

Bag-of-Features Kernel Eigen Spaces for Classification

Gaurav Sharma¹ Santanu Chaudhury² J.B. Srivastava¹
¹Dept. of Mathematics, ²Dept. of Electrical Engg.
Indian Institute of Technology, Delhi
santanuc@ee.iitd.ac.in

Abstract

We present a classifier unifying local features based representation and subspace based learning. We also propose a novel method to merge kernel eigen spaces (KES) in feature space. Subspace methods have traditionally been used with the full appearance of the image. Recently local features based bag-of-features (BoF) representation has performed impressively on classification tasks. We use KES with BoF vectors to construct class specific subspaces and use the distance of a query vector from the database KESs as the classification criteria. The use of local features makes our approach invariant to illumination, rotation, scale, small affine transformation and partial occlusions. The system allows hierarchy by merging the KES in the feature space. The classifier performs competitively on the challenging Caltech-101 dataset under normal and simulated occlusion conditions. We show hierarchy on a dataset of videos collected over the internet.

1 Introduction

We present a classifier which combines subspace method with local features based representation. We also give a formulation for merging eigen spaces in feature space. The classifier is invariant to scale, illumination and affine changes in the object. The use of local features enables the method to counter substantial occlusions. We incorporate hierarchy by merging kernel eigen spaces in feature space. This leads to a forest of trees classifier, used for video object categorization, where multiple views of the object are available.

We propose to use local features based representation with subspace learning. Meltzer et al. [8] used kernel PCA to learn multiple view *feature descriptors* for wide baseline matching under deformations and transformations. We are constructing ‘object (BoF) spaces’ to represent classes and not feature (appearance) spaces.

Hence, we differ in objective from their work. Erichorn and Chapelle [1] studied various set matching measures for object classification, kernel principal angles (PA) being one of them. Using PAs is effectively comparing subspaces spanned by the set of features. It is a known fact that object classes share features. So the subspaces spanned by the feature vectors will have high overlap. Our object spaces are subspaces, not in the space spanned by the feature vectors but in a more discriminative BoF space. Hence, our method of subspaces in BoF space is expected to perform better. Among other local features based methods, Grauman and Darrell [4] use a low distortion embedding of Earth Mover’s distance and then use approximate nearest neighbor search in the embedded space.

2 Our approach

In bag-of-features (BoF) representation of the images, first, local regions extracted are clustered to give the analogous *visual words*. Each image is encoded as a histogram over these visual words, with features vector quantized to these words.. The local region detector we use for our classifier is the recently developed affine covariant region (ACR) detector [9] and then we describe the local appearances using the well established SIFT [7] descriptor. The use of ACR-SIFT¹ based feature representation makes our approach invariant to illumination, rotation, scale and affine transformations.

Intuitively (Fig. 1) a given class is expected to have few dominant visual words e.g. sunflower will have a dominant dimension corresponding to the high frequency of ‘petal’ words while the leopard class will have that corresponding to the numerous ‘spots’ in the image. Capturing these class specific dominant dimensions by fitting object spaces can thus be used for classification of a new BoF (object) to one of the two classes. Since the number of categories are large (e.g. 101 in

¹We acknowledge the use of binaries from <http://www.robots.ox.ac.uk/~vgg/research/affine/>

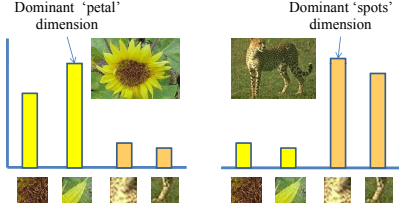


Figure 1. Visualization of BoF.

Caltech-101) and the dimension of the space is limited we use kernel mapping to improve performance.

Recently, Kernel methods [10] have been used for improving the performance of many algorithms. Kernel method maps the input space to a higher dimensional feature space. We use the kernel mapping to map our subspaces in the high dimensional feature space to increase their separability.

2.1 KES classifier

In a supervised classification problem, we are provided with class-level labeled training data. Our assumption is that in the BoF space the object classes will lie on subspaces. To extract the object spaces, we use the training images for that class and do kernel PCA [10] on them. Our KES classifier is thus the collection of kernel eigen spaces

$$\mathcal{C} = \{\Omega_j = (X_j, \alpha_j, \mu_j, n_j) | 1 \leq j \leq M\} \quad (1)$$

where, M is the number of classes, X_j the matrix with columns as the class j training vectors, α_j is the matrix with column vector as coefficients of the kernel PCs and μ_j is that of the class mean while n_j the number of training examples for that class.

The kernel used for mapping the input space to feature space is the radial basis function (rbf) kernel $K(x, y) = e^{-\frac{1}{\sigma^2} \|x-y\|^2}$. The parameter σ was adjusted experimentally and a value of 25-30 gave the best performance. The criteria for classification of a new query vector y to one of the database KES is

$$\min_{\Omega \in \mathcal{C}} \text{RE}(\Omega, y) = \|\tilde{y} - \phi(X)\alpha\alpha^T\phi(X)^T\tilde{y}\|^2 \quad (2)$$

$$\text{where, } \tilde{y} = (\phi(y) - \phi(X)\mu) \quad (3)$$

where, $\text{RE}(\Omega, y)$ is the reconstruction error of y w.r.t. KES Ω .

3 Hierarchical classifier

While detecting objects in a video sequence, we have multiple views of the object. We use the various views

from the frames of the object to construct instance level KES. Then for different instances of the same object (e.g. for different videos of ‘dog’ category) we merge the KESs (with a novel algorithm outlined in next section) to get a category KES. Thus the final hierarchical classifier here is a forest of trees (Fig. 2) with the leaf nodes capturing the instance level information and higher nodes capturing that of the categories.

In this case, for recognizing a new video object, we again form its KES and then compare with database KESs using kernel principal angles [13]. Note, this a subspace to subspace distance measure and not the traditional vector to subspace distance measure. The distance measure we use here is the smallest kernel principal angle between the subspaces, which measure the smallest angle between any two unit vectors belonging to the two subspaces respectively.

3.1 KES merging

We merge the KES by kernalizing [5]. For two eigen space (\bar{x}, U, Λ, N) (mean, principal components, principal values and number of vectors respectively) and (\bar{y}, V, Δ, M) , the final approximated eigen value problem is,

$$\frac{N}{P} \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} + \frac{M}{P} \begin{bmatrix} G\Delta G^T & G\Delta\Gamma^T \\ \Gamma\Delta G^T & \Gamma\Delta\Gamma^T \end{bmatrix} + \quad (4)$$

$$\frac{NM}{P^2} \begin{bmatrix} gg^T & g\gamma^T \\ \gamma g^T & \gamma\gamma^T \end{bmatrix} = R\Pi R^T$$

Where, $P = N + M$, $G := U^T V$, $\Gamma := \nu^T V$, $g := U^T(\bar{x} - \bar{y})$ and $\gamma := \nu^T(\bar{x} - \bar{y})$. The terms after a feature space mapping ϕ reduce to,

$$G := U^T V = \alpha^T \Phi_x^T \Phi_y \beta \quad (5)$$

$$\Gamma := \nu^T V = \xi^T \Phi_{xy}^T \Phi_y \beta \quad (6)$$

$$g := U^T(\bar{\phi}(x) - \bar{\phi}(y)) = \alpha^T \Phi_x^T \Phi_x 1_N - \alpha^T \Phi_x^T \Phi_y 1_M \quad (7)$$

$$\gamma := \nu^T(\bar{\phi}(x) - \bar{\phi}(y)) = \xi^T \Phi_{xy}^T \Phi_x 1_N - \xi^T \Phi_{xy}^T \Phi_y 1_M \quad (8)$$

where, Φ_x denotes the ϕ mapped x -vectors as columns, α and β are the coefficients of principal components of the two KES (see [11]), 1_k denotes a column vector with each entry $1/k$ and ξ is obtained by orthonormalizing $\nu = \Phi_{xy}\lambda$ (λ as given below in eqn. (9)), by slight modification of [13]. After a few algebraic manipulations we get λ as,

$$[H, h] = \Phi_{xy}\lambda, \lambda := \begin{bmatrix} -\alpha\alpha^T\Phi_x^T\Phi_y\beta & \epsilon \\ \beta & -1_M \end{bmatrix} \quad (9)$$

$$\epsilon := 1_N - \alpha\alpha^T\Phi_x^T\Phi_x 1_N + \alpha\alpha^T\Phi_x^T\Phi_y 1_M \quad (10)$$

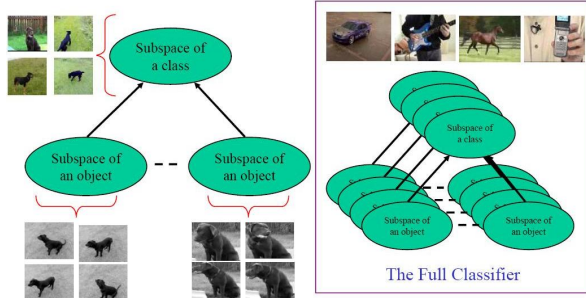


Figure 2. Forest of trees classifier for video object categorization.

Hence all the terms in eqn. (4) may be computed in the feature space using kernel trick which enables us to merge eigen spaces in feature space. (Due to space constraints, we have only given the salient computations above).

4 Experimental Results

We used Caltech-101 [2] to evaluate our classifier framework. It is a standard challenging dataset being used to evaluate classifiers. There is significant within class variability in terms of appearance and viewpoint which makes this database challenging.

We evaluated hierarchical classifier on a database that has videos of 6 object classes, *horse, guitar, dog, car, bicycle, phone*, Fig. 3(c), with two to four distinct objects per class. The video shots were segmented manually so as to have the object in all frames. Also in case of extensive clutter, a conservative tracker was used to get the object of interest from the video.

4.1 KES classifier results

We applied the KES classifier on the full Caltech-101 dataset excluding the background class. For making our results comparable to previous results, we used the same methodology (as in [14]) of 30 training and (up to) 30 testing images per class randomly sampled for each iterations (reporting average of 10 runs).

Table 1 compares the results obtained by our approach with the various other methods. Our method provides competitive result w.r.t. the other approaches. Also, our method is very simple and due to mostly matrix calculations it is fast.

Next, we simulated occlusions on the Caltech-101 dataset. The classifier was trained with full images, but, while testing we randomly removed interest points from

Table 1. Caltech-101 result comparison.

Method/Ref.	Accuracy	Std. dev.
SVM-KNN [14]	66.2 %	± 0.5
Proposed KES classifier	65.5 %	± 0.7
Local dist. func. [3]	65.2 %	-
SPM [6]	64.6 %	± 0.8

Table 2. Caltech-101 with occlusions.

% occlusion	Worst case accuracy drop
10 %	0.5 %
20 %	1.6 %
30 %	3.9 %

a certain fraction of image area (position randomly chosen). Fig. 3(b) shows example with 30% image area removed. Table 2 shows the worst case drop in accuracy (with 10 trials for each case). Caltech-101 objects do not occupy whole area in the image. So, we removed interest points from up to 30% of the image area randomly sampled in the center half of the image. With as much as 30% loss, the performance decreased by less than 4%. We are reporting the *worst* case we observed from 10 trials each. We thus conclude that the method is able to handle occlusions, where the traditional subspace methods would deteriorate substantially.

4.2 Hierarchical classifier results

We first trained each object with 30 views of the object, randomly selected from all frames, forming the eigen space of each object instance, e.g. labrador, doberman e.t.c. Then we merged the eigen spaces of each object to get the eigen space of the object class e.g. dog. When done for all the database object this gives us a forest of trees, Fig. 2. We report the average of 10 trials randomly selecting training and testing videos (mutually disjoint). For categorization we again selected 30 views of the test object (not used in training) at random. For recognition we took one of the object used for training but used frames not used in training. We used the smallest kernel principal angle [13] to find the subspace closest to the test object subspace. Table 3 shows the results for both categorization and classification. The classifier gives around 80% categorization accuracy and a very good recognition accuracy. We also performed a simple voting based categorization with the same data, wherein we classify each frame and then choose the class with the maximum number of frame classifications as the final video object class. The subspace distance based classification performed equal

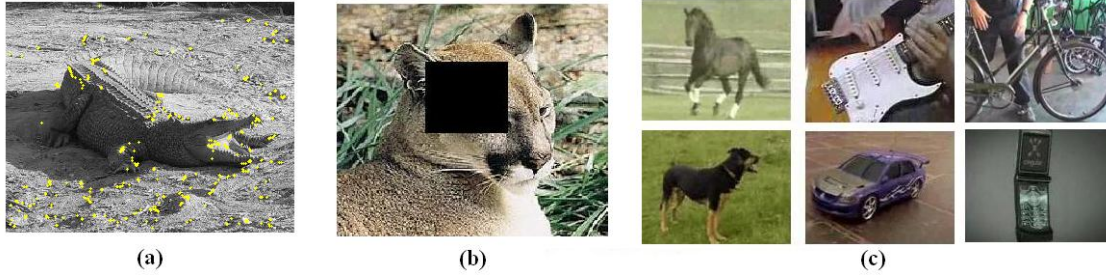


Figure 3. Instance correctly classified (a) despite interference from background, (b) despite occlusion (shown as black patch). (c) Examples from the video object dataset.

Table 3. Hierarchical classifier results

Description	Accuracy
Correctly categorized	80.2%
Correctly recognized	92.4%

or better to the voting scheme. This indicates that the subspaces are able to capture the class (and instance) properties correctly or better than a collection of frames.

5 Discussion and Conclusion

All of the methods for classification use only one (or few) of the many available cues e.g. appearance, shape, texture, color etc. Using all the cues together is surely required for achieving the best possible results. In a recent work Varma et al. [12] combined many appearance, shape and color based cues to obtain 87.8% on Caltech-101, by far the best result reported on the dataset. Our method uses only intensity based information giving encouraging performance. A natural extension would thus be to incorporate more information (e.g. color, texture, shape etc.) into our framework.

To conclude, we presented a kernel method based classifier which combines local features based representation and subspace learning concepts. Using local features makes the approach invariant to factors such as illumination, scale and affine transformations. It also makes the approach robust to substantial partial occlusions. We evaluated the approach on the challenging Caltech-101 dataset (original and with simulated occlusions). We also demonstrated the use of the framework for video object recognition. Our approach uses only the intensity information from the images. Incorporating more information based on shape, color, texture etc. is a challenging future work we would like to pursue.

References

- [1] J. Eichhorn and O. Chapelle. Object categorization with SVM: kernels for local features. *Technical report, MPI for Biological Cybernetics*, 2004.
- [2] 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. *WGMBV in CVPR*, 2004.
- [3] A. Frome, Y. Singer, and J. Malik. Image retrieval and classification using local distance functions. *NIPS*, 2006.
- [4] K. Grauman and T. Darrell. Efficient image matching with distributions of local invariant features. *CVPR*, 2005.
- [5] P. Hall, D. Marshall, and R. Martin. Merging and splitting eigen space models. *PAMI*, 2000.
- [6] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *CVPR*, 2006.
- [7] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.
- [8] J. Meltzer, M.-H. Yang, R. Gupta, and S. Soatto. Multiple view feature descriptors from image sequences via kernel principal component analysis. *T. Pajdla and J. Matas(Eds.): ECCV*, 1:215–227, 2004.
- [9] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004.
- [10] K.-R. Müller, M. Mika, G. Ratsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Trans. NN*, 12(2):181–201, 2001.
- [11] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.
- [12] M. Varma and D. Roy. Learning the discriminative power-invariance trade-off. *ICCV*, 2007.
- [13] L. Wolf and A. Shashua. Learning over sets using kernel principal angles. *JMLR*, 4:913–931, 2003.
- [14] H. Zhang, A. C. Berg, M. Maire, and J. Malik. SVM-KNN: discriminative nearest neighbor classification for visual category recognition. *CVPR*, 2007.