier weight vector $\mathbf{u'} = [\mathbf{u_1}, \mathbf{u_2}, \cdots, \mathbf{u_n}]^T \in \mathbb{R}^{nk}$ with each $\mathbf{u_j} \in \mathbb{R}^k$ denoting the classifier weight vector for sub-region R_j. Therefore the final classifier in Eq. 9 becomes:

$$f(\mathbf{X'}) = \mathbf{u'^T X' v} \quad (11)$$

2.4. Receptive Field Geometric Pooling

Inspired by receptive field learning method [7], we also propose a receptive field geometric pooling approach to further exploit the spatial structure for pooling, as illustrated in the right column of Fig. 2. The key idea is to learn adaptive sub-regions (receptive fields) for pooling instead of using grid structure of spatial pyramid.

Formally, we randomly generate a set of sub-regions in the image as $R = \{R_1, R_2, \cdots, R_n\}$. Then we apply max-pooling in Eq. 3 to obtain the feature vector $\mathbf{z_j} \in \mathbb{R}^k$ for each sub-region R_j. Then the receptive field data matrix $\mathbf{X'} \in \mathbb{R}^{k \times n}$ is defined by concatenating all the sub-regions,

$$\mathbf{X'} = [\mathbf{z_1}, \mathbf{z_2}, \cdots, \mathbf{z_n}] \quad (12)$$

After that, we define the receptive field coefficient weight $\mathbf{v'} = [v_1, v_2, \cdots, v_n]^T \in \mathbb{R}^n$, and the final classifier in Eq. 9 becomes:

$$f(\mathbf{X'}) = \mathbf{u^T X' v'} \quad (13)$$

2.5. Random Partition Geometric Pooling

It is worth noting that spatial pyramid geometric pooling considers using spatial pyramid classifier weight $\mathbf{u'}$ in Eq. 11, while the local descriptor coefficient weight $\mathbf{v}$ is left unchanged from Eq. 9. Moreover receptive field geometric pooling considers using receptive field coefficient weight $\mathbf{v'}$ in Eq. 13, while the classifier weight $\mathbf{u}$ remains unchanged from Eq. 9. In this section we propose the random partition geometric pooling method that uses both spatial pyramid classifier weight $\mathbf{u'}$ and receptive field coefficient weight $\mathbf{v'}$.

Formally, we define a spatial pyramid of ℓ levels and each level i is randomly partitioned into non-overlapping $2^i \times 2^i$ sub-regions (it is like the spatial pyramid matching method, however it is done by random partition instead of the grid partition) [18]. By doing so, we can obtain a set of sub-regions $R = \{R_{ij} | \forall i \in \{1, \cdots, n\}, \forall j \in \{1, \cdots, r\}\}$ where $n = \frac{1}{3}\left(4^{\ell+1} - 1\right)$ (See Section 2.3). Then we apply max-pooling in Eq. 3 to get the corresponding feature vector $\mathbf{z_{ij}} \in \mathbb{R}^k$ for each sub-region R_{ij}. After that, we define the random partition data matrix $\mathbf{X'} \in \mathbb{R}^{nk \times r}$ as follows,

$$\mathbf{X'} = \begin{bmatrix} \mathbf{z_{11}} & \cdots & \mathbf{z_{1r}} \\ \vdots & \ddots & \vdots \\ \mathbf{z_{n1}} & \cdots & \mathbf{z_{nr}} \end{bmatrix} \quad (14)$$

Then we further define the spatial pyramid classifier weight vector $\mathbf{u'} = [\mathbf{u_1}, \mathbf{u_2}, \cdots, \mathbf{u_n}]^T \in \mathbb{R}^{nk}$ and the receptive field coefficient weight vector $\mathbf{v'} = [v_1, v_2, \cdots, v_r]^T \in \mathbb{R}^r$, and the final classifier in Eq. 9 becomes:

$$f(\mathbf{X}') = \mathbf{u'^T X' v'} \tag{15}$$

Note that the conventional spatial pyramid matching with max-pooling method is a special case of our proposed random partition geometric pooling method, if we set $r = 1$ and use grid partition.

3. EXPERIMENT

In the experiments, we apply our algorithms to image classification problem on the 15-scene dataset. We first resize the images into 256×256 resolution, and then extract dense SIFT descriptors from 16×16 pixel patches with 6 stepsize pixels. After that, we use standard k-means ($k = 1024$) to train the codebook and apply sparse coding method to encode the dense SIFT features, as done in [3]. In the spatial pyramid experiments, we use three levels spatial pyramid, i.e., the 1×1, 2×2, 4×4 grid structures. Following the same settings in [5], we use 100 images per class for training and the rest for testing. We run our experiments 10 times and report the average accuracy in Table. 1.

From the table we can see that, in the dense grid setting our proposed weighted geometric pooling algorithm outperforms both max-pooling methods (by 2.3%) and average-pooling (by 1.6%). In the spatial pyramid setting, geometric pooling still outperforms average-pooling (by 3.7%), however interestingly the max-pooling method achieves significantly higher accuracy than average-pooling (by 12.1%) and our geometric pooling (by 7.6%). These observations have suggested that (1) our geometric pooling method can consistently outperform average-pooling since our method can iteratively optimize over the $\mathbf{u}$ and $\mathbf{v}$ to find more discriminative geometric pooling weight $\mathbf{v}$; (2) the non-linear max-pooling operation suits better with the spatial pyramid structure than the linear pooling methods, e.g., average-pooling and weighted geometric pooling.

Inspired by the latter observation, we conduct the random partition geometric pooling experiment, in which max-pooling is applied within each random partitioned receptive fields and the spatial pyramid structure is also utilized, as discussed in Section 2.5. In this experiment, we randomly partitioned the image for $r = 100$ times on the spatial pyramid of $\ell = 3$ levels.

We first report the performance of receptive field geometric pooling as 71.96% (it is also the performance of random partition geometric pooling with $\ell = 1$ level). Note that this result has improved the performance of dense grid geometric pooling (67.07%) by 4.9%, which demonstrates that using non-linear max-pooled receptive fields are more effective for classification than using dense grid local descriptors directly. Then using spatial pyramid structure with $\ell = 3$ levels upon the random partitioned receptive fields, we have achieved accuracy 81.05%, which has improved from receptive field geometric pooling method (71.96%) by 9.1% and has beaten up spatial pyramid matching with max-pooling method (78.77%) by 2.3%. This result validates the effectiveness of our proposed random partition geometric pooling method, which uses both spatial pyramid structure and weighted geometric pooling technique upon the random partitioned receptive fields. We notice that [6] reported 83.2% accuracy, however it adopted more complex ℓ_p-norm pooling, while we only use a simpler weighted geometric pooling method based on the max-pooled sub-regions.

In summation, the experiments on 15-scene dataset justify the advantages of our proposed algorithms: (1) jointly learning the spatial pooling and the image classifier outperforms the separated method; (2) fusing different spatial pooling neighborhood structure (e.g., the spatial pyramid structure and the receptive field structure) outperforms the existing spatial pooling methods.

Table 1. Accuracy results on the 15-scene dataset.

Algorithm	Testing Accuracy (%)
Linear SPM [3]	65.32 ± 1.02
Kernel SPM [5]	81.40 ± 0.50
Kernel Codebook [19]	76.67 ± 0.39
Sparse Coding SPM [3]	80.28 ± 0.93
Locality Linear Coding [10]	79.24
Geometric ℓ_p-norm pooling [6]	$\mathbf{83.20}$
Dense Grid + Avg-Pooling	65.46 ± 0.94
Dense Grid + Max-Pooling	64.72 ± 0.49
Dense Grid + Geo-Pooling (Ours)	67.07 ± 0.53
Spatial Pyramid + Avg-Pooling	66.63 ± 0.71
Spatial Pyramid + Max-Pooling	78.77 ± 0.39
Spatial Pyramid + Geo-Pooling (Ours)	70.39 ± 0.73
Receptive Field + Geo-Pooling (Ours)	71.96 ± 0.65
Random Partition + Geo-Pooling (Ours)	$\mathbf{81.05 \pm 0.83}$

4. CONCLUSION

We propose to learn weighted geometric pooling by support tensor machine for image classification. Our proposed joint learning of spatial pooling and image classification can improve the image recognition performance compared with existing approaches that treat these two steps separately. Inspired by previous work of spatial pyramid matching and receptive field learning, we also propose spatial pyramid geometric pooling, receptive field geometric pooling and random partition geometric pooling methods to further boost the classification performance. Experiments on 15-scene dataset validate the advantages of our proposed algorithms.

References

[1] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, 2004.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *CVPR*, 2005.

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

[4] Y.L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," *ICML*, 2010.

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

[6] J. Feng, B. Ni, Q. Tian, and S. Yan, "Geometric lp-norm feature pooling for image classification," *CVPR*, 2011.

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

[8] L. Xie, Q. Tian, and B. Zhang, "Spatial pooling of heterogeneous features for image applications," *ACMMM*, 2012.

[9] R. Raina, A. Battle, H. Lee, B. Packer, and A.Y. Ng, "Self-taught learning: transfer learning from unlabeled data," *ICML*, 2007.

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

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

[12] J. Wang, J. Yuan, Z. Chen, and Y. Wu, "Spatial locality-aware sparse coding and dictionary learning," *ACML*, 2012.

[13] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," *ICCV*, 2003.

[14] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *IJCV*, 2007.

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

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

[17] D. Tao, X. Li, X. Wu, W. Hu, and S.J. Maybank, "Supervised tensor learning," *Knowledge and Information Systems*, 2007.

[18] Y. Jiang, J. Yuan, and G. Yu, "Randomized spatial partition for scene recognition," *ECCV*, 2012.

[19] J. van Gemert, J.M. Geusebroek, C. Veenman, and A. Smeulders, "Kernel codebooks for scene categorization," *ECCV*, 2008.