Efficient Kernels Couple Visual Words Through Categorical Opponency

Ioannis Alexiou

Anil Anthony Bharath
http://www.bg.ic.ac.uk/research/a.bharath/

Biologically Inspired Computer Vision
Department of Bioengineering
Imperial College London, UK

Abstract

Recent progress has been made on sparse dictionaries for the Bag-of-Visual-Words (BOVW) approach to object recognition and scene categorization. In particular, jointly encoded words have been shown to greatly enhance retrieval and categorization performance by both improving dictionary sparsity, which impacts efficiency of retrieval, and improving the selectivity of categorization. In this paper, we suggest and evaluate different functions for the "soft-pairing" of words, whereby the likelihood of pairing is influenced by proximity and scale of putative word pairs. The methods are evaluated in both the Caltech-101 database and the Pascal VOC 2007 and 2011 databases. The results are compared against spatial pyramids using BOVW descriptions, standard BOVW approaches, and across different parameter values of pairing functions. We also compare dense and keypoint-based approaches in this context. One conclusion is that word pairing provides a means towards attaining the performance of much larger dictionary sizes without the computational effort of clustering. This lends it to situations where the dictionaries must be frequently relearned, or where image statistics frequently change.

1 Introduction

Many modern approaches to vision problems, such as object recognition and categorization, are based on a series of well-defined processing stages which produce image representations that support the task at hand. For categorization or object recognition, such implementations typically involve keypoint detection or dense sampling [4]. Commonly used keypoints detectors are the Harris and Hessian multiscale [11], MSER [10] and DoG [9]. These reduce the dense grid to a sparse representation, based around the keypoint locations and scales, yielding highly scalable performance. Recent research has shown that dense sampling can drastically increase the recognition performance through the inclusion of sparse coding and pooling strategies (e.g. max pooling) [1, 15].

Several techniques can be used to bind mid-level descriptor-based features into a compact representation whilst encoding spatial information. The methods for doing this include the well known spatial pyramids [6]. Learning networks using pairwise potentials between edge features [7] and other higher-order spatial features [16] can also be employed. Spatial pyramids provide a description of relatively coarse spatial relationships, producing good recognition rates [1, 6, 15]. Geometric relationships can also be encoded by approaches

http://dx.doi.org/10.5244/C.26.82

which hypothesize a fixed number of features (parts). Specifically, part-based models have shown that fixed numbers of descriptor-based components can be a powerful representation tool for object detection applications (see, for example, Felzenszwalb *et al.* [5]). These models have been tested in a variety of different arrangements by Crandall *et al.* [2]. Typical disadvantages of such models include having a fixed numbers of parts where a computationally expensive search assigns parts to locations and sets a hypothesised center for the model. Yet another approach binds descriptors into high-order features yielding much improved recognition performance compared to an ungrouped bag of features approach [16, 17]. The latter approach attempts to encode spatial relationships using relative features' distances (*correspondence transform*). Co-occurrences of these high-order features are mapped into an offset space [16], where occurrence counts are assigned to those features which satisfy predefined distance criteria. In this research, we focus on whether a selection mechanism can reduce the dictionary size of such high-order features. The transition from visual words to 2^{nd}, 3^{rd}, ..., k^{th} order grouping increases the computational complexity of BOVW methods by a factor of N^k, where N is the size of the visual word dictionary.

In this work, we first propose the use of kernels to create small-sized dictionaries of paired words utilizing *categorical opponency* to select such pairs. Secondly, we examine how word pairs in close relative proximity can be turned into a scalable indexing scheme, and how different coupling functions affect classification performance. Thirdly, we propose a method whereby *coupling kernels* are applied, suggesting decision functions that can detect pair occurrences online.

2 Coupling Visual Words

This work is partly inspired by evidence that binding low-level features can improve recognition rates and pose invariance. Additional evidence has been found by Leorndeanu, *et al.* [7] who used contour-like representations and Conditional Random Fields (CRFs) as a conveyor of spatial information, yielding improved classification. In addition, findings by Zhang, *et al.* [16] indicate that 2^{nd} order features produce high performance gains compared to simple Bag-Of-Visual-Word approaches. Higher orders than the second, especially for the case in which an SVM is used for final classification, show almost no further improvement [16]. Thus, this work is focussed on 2^{nd} order features, or visual word pairs. Note that this is different to concatenations of descriptor pairs, which would in principle need to be clustered in a higher (double) dimensional-space. There is no work in the literature on the exact size of *pair dictionaries* used by researchers, or of the effect on performance. This motivated us to explore small codebooks of pairs with discriminative power that is, at least in principle, equal to that of a full paired dictionary of order N^2. For instance, a codebook of 500 visual words can produce $500^2 = 250,000$ pair combinations, but some of these are repeated. Assuming symmetry in the pairs, this yields a paired combination of $N(N-1)/2 = 124,750$ unique pairs, or including the case that pairs with the same word IDs are allowed, $N(N+1)/2 = 125,250$ combinations can be generated. This is an inconveniently large size for a dictionary produced by clustering alone, which discourages implementation when the dictionary needs to be regenerated in on-line use. Thus, a key question is: to what extent can word pairing produce an effectively much greater dictionary size, and provide the performance gains usually associated with large vocabularies of single-word dictionaries [12] ?

2.1 A Dictionary of Paired Visual Words

Grouping visual words of a small dictionary size can exponentially increase the effective size of a visual vocabulary. Even paired words can yield numerous combinations, depending on the size of the initial codebook. Among these numerous pairs, a subset can exist that makes the pairing task more efficient. In addition, it is assumed that specific pairs are very unique to each class and others are non-informative. Thus, a *pair mining* method is described to select those pairs from the training set given the ground truth of the object classes.

We can treat pairing as estimating the joint occurrence of certain words, $w_i \cap w_j$, and consider the probability that these words occur jointly, $P(w_i \cap w_j)$. Bayes' Theorem suggests that $P(w_i \cap w_j) = P(w_i) \cdot P(w_j|w_i)$. However, we found it intuitive to use a kernelized form, K_h, which incorporates a term similar to a prior on individual word occurrence:

$$K_h(w_i, w_j) = P(w_j|w_i) \cdot G_i \tag{1}$$

and where $G_i = -\log(P(w_i))$ provides a weighting similar to an inverse-document frequency [13]. The kernelized expression (1) provides a statistical means to monitor specific visual words down-weighted by the G_i term as introduced by Sivic, *et al.* [13]. $P(w_j|w_i)$ refers to the probability that codebook member w_j has occurred, given that the word w_i has occurred in the image. We found that by stopping the bins corresponding to w_i in estimating $P(w_j|w_i)$ reduces bias in (efficiently) estimating the co-occurrence of other words in the image. Specifically, the conditional histogram of visual words is constructed by removing the candidate w_i word-bin from the histogram. This procedure favours the pairing of words with those other than itself. The conditional word estimation is performed per object, then the kernels are averaged over the same class and finally L_1 normalized, arranged according to object classes.

We now introduce the *categorization opponency* kernel, K_{Op},

$$K_{Op}(w_i, w_j) = \frac{\max\limits_{c_t} K_h(w_i, w_j, c_t)}{\max\limits_{c'_t \neq c_t} \left[K_h(w_i, w_j, c'_t) \leqslant \max\limits_{c_t} K_h(w_i, w_j, c_t) \right]} \tag{2}$$

c_t represents the object label assigned to each normalized kernel. Presumably, if a histogram of words can provide a rough discrimination among categories, then a pair might exist enhancing this behaviour. Equation (2) captures our approach to detect which pairs have high dominance within a specific class. Searching along the categories $c_t \in \{1, \dots, C\}$ the maximum of a kernel entry is divided by the second maximum, found in another category. This forms a ratio such that the higher the value of the RHS of Equation (2), the higher the *opponency* of this class against the others. This means that for a given combination of visual words, the detected pair tends to be unique to the examined class.

2.2 Decision Functions for Coupling

In the training and testing phase, histograms of pair occurrences are derived based on the pair codebook described in Section 2.1. Zhang, *et al.* [16] used a correspondence transform to build histograms of co-occurrences over predefined regions. However, their approach did not deal with the size variation of objects. For example, if salient patches on one object were to occur with a size such that multiple patches fall into the allowed offset space, a different result would be obtained if the object were scaled such that the relative distance

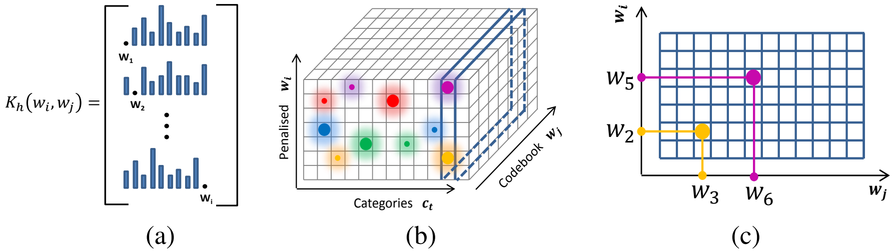


Figure 1: Starting from left to right: (a) illustrates how a kernel may be constructed using conditional histograms. (b) This figure shows the concept of first and second maxima. Each colour represents a different pair, where the size of the coloured/shaded blobs illustrates the 1^{st} and 2^{nd} maxima. (c) Once a candidate category is selected, we focus on the maximum votes that belong to this category. Finally, the ratios of these pairs are selected from highest to lowest.

between its salient patches was increased, despite the approximate scale-invariance of many descriptors. An alternative approach was used by H. Ling, *et al.* [☒], in which cumulative proximity distributions were employed. Both approaches lack the inclusion of scale-relative distances. So, the distances between descriptor pairs can vary as both the *pose* and *size* of the object change. Visual words, especially at the descriptor level, typically only convey information from a local region. However, the rough scale estimate provided by many of the approaches to building scale-invariant descriptors can be interpreted as the radius of a patch descriptor, and this should scale with patch zooms. A hypothesis can be made that these keypoint-associated scale estimates may provide reasonable scale predictions as one zooms into an object. In addition, the relative distances of visual words in space is trivially obtained. We thus propose that the coupling functions should be based on the relative scale coverage that two visual words have between them. For instance, two words might slightly overlap each other, suggesting that the gradient fields encoded by both also jointly encode an object contour. We therefore define coupling functions that relative distance to the effective descriptor radii. This transforms the relative overlap in image space into a probability of two co-occurring words forming a proximal pair.

An abstracted (*"distilled"*) feature $\phi_{w_j}^{(n)} = (x_n, y_n, \sigma_n, w_j)$ is defined by its (x_n, y_n) image location estimated by a detector while n represents the instance of a single word w_j in the image. The parameter σ_n is the scale estimate for the keypoint, and may be thought of as representing the descriptor radius. The w_j entry represents the index of the assigned visual word. Next, define $r = \sigma_n + \sigma_m$ as the summation of radii of two different features $\phi_{w_j}^{(n)}$ and $\phi_{w_i}^{(m)}$. Finally, the parameter $d = \sqrt{(x_n - x_m)^2 + (y_n - y_m)^2}$ represents the Euclidean distance between candidate features. Often, in on-line image pair assignment, several co-occurrences $\phi_{w_j}^{(n)} \cap \phi_{w_i}^{(m)}$ of the same words w_i and w_j may be found. A kernel of visual word co-occurrences is constructed in which the first dimension corresponds to the number of times word w_j is found (N) and the second dimension, the number of times word w_i is found (M). These are used in the construction of the word pairing histogram, $\mathcal{B}^{(p)}$, defined by:

$$\mathcal{B}^{(p)} = \sum_{m|n=1}^{Q} \max_{n|m=1}^{Q} K_{N \times M}^{(p)}(\phi_{w_j}^{(n)} = w_j, \phi_{w_i}^{(m)} = w_i), \text{ with } Q = \min(M, N). \tag{3}$$

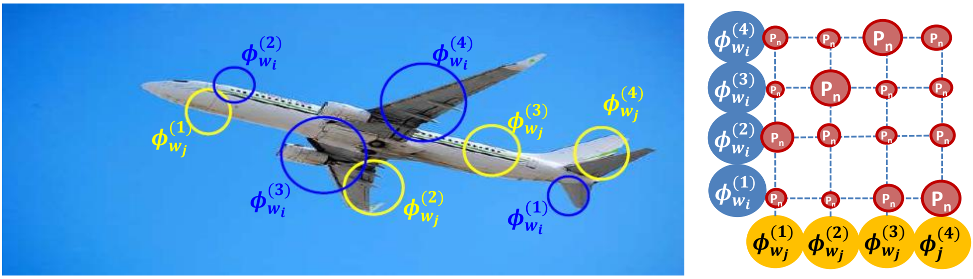


Figure 2: This figure shows two word co-occurrences before pairing. The size of the circles indicates the scale at those locations, and the colour represents a unique word ID. The pair detection is performed by flagging the word IDs of a single pair. At this point, several word co-occurrences might happen. The kernel on the right of the figure, as defined in Equation (3) arranges the co-occurrences and selects the maximum values along the largest dimension. The remainders are added as probabilistic votes into a histogram bin which represents the examined pair.

A maximization operation is applied along the larger dimension. This approach was found to yield the best suited soft-assigned pairs which, in turn, will be added as pair occurrences into histogram bins $\mathcal{B}^{(p)}$. The kernel mapping $K_{N\times M}^{(p)}$ is obtained by applying one of three pairwise coupling functions, S_1, S_2, S_3, independently; in Section (3.1) we compare the different coupling criteria to assess which one yields the best performance. It is assumed that a Support Vector Machine (SVM) classifier can learn a separating hyperplane that splits these classes based on $\mathcal{B}_p$. The first coupling function is defined as:

$$S_1(P_p|\phi_{w_j}^{(n)} \cap \phi_{w_i}^{(m)}) = d \tag{4}$$

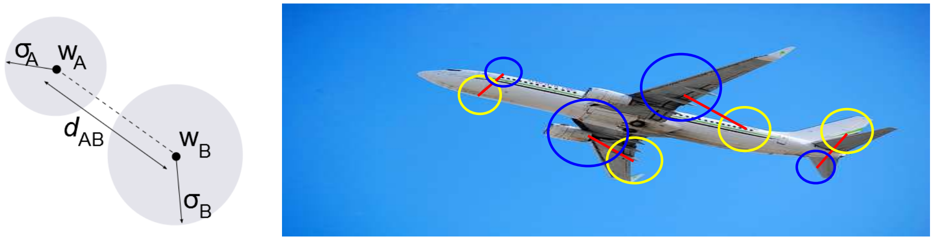
Equation (4) is merely the Euclidean distance between words. The effect of using this distance is to pair words that are adjacent to each other, irrespective of scale. The distances are accumulated in the corresponding pair bin. The votes can capture trends that might exist in distances between word pairs. The second function is a *scale-relative* distance:

$$S_2(P_p|\phi_{w_j}^{(n)} \cap \phi_{w_i}^{(m)}) = \max\left\{0, \frac{d-r}{r}\right\} \tag{5}$$

The max() operation in Equation (5) rejects negative outputs from the $\frac{d-r}{r}$ term. Occasionally, closely adjacent words might produce negative values. We wish to penalise such occurrences, based on the idea that two distinct words with large overlap provide little new information. Finally, we have a sigmoidal distance:

$$S_3(P_p|\phi_{w_j}^{(n)} \cap \phi_{w_i}^{(m)}) = \frac{1}{1+e^{\frac{d-\alpha\cdot r}{r}}} \tag{6}$$

Equation (6) expresses the likelihood that the pairs P_p, given that at least one $\phi_{w_j}^{(n)} \cap \phi_{w_i}^{(m)}$ co-occurrence happened. One advantage of using this function is that its output ranges always in $(0,1)$, which in turn reduces the dynamic range of the elements in $\mathcal{B}_p$. The raised term in the exponent represents the relative overlap of the two words. Also, the α parameter affects the slope of the sigmoid curves; the higher the α value, the further away in image space co-occurrences are paired.



(a) (b)

Figure 3: (a) illustrates a word pair comprised of two words w_A and w_B. The associated key-point information captures the effective descriptor (spatial) radius (σ_A and σ_B), and relative Euclidean distance (d_{AB}), which is used to derive the paired words. (b) illustrates the single word pair occurring 4 times; multiple occurrences such as this are used in Equation (3).

3 Experiments

The evaluation of the proposed approach starts with the size of a pair dictionary. In our experiments to determine the effect of dictionary size, we used the SIFT descriptor from VLFEAT [14] with both the built in detector and grid-based sampling. 128-dimensional SIFT descriptors were used, and for the case of the dense sampling, the patch size was 16×16 with 8-pixel spacing. For the first experiment, the single-word codebook was set to 500. The number of codewords in the paired codebook was then varied in size between 500 to just below 4000. One of the most important effects is on the *variance* of the classification accuracy with partial training sets. To measure this, a subset of training images (600 per class), was randomly sampled from the full Pascal VOC 2011 training database and used to learn a classification model with a linear SVM classifier [3]. We used the Pascal VOC 2011 set because it contains slighter greater pose changes across many of the classes. A sample of validation images was drawn (600 per class) and accuracy was tested. By repeating this process 10 times per class, an estimate of the standard deviation is obtained. We found that for dictionary sizes below 4000, the standard deviation is large, becoming greater as the number of word pairs is decreased. The standard deviation rose to as much as 20%, and this could significantly affect real-world performance. Details of the classifier training are as follows: half of the kernel entries were randomly selected positive examples of the training class and the other half were randomly selected negative examples. The overall size of the kernel is 600×600 entries balanced into positive and negatives which helps to obtain realistic accuracy estimates. Having determined that paired codebook sizes of beyond 4000 lead to standard deviations of less than 2.5%, we selected a paired size of 5000 for further experiments.

3.1 Evaluation of Coupling Functions

The performance of Equations (4), (5) and (6) is evaluated on the Pascal VOC 2011 as well. The same experimental setup, described above, is applied to assess which pairing kernels (S_1, S_2 or S_3) provide better performance.

	S_1 (4)	Equation (5) S_2	Equation (6) S_3
Average Accuracy	63.92%	65.93%	66.42%

Table 1: This table clearly shows that the best performance is obtained by S_3 ($\alpha = 1$).

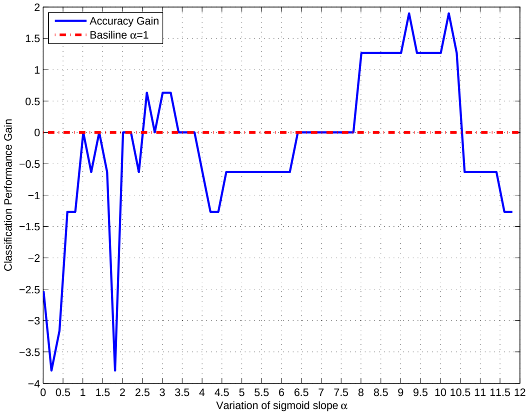


Figure 4: This is the only experiment in which the α parameter of Equation (6) was varied. A red line sets the baseline performance of the sigmoid function where the blue curve shows which values of α increase or reduce the accuracy. This test has been performed on the class "Cow" of Pascal VOC 2011. There is a clear trend for the specific range of $\alpha = (8, 10.5)$ which can yield better performance for this class. The nature of this effect is class specific.

3.2 Pascal VOC 2011

In this test, we compare the performance of the best performing coupling kernel (S_3) in all categories of the Pascal VOC 2011 dataset against the performance of spatial pyramids (Table 2), both keypoint and grid-based. The size of the histogram descriptors was 10500, using three levels (0,1, and 2) [6]. The best performing method is shown in each column in bold. In Table 3, we summarise the average performance and standard deviation of the runs across the whole database when either pairs or single words are used, both with keypoint and grid-based approaches. It is clear that the pairs also lend themselves to grid-based approaches, resulting in the highest performance in our tests.

Average Accuracy	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow
Keypoint-Based Pyramids (10500)	70.98	53.57	55.46	59.27	54.60	61.58	60.83	57.06	56.03	49.90
Grid-Based Pyramids (10500)	79.61	64.01	64.56	70.94	58.5	**79.78**	69.75	67.83	**68.45**	62.43
Keypoint-Based Pairs (5000)	77.06	65.38	**70.06**	68.30	59.06	73.60	61.23	66.76	68.06	**69.01**
Grid-Based Pairs (5000)	**80.80**	**70.91**	65.35	**71.16**	**64.48**	79.23	**71.43**	**70.85**	66.43	68.61

	D-Table	Dog	Horse	Motorbike	Person	P-Plant	Sheep	Sofa	Train	Tvmonitor
Keypoint-Based Pyramids(10500)	52.45	57.81	54.61	54.25	53.91	50.27	58.53	55.60	52.01	58.00
Grid-Based Pyramids(10500)	67.82	64.11	63.83	66.45	59.66	61.24	70.94	**66.80**	70.23	73.45
Keypoint-Based Pairs(5000)	**70.70**	64.70	55.30	**65.63**	**65.26**	59.57	65.58	63.32	67.41	72.35
Grid-Based Pairs(5000)	69.08	**65.68**	**66.18**	65.44	64.83	**61.64**	**74.02**	66.50	**73.80**	**74.22**

Table 2: Detailed performance evaluation on Pascal VOC 2011.

	Pyramids (10500)	Pairs (5000)
Keypoint-Based	$56.34_{\pm 1.63}$	$66.42_{\pm 2.07}$
Grid-Based	$67.52_{\pm 2.04}$	$69.53_{\pm 2.03}$

Table 3: Summary of table (2). These are the accuracies averaged over all classes, with standard deviation reflecting variability across all classes.

3.3 Caltech101

In these experiments, we test performance in a larger categorization database using keypoints and grid-based sampling with either a pair approach or a pyramid approach. In this case, we train on 30 examples and test on 50 randomly selected samples. The single-word dictionary size is again 500, with the paired-word dictionary being 5000. A linear SVM was used as the classifier. In the case that a category contains less than 50 samples, we test on the remaining number of samples. Figure 5 shows the classification accuracy. The word pairing significantly enhances results, which are summarised in Table (4).

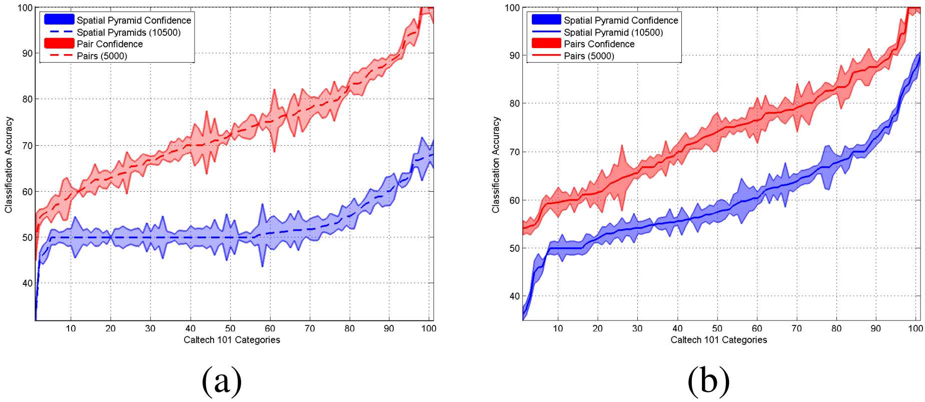


Figure 5: In (a) a comparison of spatial pyramids (blue) and pair (red) with descriptors produced by keypoint locations. (b) The same comparison is illustrated, but this time a grid sampling approach has been implemented. In both (a) and (b) the "confidence" represents the unit-standard deviation envelope estimated over 10 runs.

	Pyramids (10500)	Pairs (5000)
Keypoint-Based	$52.30_{\pm 1.28}$	$71.00_{\pm 2.00}$
Grid-Based	$60.16_{\pm 1.56}$	$73.88_{\pm 2.10}$

Table 4: Average accuracies per method for Caltech 101.

3.4 Pascal VOC 2007

We included a test on the Pascal 2007 database using a larger single codebook dictionary containing 4000 words. This test shows better performance in the spatial pyramid, and a comparable test of word pairing performance. The pyramid representation consisted of 84000-element descriptors, whilst 65000 word pairs were used. This was chosen as being roughly equivalent in performance improvement to the 5000 paired number relative to a 500 single codeword dictionary.

Average Accuracy	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow
Grid-Based Pyramids (84000)	73.65	60.92	60.19	71.22	**61.60**	60.94	65.13	57.10	66.65	57.79
Grid-Based Pairs (65000)	**78.50**	**66.23**	**64.48**	**74.79**	61.48	**73.33**	**74.08**	**62.35**	**70.06**	**70.31**
	D-Table	Dog	Horse	Motorbike	Person	P-Plant	Sheep	Sofa	Train	Tvmonitor
Grid-Based Pyramids(84000)	**68.10**	58.38	60.40	56.98	60.76	57.12	62.93	60.06	62.47	61.94
Grid-Based Pairs(65000)	62.92	**58.45**	**63.92**	**64.21**	**61.05**	**64.59**	**68.50**	**61.99**	**68.80**	**66.85**

Table 5: Performance on Pascal VOC 2007. Overall, the pair approach achieves 66.84% against 62.21% for the spatial pyramids.

4 Conclusions and Future Work

The aim of this work was to a) evaluate the effect of different sizes of word-pairing dictionary well below that theoretically provided by the approximate upper number of word-pair combinations, where the subset of pairs is chosen by category opponency b) to investigate different pairing functions and c) to construct a pair histogram representation efficiently, as the visual words and locations are produced for a query image. In addition to proposing and describing the approach, we investigated the performance of classification in a number of standard datasets, including Pascal VOC 2007, Pascal VOC 2011 and the Caltech 101 database.

4.1 Implications for Classification Performance

Category opponency makes use of labelled categories to select word pairs that are more suitable for category discrimination. The results show that the use of category opponency can decrease the number of word-pairs, that are actually employed, reducing those from a theoretical vocabulary of at most $N(N+1)/2$ to a size that is tunable by the user. This allows storage space to be reduced in representing histograms of word pairs. The results show that the number of words can be flexibly chosen, but very small word-pair combinations can lead to high variance (=low reproducibility) in learning performance.

4.2 Coupling Functions

The coupling functions assign a weight to each paired histogram bin based on proximity of two paired words and the scale of the words. We investigated three types of pairing functions, S_1, S_2 and S_3. The results were best using S_3, a sigmoidal function with one parameter which was fixed for all but one experiment.

4.3 Overall Performance

Empirically, the pairs have the ability to improve keypoint performance to the point that it is close to that of grid-based methods. This is an important saving: rather a than dense sampling of descriptors, or very large vocabulary sizes (though expensive clustering with a large number of cluster centres), word pairs allow a computationally efficient and flexible way of using the same single-word codebook to effectively achieve much higher vocabulary sizes. The fact that combinations of word pairs can be selected to boost categorization performance helps keep the vocabulary size low, yet with good performance.

4.4 Future Work

In future work, we intend to merge pyramids with word pairs to investigate whether there are further improvements in performance. In addition, recent approaches to sparse coding with max pooling [1] has shown major performance gains with respect to hard-word assignment. This encourages us to explore sparse coding on structured dictionaries such as in our case where two words form a pair: there is a hidden structure that we have not utilized yet of hierarchical information derived from our approach. For example, if two words form a pair then this can be treated as a rough tree structure comprised of a root node (pair) and two children (two words).

References

[1] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010. IEEE Conference on*, pages 2559–2566, 2010. doi:10.1109/CVPR.2010.5539963.

[2] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 10 – 17 vol. 1, june 2005. doi:10.1109/CVPR.2005.329.

[3] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, 2008. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1390681.1442794.

[4] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition (CVPR), 2005. IEEE Computer Society Conference on*, volume 2, pages 524–531, 2005. doi:10.1109/CVPR.2005.16.

[5] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition (CVPR), 2008. IEEE Conference on*, pages 1–8, 2008. doi:10.1109/CVPR.2008.4587597.

[6] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition (CVPR), 2006. IEEE Computer Society Conference on*, volume 2, pages 2169–2178, 2006. doi:10.1109/CVPR.2006.68.

[7] M. Leordeanu, M. Hebert, and R. Sukthankar. Beyond local appearance: Category recognition from pairwise interactions of simple features. In *Computer Vision and Pattern Recognition (CVPR), 2007. IEEE Conference on*, pages 1–8, 2007. doi:10.1109/CVPR.2007.383091.

[8] H. Ling and S. Soatto. Proximity distribution kernels for geometric context in category recognition. In *Computer Vision, 2007 (ICCV 2007). IEEE 11th International Conference on*, pages 1–8, 2007. doi:10.1109/ICCV.2007.4408859.

[9] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60:91–110, 2004. doi:10.1023/B:VISI.0000029664.99615.94.

[10] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004. ISSN 0262-8856. doi:10.1016/j.imavis.2004.02.006.

[11] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60:63–86, 2004. ISSN 0920-5691. doi:10.1023/B:VISI.0000027790.02288.f2.

[12] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2161–2168, 2006. doi:10.1109/CVPR.2006.264.

[13] J. Sivic and A. Zisserman. Video Google: a text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings of the 9th IEEE International Conference on*, volume 2, pages 1470 –1477, Oct. 2003. doi:10.1109/ICCV.2003.1238663.

[14] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[15] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3360–3367, 2010. doi:10.1109/CVPR.2010.5540018.

[16] Y. Zhang and T. Chen. Efficient kernels for identifying unbounded-order spatial features. In *Computer Vision and Pattern Recognition (CVPR), 2009. IEEE Conference on*, pages 1762–1769, 2009. doi:10.1109/CVPR.2009.5206791.

[17] Yimeng Zhang, Zhaoyin Jia, and Tsuhan Chen. Image retrieval with geometry-preserving visual phrases. In *Computer Vision and Pattern Recognition (CVPR), 2011. IEEE Conference on*, pages 809–816, 2011. doi:10.1109/CVPR.2011.5995528.