LEARNING RELATIVE FEATURES THROUGH ADAPTIVE POOLING FOR IMAGE CLASSIFICATION

Ming Shao[1], *Sheng Li*[1], *Tongliang Liu*[3], *Dacheng Tao*[3], *Thomas S. Huang*[4], *Yun Fu*[1,2]

[1] Electrical and Computer Engineering, [2]Computer and Information Science, Northeastern University
[3]Centre for Quantum Computation & Intelligent Systems, University of Technology, Sydney
[4]Electrical and Computer Engineering, University of Illinois at Urbana-Champaign
{mingshao, shengli, yunfu}@ece.neu.edu, tongliang.liu@student.uts.edu.au,
dacheng.tao@uts.edu.au, huang@ifp.illinois.edu

ABSTRACT

Bag-of-Feature (BoF) representations and spatial constraints have been popular in image classification research. One of the most successful methods uses sparse coding and spatial pooling to build discriminative features. However, minimizing the reconstruction error by sparse coding only considers the *similarity* between the input and codebooks. In contrast, this paper describes a novel feature learning approach for image classification by considering the *dissimilarity* between inputs and prototype images, or what we called reference basis (RB). First, we learn the feature representation by max-margin criterion between the input and the RB. The learned hyperplane is stored as the relative feature. Second, we propose an adaptive pooling technique to assemble multiple relative features generated by different RBs under the SVM framework, where the classifier and the pooling weights are jointly learned. Experiments based on three challenging datasets: Caltech-101, Scene 15 and Willow-Actions, demonstrate the effectiveness and generality of our framework.

***Index Terms*—** Image classification, reference basis, adaptive pooling, feature learning

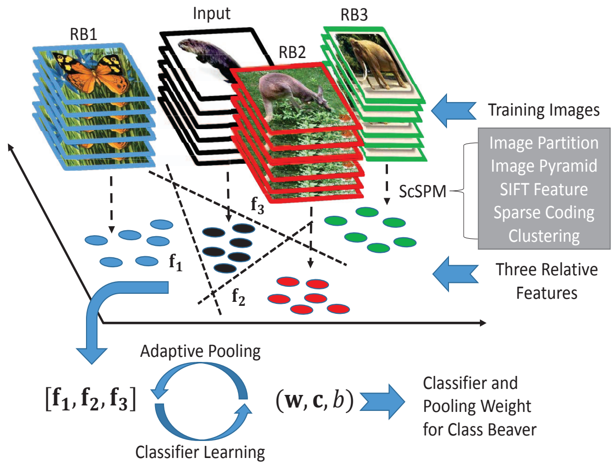

Fig. 1. Illustration of our framework. Features are first extracted by traditional ScSPM [1]. Then we compute discriminative hyperplanes $\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3$ to describe the difference between the input and RBs. After that we assemble these feature vectors by adaptive pooling and use the result as the input for classifier. The output of the training process is the classifier $(\mathbf{w}, b)$, and sparse pooling weight $\mathbf{c}$ for the class "beaver".

1. INTRODUCTION

Most of the recent works employ the *bag-of-feature* (BoF) [2] representation to model image features. However, BoF may fail to capture important characteristics because it discards the feature layout and spatial information. To overcome this, many extensions have been proposed. For example, histogram resolution is explicitly modeled in [3] and spatial layout is considered in [4]. Recent state-of-the-art research in image classification has proposed the following pipeline: first, some handcrafted low-level features, e.g., SIFT, HOG, are extracted from the image; second, either vector quantization or sparse coding is used to formulate discriminative features; third, these features are assembled by an operation that integrates features from all local patches; finally, either linear [1] or non-linear [4] classifier is adopted for classification. Frameworks that follow such a pipeline have received lots of attention and achieved competitive performance on several challenging datasets, e.g., Caltech-101 and Pascal VOC.

The above pipeline, especially the `sparse coding and spatial pooling` framework (ScSPM) [1] attracts substantial research attention because of its biological plausible. From neuroscience, we know that the low-level visual information is encoded by simple neurons in the human visual cortex V1. This encoding uses over-complete dictionaries to produce sparse representation [5]. These observations have caused much interest in coding technique [6], dictionary learning [7], or combination of both [8]. Spatial pooling was introduced by Hubel et al. [9] and was inspired by complex cells in malian visual cortex that identifies mid-level image features invariant to the small spatial shifting. Two common strategies are max pooling and average pooling. A systematic comparisons between these two methods can be found in [10].

Although many sparse coding and pooling techniques have been discussed, they share fundamental similarities. The main purpose of sparse coding is to minimize reconstruction error, which guarantees that the input is similar to certain samples from an over-complete dictionary where the similarity is reflected by the weights. However, we believe that the dissimilarity between the input and certain prototype images is also useful in describing and discriminating an object.

In this paper, we explicitly model the relation between the input and prototype images, or what we called `reference basis` (RB) in different classes, and their difference is then used as the discriminative feature (relative feature) to describe the input (See Fig. 1). Furthermore, since average and max pooling lacks of flexibility due to their use of fixed patterns in assembling features, we propose a novel approach called `adaptive pooling` that learns the weights for pooling of features generated by different RBs. Both the adaptive pooling and the discriminative classifier are jointly learned under the SVM framework, which can be efficiently solved by an alternative optimization algorithm. We demonstrate the effectiveness and generality of our framework on three challenging datasets: Caltech 101, Scene 15 and Willow-Actions.

1.1. Related Work

Inspired by the fact that sparse coding could reproduce the linear filters similar to receptive fields observed in V1 which is the first layer of visual cortex [5], researchers have proposed many algorithms in this line [1, 11, 6, 8, 12, 13]. Different from them, we build our feature sets by the dissimilarity between the input and RBs, and assemble them through adaptive rather than pooling with fixed pattern.

Recently, prototype based methods for image classification have become popular since they can build mid-level features by comparing inputs to a few common images (prototypes) [14, 15, 16, 17]. These comparison results are then used as new descriptors for the inputs. Compared with the over-complete dictionaries and time-consuming sparse coding, prototype image sets are usually small and similarity scores can be computed easily through a linear operation. To name a few: in [15], similarity scores between two data are represented by SVM output scores learned through inputs and an auxiliary dataset; additionally, in [17], SVM output scores learned from inputs and prototype images are utilized as a discriminative metric for inputs; finally, in [16], a few common subspaces are learned by sampling a few data from the training set, and then each training datum can generate the new mid-level feature by projecting itself to the learned subspaces. However, these methods still focus on using the similarity to describe objects.

Most recent work in [18] proposed a novel method for face image set recognition by using prototype images. It is similar to the approach described in this paper since they also use prototype image sets to learn the dissimilarity as the descriptor for the input set. However, it only works well on well-aligned images such as face datasets, and it is hard to extend this approach to objects in arbitrary poses. Instead of learning the global discriminative feature at one time, our framework starts by learning the discriminative feature locally, and then builds up the global representation using all the local features in a hierarchical structure. This enables adaptive pooling to handle objects in arbitrary poses.

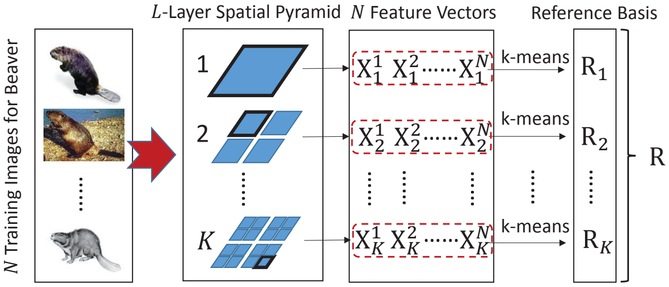
L-Layer Spatial Pyramid
N Feature Vectors
Reference Basis
N Training Images for Beaver
1
2
K
$X_1^1\ X_1^2 \cdots\cdots X_1^N$
$X_2^1\ X_2^2 \cdots\cdots X_2^N$
$X_K^1\ X_K^2 \cdots\cdots X_K^N$
k-means
k-means
k-means
R_1
R_2
R_K
R

Fig. 2. Illustration of generating the reference basis (RB). The first step is to run ScSPM on each patch, and in total we have 3 layers and 21 patches for each image. Since there are N training images for the class "beaver", we consequently obtain N feature sets for each patch, e.g., $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N$. After that, we use k-means to compress feature sets by t representatives. The resulting feature matrix $\mathbf{R} \in \mathbb{R}^{D\times t}$ including K patches will be used as the RB of class "beaver".

2. PROPOSED MODEL

2.1. Reference Basis

Motivated by using the dissimilarity to describe objects, we need to find appropriate reference images from each object category. In this paper, we use reference basis (RB) as the target to which the input image compares. RB is a set of images from the same category, and is compressed by clustering algorithms to balance the feature space. The pipeline of learning one RB is shown in Fig. 2.

First, we use ScSPM[1] [1] to extract low-level features from the input images. In ScSPM, an image is first partitioned into $2^l \times 2^l$ patches at different scales $l = 0, 1, \ldots, L$ to capture the spatial configuration of these features. Then, SIFT is employed on each patch to extract dense local descriptors. After that we adopt sparse coding to encode the SIFT feature into a sparse vector. This represents each patch as a group of sparse vectors, and these patches form an image pyramid. Specifically, for a given image, we segment the images into 21 patches in 3 layers (1+4+16). In Fig. 2, $\mathbf{X}_k^n$ represents the k-th patch in the n-th image, and $\mathbf{X}_k^n \in \mathbb{R}^{d\times t}$, where d is the dimension of the sparse vector (size of the dictionary) and t is the number of feature points in this patch.

[1] Note we do not use max-pooling after sparse coding of each patch.

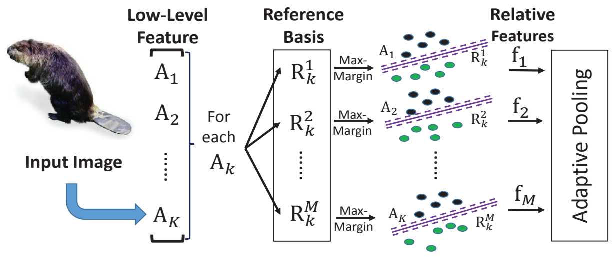

Fig. 3. Pipeline of relative feature learning. For the input, we first extract low-level feature by ScSPM as in Fig. 2. Then for a feature $\mathbf{A}_k$ from the k-th patch, we compute the relative features by comparing it with the corresponding patches from M RBs, and yield relative features $\mathbf{f}_1, \ldots, \mathbf{f}_M$. We repeat this for K patches, and the resulting feature $[\mathbf{f}_1, \ldots, \mathbf{f}_M] \in \mathbb{R}^{D\times M}$ is the input for adaptive pooling.

Note that since we uniformly sample the interest points, the number of feature points in the patches that are located at different layers must be different. To balance the sample space for different patches, we use a clustering algorithm, i.e., k-menas to compute t centers as representatives for each patch over N reference images. To better use dissimilarity to describe objects, more RBs from different object categories are preferred. However, more RBs will raise other issues such as computational complexity. We will discuss how multiple RBs can be better utilized in section 2.2 and 2.3.

2.2. Relative Feature Learning

After we use ScSPM to extract low-level feature for both the input and reference images, we proceed with relative feature learning, a process which quantitatively describes the difference between the input and the RBs. Inspired by [18], the relative feature can be learned through max-margin criterion:

$$\min_{\mathbf{f}_m,\xi,b} \frac{1}{2}\|\mathbf{f}_m\|_2^2 + C\sum_{i=1}^{2\times t} \xi_i, \tag{1}$$

where $\mathbf{f}_m$ is the learned relative feature from the m-th RB, C controls the trade-off between constraint violation and margin maximization, and ξ_i is a slack variable. The optimization is subject to the constraint:

$$y_i\left(\mathbf{f}_m \mathbf{x}_i - b\right) \geq 1 - \xi_i, \quad \xi_i \geq 0, i = 1, \ldots, 2\times t, \tag{2}$$

where $\mathbf{x}_i \in \{\mathbf{A}_k, \mathbf{R}_k^m\}$, y_i is the label corresponding to positive samples from $\mathbf{A}_k$ or negative samples from $\mathbf{R}_k^m$, and k indexes the current patch. The above optimization problem is essentially a soft-margin SVM and the learned hyperplane can separate two classes with a maximal margin. The intuition of using the hyperplane for relative feature is that it quantitatively describes the difference between two classes. For example, if two classes data are quite different in certain dimensions, SVM will weight more in $\mathbf{f}_m$ correspondingly. Therefore this hyperplane reflects the distribution of feature points from the input, as compared to the reference images.

The steps for relative feature learning is shown in Fig. 3. Assume that there are M RBs, then there will be M learning processes to obtain relative features $\mathbf{f}_1, \ldots, \mathbf{f}_M$, for a particular patch k. The relative feature learning for each patch is independent and can be computed in parallel way. After all relative features from K patches are learned, we concatenate each K vectors into a long feature. In the later part, unless stated otherwise, we use $\mathbf{f}_m$ to denote relative feature vector including all the patches from the m-th RB. Therefore, the resulting relative feature $\mathbf{f}_m$ is in $\mathbb{R}^D$, where $D = 21 \times d$.

2.3. Adaptive Pooling

It is know more RBs can describe the characteristics of the input images better, but more RBs require more computational resources. Therefore, we propose a new pooling method to select critical RBs and relative features generated from these RBs as the input feature vector for the classifier. In practice, we can implement this by weighting different relative features, which is similar to the pooling operation such as average pooling. However, the weights in our method are learned for each object category rather than using predefined weights. We call this "adaptive pooling", which can be jointly learned with discriminative hyperplane under the SVM framework.

Suppose we use $\mathbf{B} = [\mathbf{f}_1, ..., \mathbf{f}_M] \in \mathbb{R}^{D\times M}$ to denote the set of relative features, then adaptive pooling can select a few, yet the most important relative features from $\mathbf{B}$, and give higher weights to these features. The linear combination of these relative features is used to train a binary SVM classifier. This process can be formulated as:

$$\begin{aligned} &\min_{\mathbf{w},\mathbf{c},b} \frac{1}{2}\|\mathbf{w}\|^2 + \lambda\|\mathbf{c}\|_1, \\ &\text{s.t.} \quad \forall \mathbf{B}_i \quad y_i(\mathbf{w}^\top \mathbf{B}_i \mathbf{c} - b) > 1, \end{aligned} \tag{3}$$

where $\mathbf{w} \in \mathbb{R}^D$ represents the hyperplane separating positive samples from negative samples, and $\mathbf{B}_i \in \mathbb{R}^{D\times M}$ represents the learned relative feature for $\mathbf{x}_i$, and $\mathbf{c} \in \mathbb{R}^M$ is a sparse weight vector for relative features. The motivation behind the sparsity of $\mathbf{c}$ is to select only the most relevant relative features, rather than considering all of them.

Eq. 3 is convex in either $\mathbf{w}$ or $\mathbf{c}$, but not in both. Therefore, we consider alternatively optimizing one while keeping another fixed, and solve the whole problem in an iterative way. We first convert the original problem into an unconstrained formulation by the Lagrangian Multiplier method:

$$\begin{aligned} L(\mathbf{w}, \mathbf{c}, b, \alpha) = &\frac{1}{2}\|\mathbf{w}\|^2 + \lambda \mathbf{1}^\top[\mathbf{c}_+; \mathbf{c}_-] \\ &- \sum_{i=1}^{N} \alpha_i\{y_i(\mathbf{w}^\top[\mathbf{B}_i, -\mathbf{B}_i][\mathbf{c}_+; \mathbf{c}_-] - b) - 1\}, \end{aligned} \tag{4}$$

where $\alpha_1, \ldots, \alpha_n$ are multipliers, and $\mathbf{c}_+, \mathbf{c}_-$ are positive and negative coefficients of $\mathbf{c}$, namely, $\mathbf{c}_+ \geq 0$ and $\mathbf{c}_- \geq 0$. Then

Algorithm 1 Solving Eq. 4

```
Require: Feature matrix $\mathbf{B}_i$, label vector $y_i$ and $\alpha$
    Initialize: $\mathbf{c} = \frac{1}{M}[1, ..., 1]$
    while not converged do
        update $\alpha$ by traditional dual form solution [19]
        update $\mathbf{c}$ by $\nabla_\mathbf{c}\widetilde{L}(\mathbf{c})$ indicated in Eq. 9, and
        $\mathbf{c}^{(t+1)} = \mathbf{c}^{(t)} - \eta^{(t)}\nabla\widetilde{L}(\mathbf{c}^{(t)})$,
        where $\eta^{(t)}$ is the step size learned through line search.
    end while
    compute $\mathbf{w}$ by
    $\mathbf{w} = \sum_{i=1}^{N}\alpha_i y_i[\mathbf{B}_i, -\mathbf{B}_i][\mathbf{c}_+; \mathbf{c}_-]$
    compute $b$ by
    $b = \frac{1}{\|S\|}\sum_{i \in S}(y_i - \mathbf{w}^\top[\mathbf{B}_i, -\mathbf{B}_i][\mathbf{c}_+; \mathbf{c}_-])$
```

we derive the dual form of the original problem by the following two equations:

$$\frac{\partial L(\mathbf{w}, \mathbf{c}, b, \alpha)}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N}\alpha_i y_i[\mathbf{B}_i, -\mathbf{B}_i][\mathbf{c}_+; \mathbf{c}_-] \tag{5}$$

$$\frac{\partial L(\mathbf{w}, \mathbf{c}, b, \alpha)}{\partial b} = \sum_{i=1}^{N}\alpha_i y_i \tag{6}$$

By setting Eq. 5 and Eq. 6 to zero and substituting the results into Eq. 4, we obtain the following dual problem:

$$\begin{aligned}\widetilde{L}(\mathbf{c}, \alpha) = &\sum_{i=1}^{N}\alpha_i + \lambda\mathbf{1}^\top[\mathbf{c}_+; \mathbf{c}_-] \\ &- \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j[\mathbf{c}_+; \mathbf{c}_-]^\top\mathbf{D}_i[\mathbf{c}_+; \mathbf{c}_-] \qquad (7)\\ \text{s.t.} \quad &\sum_{i=1}^{N}\alpha_i y_i = 0 \quad \text{and} \quad \alpha_i \ge 0,\end{aligned}$$

where

$$\mathbf{D}_i = \begin{bmatrix}\mathbf{B}_i^\top\mathbf{B}_i & -\mathbf{B}_i^\top\mathbf{B}_i \\ -\mathbf{B}_i^\top\mathbf{B}_i & \mathbf{B}_i^\top\mathbf{B}_i\end{bmatrix}. \tag{8}$$

Finally, $\mathbf{c}$ can be updated by the gradient of $\widetilde{L}(\mathbf{c})$ with respect to $\mathbf{c}$:

$$\nabla_\mathbf{c}\widetilde{L}(\mathbf{c}) = \lambda\mathbf{1} + \sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j\mathbf{D}_i[\mathbf{c}_+; \mathbf{c}_-]. \tag{9}$$

Algorithm 1 summarizes the procedure where we solve two sub-problems. First, we fix $\mathbf{c}$ and update α by traditional solution of the dual form SVM [19]. Second, we fix $\mathbf{w}$ and update $\mathbf{c}$ through gradient projection sparse representation, which is originally proposed for lasso in [20]. These two steps are repeated until both α and $\mathbf{c}$ converge. After we obtain the optimized α and $\mathbf{c}$, both $\mathbf{w}$ and b in the prime form can be computed accordingly. We initialize $\mathbf{c}$ by average pooling vector $\mathbf{c} = \frac{1}{M}[1, ..., 1]$.

2.4. Theoretical Analysis

In this section, we discuss the error bound of the model proposed in Eq. 3. One issue is that the proposed method is not convex. Therefore, the solution by Algorithm 1 is not guaranteed to be global optimal. Although $\mathbf{c}$ is initialized with a fairly good value which makes it unlikely to be trapped at a bad local minimum, we still want to know how well the model can perform in terms of classification error under the current formulation. Enlightened by the uniform stability proposed in [21], we are convinced that the expected classification error (generalization error) is bounded by the empirical error and some constant. This indicates that the proposed model can be well extrapolated to the test data if drawn from the same distribution.

Theorem 1 *Let the learned parameter* $\mathbf{w}$ *and the relative feature* $\mathbf{f}$ *be bounded, which means* $\|\mathbf{w}\| \le C_1$ *and* $\|\mathbf{f}\| \le C_2$*. For any hyperplane learned from problem (3) with the soft constraint about* $\|c\|_1$ *being replaced by the hard constraint* $\|c\|_1 \le \alpha$*, and any* $\delta > 0$*, with probability at least* $1 - \delta$*, we have:*

$$\begin{aligned}&E_z\left[\max\left\{0, 1 - y\left(\mathbf{w}^\top\mathbf{B}_i\mathbf{c} - b\right)\right\}\right] \\ &\le \frac{2\alpha^2 C_2^2}{N} + (4\alpha^2 C_2^2 + 1 + \alpha C_1 C_2 + |b|)\sqrt{\frac{\log 1/\delta}{2N}}.\end{aligned}$$

Let $\|\mathbf{c}\|_1 \le \frac{1}{\lambda}$*, we have*

$$\begin{aligned}&E_z\left[\max\left\{0, 1 - y\left(\mathbf{w}^\top\mathbf{B}_i\mathbf{c} - b\right)\right\}\right] \\ &\le \frac{2C_2^2}{\lambda^2 N} + \left(\frac{4C_2^2}{\lambda^2} + 1 + \frac{C_1 C_2}{\lambda} + |b|\right)\sqrt{\frac{\log 1/\delta}{2N}}.\end{aligned}$$

3. EXPERIMENTS AND RESULTS

This section compares our method with the state-of-the-art algorithms on several datasets: Caltech 101, Scenes 15, and Willow-Actions. In all experiments, images are first converted to gray scale. Then we use the SIFT feature in a dense grid fashion as the object descriptor. The size of the codebook for sparse coding is fixed at 1024, and both codebooks and coding are learned using the method in [1]. For each dataset, we use the training data to construct the RB, and therefore, the number of RBs is equal to that of object categories. Then the relative features for each datum are computed through Eq 1. Note that the number of centers for each RB is fixed at $t = 50$. After that, we will learn the sparse vector $\mathbf{c}$ and $\mathbf{w}$ for each class. This can be viewed as an one-vs-rest training scheme, and any training data not in the current class are seen as negative samples. The label of the test sample can be identified by: $m^* = \arg\max_m\{\mathbf{w}_m^\top\mathbf{x} + b_m\}, m = 1, 2, \ldots, M$.

3.1. Caltech 101

Caltech 101 is a popular dataset for general image classification, which includes 101 categories of object (animals, vehi-

Fig. 4. Sample images from, Caltech 101, Scene 15, and Willow-Action datasets.

Table 1. Classification accuracy (%) on Caltech 101

	Codebook	15 train	30 train
ScSPM [1]	1024(SC)	67.02± 0.4	73.20 ± 0.5
LLC [6]	1024	65.43± 0.4	73.44 ± 0.6
OC-RF [22]	1024(SC)	-	75.3 ± 0.7
LSLR [23]	-	66.1	73.6
LSGC [12]	1000	**68.7**	75.07
Ours	101(RB)	67.89 ± 0.9	**76.42** ± 1.1

cles, airplanes, etc.) with strong shape variability. The number of images per category varies from 31 to 800. Most images are about 300 × 300 pixels. In the experiment, we followed the conventional setup for Caltech-101, namely, training on 15 or 30 images per category, and testing on the rest. We repeat this 10 times by selecting different training/testing partitions. Detailed comparison results are shown in Table 1 where our method is comparable to others in 15 train, and best in 30 train. Fig. 5 show the effectiveness of the sparsity term **c** by comparing ours with average pooling as the baseline.

3.2. Scene 15

Scene 15 dataset consists of 15 natural scene categories, e.g., "building", "bedroom", with 4485 images. We follow the setup in [4] that randomly selects 100 images from each class for training and the rest of the images for testing. The sample images are shown in Fig. 4. Since the scene label is not only determined by the environments in the images, but also by the objects that appear in the images, the classification of these scene images turns out to be a very challenging task.

Table 2 shows that our method performs best on scene 15 dataset. However, the performance can be increased if we include more discriminative features such HOG, GIST, and LBP, as suggested by [24]. Since we need to balance both the numbers of positive and negative samples, and the numbers of feature points from different patches, a clustering method is used to compute a fixed number of centers as representatives for each class. However, the number of centers t is yet another parameter that should be tuned. Intuitively, a larger t will be able to describe the object better, but it will also sacrifice computation efficiency. To demonstrate the effectiveness of t on the final recognition accuracy, we showcase the recognition results on Scene 15 dataset over different t in Table 3, where t = 50 achieves the best result.

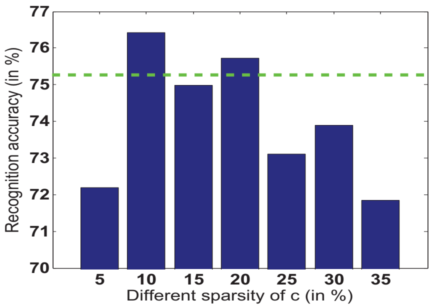

Fig. 5. Accuracy changes with different sparsity of **c** on Caltech 101 with 30 train. The green dot line is the performance of average pooling.

Table 2. Classification accuracy (%) on Scene 15 (100 train)

Lian et al. [25]	Josip et al. [26]	Kobayashi [27]	Ours
78.1	83.9	85.6	**87.7**

Table 3. Classification accuracy (%) with different t on Scene 15.

	10	20	30	40	50
80 train	50.2	65.6	77.3	84.9	86.3
100 train	56.3	72.4	81.5	87.2	87.7

3.3. Willow-Actions

Willow-Actions dataset is a still image set used for action recognition. The images in this dataset are downloaded from the Internet, and contains significant variations, making this dataset challenging. It has 7 classes of common human activities, e.g., "play instruments", "horse riding", each of which contains at least 109 subjects. We follow the conventional setup by randomly selecting 70 subjects for training and use the rest of the images for testing. Labeled human bounding boxes are used for action recognition. Similar to previous work [28], we also expand the given bounding boxes by 50% to include contextual information outside the body region.

Table 4 shows the performance of our algorithm and several state-of-the-art methods. We detail the results in each category as well as the average performance. In most of these categories (5 out of 7), our algorithm achieves better performance. We also note that all these 5 categories are highly related to objects around the human body region, e.g., playing music with instrument, riding horse. However, when actions do not have clear contextual information, our method is still comparable to others. On average, the proposed algorithm achieves the best performance.

Table 4. Classification accuracy (%) on Willow-Actions (70 train)

	inter. w/ comp.	photographing	p. music	r. bike	r. horse	running	walking	average
Delaitre et al. [28]	56.6	37.5	72.0	90.4	75.0	**59.7**	57.6	64.1
SP [4]	49.4	41.3	74.3	87.8	73.6	53.3	**58.3**	62.6
ov. SP [4]	57.8	39.3	73.8	88.4	80.8	55.8	56.3	64.6
Sharma et al. [29]	59.7	42.6	74.6	87.8	84.2	56.1	56.5	65.9
Ours	**62.7**	**44.3**	**75.7**	**90.8**	**86.3**	58.7	56.8	**67.9**

4. CONCLUSION

In this paper, we proposed a novel feature learning scheme for general image classification. Different from traditional methods that consider using the similarity between the input and codebooks, we proposed to use the dissimilarity between the input and prototype images for better descriptive capability. In addition, we proposed a new learning algorithm called adaptive pooling that jointly learns the classifier and the pooling weight. Extensive experiments on Caltech 101, Scene 15, and Willow-Actions datasets demonstrated the effectiveness and generality of our method.

5. ACKNOWLEDGEMENT

This research was supported in part by the NSF CNS award 1314484, Office of Naval Research award N00014-12-1-1028, Air Force Office of Scientific Research award FA9550-12-1-0201, and Australian Research Council Projects FT-130101457 and DP-140102164.

6. REFERENCES

[1] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *CVPR*, 2009. 1, 2, 4, 5

[2] Li Fei-Fei and Pietro Perona, "A bayesian hierarchical model for learning natural scene categories," in *CVPR*. IEEE, 2005, vol. 2, pp. 524–531. 1

[3] Kristen Grauman and Trevor Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," in *ICCV*, 2005. 1

[4] Cordelia Schmid, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006. 1, 5, 6

[5] Bruno A. Olshausen and David J. Fieldt, "Sparse coding with an overcomplete basis set: a strategy employed by v1," *Vision Research*, vol. 37, pp. 3311–3325, 1997. 1, 2

[6] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong, "Locality-constrained linear coding for image classification," in *CVPR*. 1, 2, 5

[7] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro, "Online learning for matrix factorization and sparse coding," *JMLR*, vol. 11, pp. 19–60, 2010. 1

[8] R. Rigamonti, M. A. Brown, and V. Lepetit, "Are sparse representations really relevant for image classification?," in *CVPR*, 2011. 1, 2

[9] David H Hubel and Torsten N Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, pp. 106, 1962. 1

[10] Y-Lan Boureau, Jean Ponce, and Yann Lecun, "A theoretical analysis of feature pooling in visual recognition," in *ICML*, 2010. 2

[11] Jianchao Yang, Kai Yu, and Thomas Huang, "Efficient highly overcomplete sparse coding using a mixture model," in *ECCV*, 2010. 2

[12] Amirreza Shaban, Hamid R Rabiee, Mehrdad Farajtabar, and Marjan Ghazvininejad, "From local similarity to global coding: An application to image classification," in *CVPR*. IEEE, 2013, pp. 2794–2801. 2, 5

[13] Amirreza Shaban, Hamid R. Rabiee, Mehrdad Farajtabar, and Marjan Ghazvininejad, "Low-rank sparse coding for image classification," in *ICCV*, 2013. 2

[14] Ariadna Quattoni, Michael Collins, and Trevor Darrell, "Transfer learning for image classification with sparse prototype representations," in *CVPR*. IEEE, 2008, pp. 1–8. 2

[15] Yaniv Taigman, Lior Wolf, and Tal Hassner, "Multiple one-shots for utilizing class label information.," in *BMVC*, 2009, pp. 1–12. 2

[16] Dengxin Dai and Luc Van Gool, "Ensemble projection for semi-supervised image classification," in *ICCV*, 2013. 2

[17] Meina Kan, Dong Xu, Shiguang Shan, Wen Li, and Xilin Chen, "Learning prototype hyperplanes for face verification in the wild," *IEEE TIP*, 2013. 2

[18] Mingbo Ma, Ming Shao, Xu Zhao, and Yun Fu, "Prototype based feature learning for face image set classification," in *IEEE FGR*, 2013. 2, 3

[19] Christopher J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998. 4

[20] Mário AT Figueiredo, Robert D Nowak, and Stephen J Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 1, no. 4, pp. 586–597, 2007. 4

[21] Olivier Bousquet and André Elisseeff, "Stability and generalization," *JMLR*, vol. 2, pp. 499–526, 2002. 4

[22] Yangqing Jia, Chang Huang, and Trevor Darrell, "Beyond spatial pyramids: Receptive field learning for pooled image features," in *CVPR*. IEEE, 2012, pp. 3370–3377. 5

[23] Yangmuzi Zhang, Zhuolin Jiang, and Larry S Davis, "Learning structured low-rank representations for image classification," in *CVPR*. IEEE, 2013, pp. 676–683. 5

[24] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba, "Sun database: Large-scale scene recognition from abbey to zoo.," in *CVPR*, 2010. 5

[25] Xiao-Chen Lian, Zhiwei Li, Bao-Liang Lu, and Lei Zhang, "Max-margin dictionary learning for multiclass image categorization," in *ECCV*, 2010. 5

[26] Josip Krapac, Jakob Verbeek, and Frédéric Jurie, "Learning tree-structured descriptor quantizers for image categorization," in *BMVC*, 2011. 5

[27] Takumi Kobayashi, "Bfo meets hog: Feature extraction based on histograms of oriented p.d.f. gradients for image classification," in *CVPR*, 2013, pp. 747–754. 5

[28] V. Delaitre, J. Sivic, and I. Laptev, "Learning person-object interactions for action recognition in still images," in *NIPS*, 2011. 5, 6

[29] Gaurav Sharma, Frederic Jurie, and Cordelia Schmid, "Discriminative spatial saliency for image classification.," in *CVPR*, 2012. 6