

Learning Directional Local Pairwise Bases with Sparse Coding

Nobuyuki Morioka
nmorioka@cse.unsw.edu.au

The University of New South Wales &
NICTA,
Sydney, AUSTRALIA

Shin'ichi Satoh
satoh@nii.ac.jp

National Institute of Informatics,
Tokyo, JAPAN

Abstract

Recently, sparse coding has been receiving much attention in object and scene recognition tasks because of its superiority in learning an effective codebook over k -means clustering. However, empirically, such codebook requires a relatively large number of visual words, essentially bases, to achieve high recognition accuracy. Therefore, due to the combinatorial explosion of visual words, it is infeasible to use this codebook to represent higher-order spatial features which are equally important in capturing distinct properties of scenes and objects. Contrasted with many previous techniques that exploit higher-order spatial features, Local Pairwise Codebook (LPC) is a simple and effective method to learn a compact set of clusters representing pairs of spatially close descriptors with k -means. Based on LPC, this paper proposes Directional Local Pairwise Bases (DLPB) that applies sparse coding to learn a compact set of bases capturing correlation between these descriptors, so to avoid the combinatorial explosion. Furthermore, such bases are learned for each quantized direction thereby explicitly adding directional information to the representation. We have evaluated DLPB with several challenging object and scene category datasets. Our experimental results show that DLPB outperforms the baselines across all datasets and achieves the state-of-the-art performance on some datasets.

1 Introduction

Initially proposed by Olshausen and Field [20], sparse coding has been receiving much attention due to its effectiveness in capturing salient properties of signals, images and videos by sparse representations. Specifically, it learns an overcomplete non-parametric basis set also known as dictionary which is used to infer a set of latent variables with some regularization sparsity constraint. This technique has been shown to perform competitively against the state-of-the-art methods for image restoration [3, 14], image segmentation [14], edge detection [15] and image classification [21, 29] tasks.

In image classification tasks, methods based on sparse coding or its variant learn their dictionaries mainly from a collection of raw image patches [12, 21, 22]. These have shown to perform competitively against the state-of-the-art. Alternatively, Yang *et al.* [29] have recently proposed a technique of learning a dictionary based on SIFT descriptors [15]. Combined with spatial max pooling representation, their method has achieved the state-of-the-art

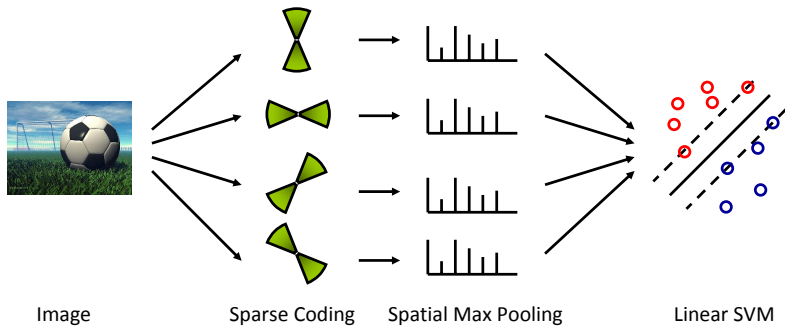


Figure 1: Directional Local Pairwise Bases. Given a feature descriptor in an image, four directional relationship kernels are applied to form local pairwise descriptors with nearby descriptors. For each direction, sparse coding is applied and statistics are computed based on spatial max pooling. The statistics from all four directions are then concatenated and fed into a linear SVM classifier

performance on several challenging scene and object category datasets indicating the potential of sparse coding in image classification tasks. However, to obtain superior performance with sparse coding, one needs to learn a significantly large number of bases. This prohibits its use in modeling higher-order spatial features, like doublets [9, 13, 14, 23, 24] and triplets [27], due to the combinatorial explosion of the features. For an example, [29] have learned 1024 bases. If pairs of such bases are to be considered, there will be about 1,000,000 possible pairs. To overcome such combinatorial explosion issue, recently Morioka and Satoh [19] have proposed Local Pairwise Codebook (LPC) that efficiently builds a compact codebook for local pairs of SIFT descriptors based on k -means clustering. This method essentially concatenates pairs of spatially close descriptors prior to the clustering process, thereby directly controlling the codebook size as well as avoiding the need of feature selection.

Based on LPC, this paper proposes a new compact sparse image model called *Directional Local Pairwise Bases* (DLPB) which captures salient local pairs of SIFT descriptors with sparse coding. As mentioned above, the use of sparse coding thus far is limited to local appearance feature descriptors or raw image patches only. Therefore, our work is the first attempt to learn a dictionary using sparse coding based on local pairs of the descriptors. To avoid quadratic growth in the number of bases, before applying sparse coding, we first represent each pair of spatially close local features as a joint vector of the two descriptors with relative directional information encoded. The complexity of learning a dictionary is further reduced by learning a set of four independent smaller dictionaries instead where each dictionary is specific to one directional relationship, as illustrated in Figure 1. We then combine DLPB with spatial max pooling presented in [29] and simply use a linear SVM for classification. This achieves the state-of-the-art performance in some of the challenging object and scene category datasets, namely Caltech-101, Caltech-256, MSRCv2, Pascal VOC 2007 and 15 Scenes. Our work is different from [29] in that we are learning the dictionary of higher-order spatial features, specifically pairs of spatially close SIFT descriptors, to jointly capture local appearance and spatial information.

The paper is organized as follows. Section 2 reviews sparse coding and also its connection to k -means clustering. Then, in Section 3, DLPB is explained in detail. This is followed by the experimental results in Section 4 to demonstrate the effectiveness of DLPB. Lastly,

Section 5 concludes the paper with possible future work.

2 Background

The goal of sparse coding is to reconstruct an input signal vector \mathbf{d}_i (e.g. image patch or feature descriptor) by a linear combination of a dictionary D in $\mathbb{R}^{n \times k}$ and a sparsely represented vector α_i in \mathbb{R}^k . Each column of D is referred to as a basis vector D_j in \mathbb{R}^n and can sometimes be interpreted as a salient feature. Given N input vectors $\{\mathbf{d}_i\}$ where $i = 1 \dots N$, both D and $\alpha = \{\alpha_1, \dots, \alpha_N\}$ are unknown apriori, so they are learned by solving the following minimization problem:

$$\begin{aligned} \min_{D, \alpha} \sum_{i=1}^N \|\mathbf{d}_i - D\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1 \quad (1) \\ \text{subject to } \|D_j\|^2 \leq c, \forall j = 1, \dots, k. \end{aligned}$$

The ℓ_1 regularization sparsity constraint on α_i enforces each \mathbf{d}_i to be encoded with a small number of bases from D . While it is possible to use other regularization constraints like ℓ_0 or ℓ_2 norm, it has been shown previously in [21, 29] that ℓ_1 norm performs well in classification tasks. Thus, in our work, we use the ℓ_1 regularization. Since the optimization problem can be interpreted as a joint minimization of reconstruction error and sparsity term, a positive constant λ controls the trade-off between accuracy of reconstruction and sparseness of α . To avoid learning trivial solutions, e.g. D and α taking large and small values respectively to achieve low energy, norm of each basis vector in D is constrained by a constant c , often set to 1.

The cost function given above is non-convex with respect to both D and α . However, it is convex when one is fixed. Thus, we alternate between learning D while fixing α and inferring α while fixing D . Although this is likely to converge to a local minima instead of the global minima, the learned D and inferred α work reasonably well in practice. Thus, given α is fixed, to learn or update D , we need to solve the constrained optimization problem. We adopt an efficient method presented in [10] where by the dual problem of its Lagrangian is first obtained and is optimized using Newton's method. Then, given D is fixed, to infer α , we to solve the ℓ_1 regularized optimization problem. We use the efficient feature-sign search algorithm also described in [10]. This two-step algorithm is repeated until convergence or maximum number of iterations is reached. In our experiments, we set this number to 50.

As we are applying sparse coding to build a codebook, it is worthwhile mentioning the relationship or connection to k -means clustering which is a standard method of learning the codebook. To view k -means clustering as an optimization problem, we formulate as follows:

$$\begin{aligned} \min_{D, \alpha} \sum_{i=1}^N \|\mathbf{d}_i - D\alpha_i\|_2^2 \quad (2) \\ \text{subject to } \|\alpha_i\|_0 = \|\alpha_i\|_1 = 1, \forall i = 1, \dots, N. \end{aligned}$$

In contrast to the regularization constraint on α in sparse coding, it is much more constrained in that each α_i is a binary vector taking exactly one 1 and the rest is all 0 due to the hard assignment of clusters. Such assignment, however, can be problematic when working in a high-dimensional feature space as addressed in [26]. In one perspective, sparse coding relaxes the hard assignment rule in k -means like the soft assignment technique presented in

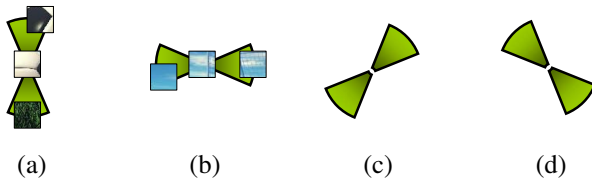


Figure 2: Illustrated examples of different relative directional relationship kernels of neighborhood size δ : (a) vertical (vt), (b) horizontal (hz), (c) first diagonal ($d1$), (d) second diagonal ($d2$). Given a pair of features, regardless of which feature is centered at the kernel, the relative directional relationship is the same

[16], but is more principled where reconstruction error and sparsity term are jointly minimized under one optimization framework.

3 Directional Local Pairwise Bases

This section describes our work on Directional Local Pairwise Bases, DLPB for short. Many previous works on feature pairing [9, 13, 14, 23, 24] first learn visual words from a set of local descriptors and then consider possible pairs of such visual words. However, the number of such pairs grows quadratic with respect to the number of visual words. To avoid such issue, we use Local Pairwise Codebook (LPC) proposed by Morioka and Satoh [19] which first represents each pair of spatially close descriptors as a joint descriptor by simple vector concatenation and then applies k -means clustering in the feature space of the joint descriptors to achieve a compact codebook capturing correlation between spatially close descriptors. Such technique is particularly suited for DLPB which replaces k -means clustering with sparse coding, as relatively a large number of bases is typically required by sparse coding. Furthermore, contrasted with LPC, DLPB captures directional information explicitly making its representation richer.

3.1 Representing Local Pairwise Features

Given an image, we densely sample feature points as similar to [11, 19]. Each feature point f_i in the image is encoded as (x_i, y_i, \mathbf{d}_i) where x_i and y_i denote the feature location and \mathbf{d}_i is the feature descriptor. This is followed by pairing up the features that are within δ pixels away from each other. For each pair of spatially close descriptors, we extend the joint descriptor representation of LPC by assigning one of the discretized directional relationships which semantically define vertical (vt), horizontal (hz), first diagonal ($d1$) and second diagonal ($d2$) directions as illustrated in Figure 2. We set the partition of each directional kernel to be roughly equal to each other. Our definition of directional relationships is slightly different from others like Correlogram [11] which treat opposing directions to be separate. For example, regardless of whether f_i appears spatially above f_j or the other way around, the relative directional relationship is always vertical by our definition, but with other methods, it will be different depending on which feature becomes the pivot of the kernel. By introducing this invariance, we can reduce the number of pairs formed by half and the final representation of an image becomes relatively more sparse. Once r_{ij} is determined, each pair is represented

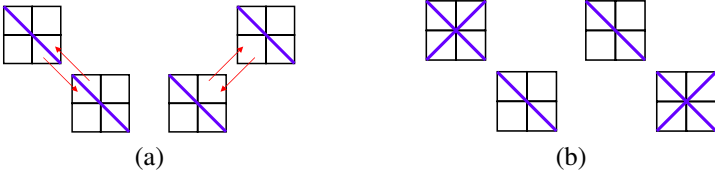


Figure 3: Illustrated examples of local pairwise descriptors. (a) The left shows one long line segment and the right shows two lines in parallel. The two pairwise features are associated with different directional relationships, *i.e.* diagonal 1 and diagonal 2, so to avoid unwanted equivalence class of the features. The red arrows show the relative directions. With our directional kernel, the two arrows for each pairwise feature map to the same directional relationship to avoid duplicates. (b) The two pairwise features with the same pair of descriptors. By ordering the two descriptors based on their spatial locations, these two pairwise features are treated as different features

as:

$$f_{(i,j)} = \left(\frac{(x_i + x_j)}{2}, \frac{(y_i + y_j)}{2}, r_{ij}, \mathbf{p}_{ij} \right) \quad (3)$$

where \mathbf{p}_{ij} denotes vector concatenation of two feature descriptors $[\mathbf{d}_i \mathbf{d}_j]$. The order of \mathbf{d}_i and \mathbf{d}_j is based on the spatial locations of the features and their assigned directional relationship. Formally speaking, the spatial coordinates are first rotated so that their assigned directional relationship becomes the x-axis of a new coordinate system and then we decide the order of two features based on their x-coordinates. However, in our implementation, we simply compare y-coordinates when the directional relationship is vertical and x-coordinates for the other three directions. With such ordering, the relative location of the two feature descriptors are encoded in the joint descriptor. As illustrated in Figure 3, when the two feature descriptors that are in diagonal relationship are concatenated based on the ordering, we can tell from the concatenated descriptor \mathbf{p}_{ij} that which descriptor is above of the other.

3.2 Learning Direction Specific Dictionaries

For each directional relationship r , DLPB learns a specific dictionary. If we denote a set of possible directional relationships as $R = \{vt, hz, d1, d2\}$, then D and α are redefined as $D = \{D^r\}_r^R$ and $\alpha = \{\alpha^r\}_r^R$. The learning problem stated in (1) can also be reformulated as follows:

$$\begin{aligned} \min_{D, \alpha} \sum_{r \in R} \sum_i^{N^r} \|\mathbf{p}_i^r - D^r \alpha_i^r\|_2^2 + \lambda \|\alpha_i^r\|_1 \\ \text{subject to } \|D_j^r\|^2 \leq c, \forall j = 1, \dots, k. \end{aligned} \quad (4)$$

where the sparse representation of a pairwise feature descriptor \mathbf{p}_i^r with a directional relationship r is α_i^r and is inferred with D^r . N^r is the number of pairwise feature descriptors sampled for learning D^r . Since each set $\{D^r, \alpha^r\}$ can be learned and inferred independently from each other, the computational cost is significantly reduced if a large dictionary D is required. In the case of our experiments, we have learned 12800 bases in total by learning four sets of 3200 bases separately. Learning all bases jointly would be computationally too expensive. The decomposition of the dictionary not only improves efficiency, but avoids

unnecessary confusion or equivalence relation between the local pairwise features with different directions as depicted in Figure 3.

3.3 Spatial Max Pooling

Once the dictionaries are learned, we represent each image with spatial max pooling [29]. This is done by first inferring a sparse code vector α_i^r for each local pairwise feature descriptor \mathbf{p}_i^r formed within an image. Then, we obtain a set of local statistics over multiple scales. At every level of scale l , the image is partitioned into $M_l \times M_l$ disjoint regions where M_l is defined as 2^l . For each region m , its statistics \mathbf{h}_m^r are represented as:

$$\mathbf{h}_{l,m}^r(j) = \max\{|\alpha_1^r(j)|, |\alpha_2^r(j)|, \dots, |\alpha_{N_m}^r(j)|\} \quad (5)$$

where each dimension j stores the max value of each basis \mathbf{D}_j^r spatially pooling all pairwise descriptors with a directional relationship r within m and N_m denotes the number of descriptors inside m . Once the local statistics are computed for all regions at scale l , the vectors are concatenated and are ℓ_2 normalized as:

$$\mathbf{h}_l^r = \frac{[\mathbf{h}_{l,1}^r \mathbf{h}_{l,2}^r \dots \mathbf{h}_{l,m}^r]}{\|[\mathbf{h}_{l,1}^r \mathbf{h}_{l,2}^r \dots \mathbf{h}_{l,m}^r]\|_2} \quad (6)$$

To obtain the final representation of the image denoted as \mathbf{h} , we further concatenate all \mathbf{h}_l^r over multiple levels of scale ($l \in \{0 \dots L\}$) and multiple dictionaries ($r \in R$). Finally, we train a model with a linear SVM and use the learned model for classifying test images.

4 Experimental Results

In this section, we present our experimental results on DLPB compared against several baselines and previously published techniques on challenging datasets, namely Caltech-101, Caltech-256, MSRCv2, Pascal VOC 2007 and 15 Scenes. For all datasets, SIFT descriptors are densely sampled at every 8 pixel step. We use 32 dimensional SIFT descriptors extracted from 16×16 patches with 2×2 grids and 8 orientation bins. Such descriptor is shown to perform competitively against the original 128 dimensional SIFT descriptors [25]. The baselines used to compare against DLPB are as follows:

1. **SPM**: Codebook is learned using k -means based on a set of single SIFT descriptors. This is combined with spatial pyramid matching kernel as described in [10].
2. **ScSPM**: Bases are learned with sparse coding based on a set of single SIFT descriptors. This is combined with spatial max pooling and classified using a linear SVM. It is our reimplement of [29].
3. **DLPC**: Directional Local Pairwise Codebook, a variant of DLPB. Instead of sparse coding, k -means is used to build direction specific codebooks. We reformulate (2) to the one like (4), so to learn direction specific codebooks. We compute spatial pyramid matching kernel for each directional relationship and then simply take the average of the kernels.

As far as the codebook size is concerned, we have tested $\{400, 800, 1600, 3200\}$ and reported the best performing one for each method. λ is set between 0.1 and 0.25. δ is set to 24 unless otherwise stated. We set L to be 2.

Methods	15 training	30 training
KC [26]	-	64.14±1.18
SPM [10]	56.40	64.60±0.80
KSPM [8]	-	67.40
HC [28]	57.70±0.59	-
ML+CR [9]	65.00±1.14	69.60
NBNN [9]	61.00	70.40
HG [50]	-	73.10
ScSPM [49]	67.00±0.45	73.20±0.54
SPM ($k=400$)	56.85±0.71	65.38±1.16
ScSPM ($k=800$)	64.79±0.91	71.98±0.70
DLPC ($k=3200$)	62.76±0.81	71.60±0.65
DLPB ($k=3200$)	69.90±0.63	77.33±0.63

Table 1: Recognition accuracy (%) on Caltech-101

4.1 Caltech-101

The Caltech-101 dataset [9] consists of 9144 images in 102 different object classes. In our experiments, the Background class is removed. Each class has between 31 and 800 images and significantly varies in shape. Images are resized to be less than 300x300 pixels, but preserving their aspect ratios. We have used 15 and 30 training images per class and the rest for testing. The results are presented in Table 1. Both ScSPM and DLPC outperform SPM by at least 6%. On the other hand, DLPB is significantly better than all baselines. For 30 training images, it has obtained performance increase of 12% from SPM and at least 5% from both ScSPM and DLPC. When compared against the previously published results based on a single descriptor type, DLPB achieves the state-of-the-art.

4.2 Caltech-256

The Caltech-256 dataset [9] consists of 30,607 images in 257 different object classes. Compared to Caltech-101, there are several improvements including higher intra-class variability and higher variability in object poses and locations. Similar to the previous experiments, the Clutter class is removed and the images are resized down to 300x300 pixels. We have ran our experiments using 15, 30, 45 and 60 training images and 25 testing images. As shown in Table 2, DLPB shows consistent improvement over other methods including previously published results. Interestingly, as the number of training images increases, the performance gap between DLPC and ScSPM increases. A similar trend is seen between DLPB and ScSPM as well. In sum, jointly capturing local spatial and appearance information with DLPB has achieved state-of-the-art performance on this dataset as well.

4.3 MSRC Object Classes Database

The MSRCv2 dataset is a relatively small object class database compared to Caltech-101/256, but is considered be a more difficult dataset due to its high intra-class variability [27]. We have simply followed the experimental setup used by the unbounded-order spatial feature method proposed by Zhang and Chen [50], except we have used dense sampling instead of

Methods	15 training	30 training	45 training	60 training
KC [26]	-	27.17±0.46	-	-
KSPM [8]	-	34.10	-	-
ScSPM [29]	27.73±0.51	34.02±0.35	37.46±0.55	40.14±0.91
SPM ($k=800$)	23.50±0.42	29.14±0.38	32.17±0.53	34.21±0.24
ScSPM ($k=1600$)	29.21±0.50	35.19±0.52	38.94±0.42	40.84±0.61
DLPC ($k=3200$)	29.59±0.36	36.74±0.41	41.23±0.62	43.93±0.80
DLPB ($k=3200$)	33.35±0.58	40.81±0.59	44.95±0.61	47.53±0.54

Table 2: Recognition accuracy (%) on Caltech-256

Methods	[30]	SPM	ScSPM	DLPC	DLPB
k	-	800	3200	3200	3200
Acc.	80.4±2.5	84.4±2.5	87.0±3.4	86.4±2.7	88.5±3.3

Table 3: Recognition accuracy (%) on MSRCv2

an interest point detector to extract feature descriptors. Nine out of fifteen classes are chosen (*i.e.* cow, airplanes, faces, cars, bikes, books, signs, sheep and chairs) where each class contains 30 images. For each experiment, we have randomly sampled 15 training and 15 testing images class and no background is removed from the images. The results are reported in Table 3.

4.4 Pascal VOC 2007

The Pascal VOC 2007 dataset [9] is a collection of 9963 images containing 20 object classes, *i.e.* aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair cow, dinning table, dog, horse motorbike, person, potted plant, sheep, sofa, train and tv monitor. The images are compiled from Flickr where appearance, pose, location and illumination significantly vary, hence, the dataset is considered to be very challenging. We have used both the training and validation image sets for training and measured average precision for each object class on the testing image set. The results are reported in Table 4. For all methods, we have set k to be 3200. DLPB has outperformed ScSPM on 18 out of 20 object classes, obtaining 2.1% improvement on average. It has also performed better than DLPC on 15 out of 20 object classes, obtaining 1.4% increase on average. While we have only used SIFT descriptors from a single scale as similar to previous experiments, the performance of DLPB on this dataset is likely to increase further when multiple descriptors across different scales are used like the state-of-the-art method [9]. In fact, we have seen slight performance increase, when ScSPM and DLPB are combined.

4.5 15 Scenes

The 15 Scenes dataset [10] contains 4485 images in 15 categories ranging from indoor scenes (*e.g.* living room and kitchen) to outdoor scenes (*e.g.* mountain and street). 100 images per category are used for training and the rest is used for testing. We set δ to be 16 for this experiment. The results are shown in Table 5. As similar to the previous experiments, DLPB achieves a better result than the baselines. However, the performance gap is

class	aero	bicyc	bird	boat	bottle	bus	car	cat	chair	cow
SPM	66.6	49.2	33.1	61.3	16.7	46.6	66.8	46.1	48.1	29.3
ScSPM	68.7	56.6	39.7	65.8	19.5	53.8	70.9	53.7	50.1	35.4
DLPC	66.9	56.3	40.3	63.0	21.7	56.9	72.9	52.1	50.0	35.9
DLPB	71.2	60.5	44.8	63.9	20.0	56.5	74.0	54.2	50.4	41.9

class	dtable	dog	horse	mbike	person	plant	sheep	sofa	train	tv	average
SPM	36.8	37.0	71.2	48.7	77.1	17.5	34.8	37.3	66.6	42.2	46.7
ScSPM	44.6	39.2	75.0	56.9	79.5	24.0	41.4	47.0	69.0	47.6	51.9
DLPC	46.6	40.8	75.3	60.0	81.4	25.4	39.4	48.4	70.7	47.3	52.6
DLPB	47.1	42.5	75.6	60.6	81.2	24.2	42.8	51.5	69.0	48.8	54.0

Table 4: Average precision for each object class from Pascal VOC 2007

Methods	Accuracy	Methods	Accuracy
ScSPM [29]	80.28±0.93	SPM ($k=400$)	81.76±0.47
SPM [11]	81.40±0.50	ScSPM ($k=3200$)	84.32±0.58
HC [28]	82.30±0.49	DLPC ($k=3200$)	83.80±0.32
HG [61]	85.20	DLPB ($k=3200$)	85.22±0.55

Table 5: Recognition accuracy (%) on 15 Scenes

rather marginal compared to the results obtained for Caltech-101 and Caltech-256. While extra local spatial information does help to increase the recognition accuracy, it might be less significant to recognize scenes where capturing global structure information is more essential. Nevertheless, DLPB performs competitively against other state-of-the-art techniques. Overall, with all five datasets, DLPB has consistently outperformed all baseline techniques.

5 Conclusion

This paper has introduced a novel sparse image model for image classification tasks, namely Directional Local Pairwise Bases (DLPB). The dictionary is learned efficiently by splitting into multiple smaller independent dictionaries each of which learns a set of bases specific to a particular directional relationship. The results on several challenging datasets have shown remarkable recognition accuracy improvement with DLPB especially in object categorization tasks. While inferring sparse codes with ℓ_1 regularization sparsity constraint is efficiently done by the feature-sign search algorithm [11], it is a non-linear optimization problem and is still slow. Therefore, in order to speed up the inference without compromising the recognition accuracy, we would like to explore approximate inference such as Predictive Sparse Decomposition [8] which infers the codes by applying a non-linear function on linearly transformed data. Another interesting avenue of research is to explore whether or not a distance based kernel presented in [23] has any merit, and combine both directional and distance information to capture richer spatial information than DLPB.

Acknowledgement

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

- [1] J. Amores, N. Sebe, and P. Radeva. Context-based object-class recognition and retrieval by generalized correlograms. *TPAMI*, 2007.
- [2] O. Boiman, E. Shechtman, and M. Irani. In defence of Nearest-Neighbor based image classification. In *CVPR*, 2008.
- [3] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *TIP*, 2006.
- [4] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results, 2007.
- [5] 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*, 2004.
- [6] G. Griffin, A. Holub, and P. Perona. Caltech 256 object category dataset. *Technical Report UCB/CSD-04-1366, California Institute of Technology*, 2006.
- [7] P. Jain, B. Kulis, and K. Grauman. Fast image search for learned metrics. In *CVPR*, 2008.
- [8] K. Kavukcuoglu, M. Ranzato, R. Fergus, and Y. LeCun. Learning Invariant Feature through Topographic Filter Maps. In *CVPR*, 2009.
- [9] S. Lazebnik, C. Schmid, and J. Ponce. A Maximum Entropy Framework for Part-Based Texture and Object Recognition. In *ICCV*, 2005.
- [10] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *CVPR*, 2006.
- [11] H. Lee, A. Battle, R. Raina, and A.Y. Ng. Efficient sparse coding algorithms. In *NIPS*, 2006.
- [12] H. Lee, R. Grosse, R. Ranganath, and A.Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML*, 2009.
- [13] H. Ling and S. Soatto. Proximity Distribution Kernels for Geometric Context in Category Recognition. In *ICCV*, 2007.
- [14] D. Liu, G. Hua, P. Viola, and T. Chen. Integrated feature selection and higher-order spatial feature extraction for object categorization. In *CVPR*, 2008.
- [15] D.G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 2004.

- [16] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative Learned Dictionaries for Local Image Analysis. In *CVPR*, 2008.
- [17] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *TIP*, 2008.
- [18] J. Mairal, M. Leordeanu, F. Bach, M. Hebert, and J. Ponce. Discriminative Sparse Image Models for Class-Specific Edge Detection and Image Interpretation. In *ECCV*, 2008.
- [19] N. Morioka and S. Satoh. Building Compact Local Pairwise Codebook with Joint Feature Space Clustering. In *ECCV*, 2010.
- [20] B.A. Olshausen and D.J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37, 1997.
- [21] R. Raina, A. Battle, H. Lee, B. Packer, and A.Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *ICML*, 2007.
- [22] M. Ranzato, F.J. Huang, Y. Boureau, and Y. LeCun. Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition. In *CVPR*, 2007.
- [23] S. Savarese, J. Winn, and A. Criminisi. Discriminative Object Class Models of Appearance and Shape by Correlatons. In *CVPR*, 2006.
- [24] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, and W.T. Freeman. Discovering objects and their location in images. In *ICCV*, 2005.
- [25] J.R.R. Uijlings, A.W.M. Smeulders, and R.J.H. Scha. Real-time Bag of Words. In *CIVR*, 2009.
- [26] J.C. van Gemert, C.J. Veenman, and J.M. Geusebroek. Visual Word Ambiguity. *TPAMI*, 2010.
- [27] J. Winn, A. Criminisi, and T. Minka. Object Categorization by Learned Universal Visual Dictionary. In *CVPR*, 2005.
- [28] J. Wu and J.M. Rehg. Beyond the Euclidean distance: Creating effective visual codebooks using the histogram intersection kernel. In *ICCV*, 2009.
- [29] J. Yang, K. Yu, Y. Gong, and T.S. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.
- [30] Y. Zhang and T. Chen. Efficient Kernels for identifying unbounded-order spatial features. In *CVPR*, 2009.
- [31] X. Zhou, N. Cui, Z. Li, F. Liang, and T.S. Huang. Hierarchical Gaussianization for Image Classification. In *ICCV*, 2009.
- [32] L. Zhu, Y. Chen, and A. Yuille. Unsupervised learning of Probabilistic Grammar-Markov Models for object categories. *TPAMI*, 2009.