

# Learning Visual Dictionaries and Decision Lists for Object Recognition

Wei Zhang, Thomas G. Dietterich  
Oregon State University  
zhangwe,tgd@eecs.oregonstate.edu

## Abstract

*Visual dictionaries are widely employed in object recognition to map unordered bags of local region descriptors into feature vectors for image classification. Most visual dictionaries have been constructed by unsupervised clustering. This paper presents an efficient discriminative approach, called Iterative Discriminative Clustering (IDC), for dictionary learning. In this approach, each dictionary entry is defined by a representative value and a learned distance metric. In IDC algorithm, the dictionary entries are initialized by unsupervised clustering and then locally adapted to improve their discriminative power. Motivated by studies of the characteristics of individual dictionary entries, we employ bagged decision lists (BDL) as our image classifier in order to explore the conjunctions of small number of informative dictionary entries for classification. Experiments on benchmark object recognition datasets show that the system based on the new discriminative dictionaries and BDL classifier give performance comparable or superior to the state-of-art generic object recognition approaches.*

## 1. Introduction

Recognizing objects in natural scenes is a fundamental problem in computer vision. Recently, significant advances have been obtained through the use of interest region detectors that can find salient regions sparsely distributed in images despite variation [6,7,12]. Each extracted region is typically represented as a vector of descriptors. The result is that the original image is transformed into a bag of region descriptor vectors. The object recognition problem thus reduces to the problem of classifying a bag of descriptor vectors into one of the possible object classes.

The purpose of a visual dictionary is to provide a way of generalizing the descriptor vectors. In previous dictionary learning work, the dictionary is constructed

by pooling all of the descriptor vectors and applying an unsupervised clustering algorithm. Each cluster defines a dictionary entry, and all descriptors in that cluster are treated as equivalent (or similar). Recently, some researchers have begun to introduce discriminative mechanisms into the dictionary learning process, for example, the generative/discriminative dictionary learning methods in [5,9]. In any case, once a dictionary is constructed, one typical way to build a classifier is to convert the bag of descriptor vectors for each image into a fixed-length image feature vector whose  $i$ -th element is the vector quantization or mapping of the descriptors according to the dictionary entry  $i$ . Standard learning algorithms have been applied to these feature vectors to train the image classifier.

There are two major challenges for any approach to generic object recognition: (a) Small training sets. (b) Low signal-to-noise ratio. For generic recognition problems, usually only a small fraction of the extracted descriptors are discriminative, while the others are noisy. This is the main motivation for developing dictionary methods that exploit the class labels to identify discriminative features. The challenge is to do this without causing overfitting.

In this paper, we introduce a new efficient dictionary learning method that combines the best of generative and discriminative learning to address the challenges in generic object recognition. Our method benefits from generative initialization in its robustness to overfitting; and obtains higher test set accuracy from discriminative learning. Unlike previous methods, which treat all elements of the descriptor vector as equally important, our method learns a cluster-specific full-rank distance metric that improves cluster generalization and discriminative power. The dictionary is learned on sparsely detected interest regions rather than densely sampled regions, so it is much more efficient to compute. In addition, we employ an efficient rule learning algorithm, decision list, to create the final classifier. This classifier can achieve low bias by iden-

tifying the logic conjunctions of small number of informative dictionary entries from the highly noisy feature pool. Extensive experiments on three well-recognized object recognition benchmark datasets show performance that matches or exceeds state-of-the-art dictionary and instance selection based object recognition approaches.

## 2. The method

Our method is composed of two major parts: discriminative visual dictionary learning and image classifier learning. It consists of the following steps:

### I. Discriminative visual dictionary learning

1. *Extract region descriptors*: *HesAff* [7], *Salient Regions* [6] and *Curvilinear* [12] detectors are applied to detect distinctive interest regions. Each region is described by a ‘‘Steerable Filters’’ descriptor [4] that summarizes the local image contents in the region.

2. *Generative dictionary initialization* (Sec 2.1): the visual dictionary entries are initialized by K-means clustering of the extracted descriptor vectors.

3. *Discriminative dictionary learning* (Sec 2.2): the values and the distance metrics of the dictionary entries are learned by locally optimizing their discriminative power on training images.

### II. Image classifier learning

4. *Feature mapping* (Sec 2.3): map the bags of descriptor vectors to image feature vectors based on the learned dictionary.

5. *Image classifier learning* (Sec 2.4): learn bagged decision lists classifier on the training image features.

## 2.1. Generative initialization of the dictionary

A ‘‘visual dictionary’’ is a set of prototypes that relate region descriptors in query images to the ones previously seen in training images. Here we define a discriminative visual dictionary (*DVD*) as

$$DVD = \{\mathbf{E}_1, \dots, \mathbf{E}_k, \dots, \mathbf{E}_K\} = \{ \langle \mathbf{x}_1, \mathbf{W}_1 \rangle, \dots, \langle \mathbf{x}_k, \mathbf{W}_k \rangle, \dots, \langle \mathbf{x}_K, \mathbf{W}_K \rangle \} \quad (1)$$

where  $\mathbf{x}_k$  and  $\mathbf{W}_k$  are the representative value and the distance metric, respectively, of dictionary entry  $\mathbf{E}_k$ . A separate dictionary  $DVD_{c,f}$  is learned for each object class  $c$  and each channel  $f$  (i.e. a detector/descriptor combination). In our method, the dictionary  $DVD_{c,f}$  is initialized by K-means clustering on the region descriptor vectors of type  $f$  from the training images of class  $c$ . Full rank covariance matrices and the Mahalanobis distance are employed during clustering. Each representative value,  $\mathbf{x}_k$ , is initialized to the corresponding cluster center, and each distance metric,  $\mathbf{W}_k$ , is initialized to the inverse of the corresponding covariance matrix. So the distance from an initial dictio-

nary entry  $\mathbf{E}_k$  to a descriptor vector  $\mathbf{x}$  is measured by the Mahalanobis distance metric:

$$d(\mathbf{E}_k, \mathbf{x}) = ((\mathbf{x}_k - \mathbf{x})^t \mathbf{W}_k (\mathbf{x}_k - \mathbf{x}))^{1/2} \quad (2)$$

## 2.2. Discriminative learning of the dictionary

Previous methods construct large universal dictionaries to capture relevant variation of object parts for all the object categories. The dictionaries are usually learned from unlabeled images of a large set of categories. But in real-world applications, it is common that the universal dictionary is suboptimal and not discriminative enough for a specific problem [9]. The main contribution of our dictionary learning method is to apply supervised learning directly to construct problem-specific discriminative dictionaries for image classification. Our *Iterative Discriminative Clustering (IDC)* algorithm combines and adapts the idea of the *EM-DD* [11] algorithm for multiple-instance learning and *Relevant Component Analysis (RCA)* [10] for distance metric learning.

The IDC algorithm is applied separately to each entry  $\mathbf{E}_k = \langle \mathbf{x}_k, \mathbf{W}_k \rangle$  in a dictionary. Let  $c$  be the class of the dictionary. Consider a training image  $i$  represented by its bag of region descriptor vectors  $\mathbf{B}_i$ . Let  $\mathbf{p}$  be the descriptor vector in  $\mathbf{B}_i$  that is closest to  $\mathbf{E}_k$  according to  $d(\mathbf{E}_k, \mathbf{B}_{ij})$ ; we will call  $\mathbf{p}$  the nearest neighbor point from image  $i$ . Let  $\{\mathbf{NN}^+\}_k$  be the set of all nearest neighbor points  $\mathbf{p}$  for dictionary entry  $k$  drawn from positive examples of class  $c$ , and let  $\{\mathbf{NN}^-\}_k$  be the corresponding set drawn from negative training examples (i.e., examples of other classes  $c' \neq c$ ). If  $\{\mathbf{NN}^+\}_k$  is compact and well-separated from  $\{\mathbf{NN}^-\}_k$ , then  $\mathbf{E}_k$  is a compact, discriminative entry, because it has consistent nearest neighbor points in images of class  $c$ , and it is far away from the images of other classes. Otherwise, if  $\{\mathbf{NN}^+\}_k$  has high variance or  $\{\mathbf{NN}^+\}_k$  and  $\{\mathbf{NN}^-\}_k$  overlap, then  $\mathbf{E}_k$  is suboptimal in term of discrimination, and we seek to improve its performance with supervised learning.

The idea of the learning algorithm is to locally adapt the representative value and the distance metric of entry  $\mathbf{E}_k$  with the goal of making it discriminative. We limit the adaptation to the local neighborhood of entry  $\mathbf{E}_k$  to avoid situations in which all dictionary entries converge to the same global maximum and the learned dictionary has low discriminative power. The pseudocode for the IDC algorithm is given in Fig 1. IDC algorithm iterates between the following two steps: in the ‘‘nearest neighbors search’’ step, the nearest neighbor point sets  $\{\mathbf{NN}^+\}_k$  and  $\{\mathbf{NN}^-\}_k$  are computed for dictionary entry  $\mathbf{E}_k$  based on the representative value and distance metric from the previous iteration. In the following ‘‘entry updates’’ step, the representative value of  $\mathbf{E}_k$  is updated to be the mean of positive nearest

neighbor points; and the RCA algorithm is used to learn a new distance metric to sphere and better separate the point sets  $\{NN^+\}_k$  and  $\{NN^-\}_k$ . These two steps iterate until convergence.

RCA learns a linear transformation to assign large weights to the relevant dimensions and small weights to the irrelevant dimensions. Here the “relevant dimensions” are the dimensions that help to discriminate between the sets:  $\{NN^+\}_k$  and  $\{NN^-\}_k$ . The adaptation of a dictionary entry is limited to its local neighborhood using early-stopping conditions. IDC algorithm described above is applied to each entry in the dictionary. It is usual that this causes several entries to converge to the same point. In this case, only one of them is kept to compress the size of the dictionary. We tested the performance of the recognition system using different settings of the dictionary learning parameters. The performance is quite robust. The details of the algorithm will be provided in supplementary material.

### 2.3. Feature mapping based on the dictionary

A separate dictionary is learned for each object class (including the background class) and each channel. All these separate dictionaries are concatenated to construct the final dictionary; suppose it has  $M$  entries. Then a new image  $\mathbf{B}_i = \{\mathbf{B}_{ij}; j=1, \dots, n_i\}$  is mapped to a image feature vector  $\mathbf{V}_i$  according to its minimum distances to the dictionary entries, that is:

$$\mathbf{V}_i = \begin{bmatrix} \min_{j=1, \dots, n_i} d(\mathbf{E}_1, \mathbf{B}_{ij}) \\ \min_{j=1, \dots, n_i} d(\mathbf{E}_2, \mathbf{B}_{ij}) \\ \vdots \\ \min_{j=1, \dots, n_i} d(\mathbf{E}_M, \mathbf{B}_{ij}) \end{bmatrix} \quad (3)$$

### 2.4. Image classifier: bagged decision lists

Based on the learned dictionary, we have chosen bagged decision lists as our classifier, which combines and adapts the boosting feature selection method in [8] and the idea of cascaded classifiers framework. Decision List classifier is better suited to this problem than other standard classifiers for two reasons. First, decision lists favor situations in which the conjunctions of a small number of features are capable of discriminating the two classes, which matches our experience and the results in [8]. Second, decision lists have low bias—they can adjust their complexity as necessary to fit the data.

**Input:** Bags of descriptor vectors of  $m$  training images:

$$\mathbf{D} = \{\langle \mathbf{B}_1, l_1 \rangle, \dots, \langle \mathbf{B}_i, l_i \rangle, \dots, \langle \mathbf{B}_m, l_m \rangle\};$$

Initial dictionary of class  $c$ :  $DVD =$

$$\{\langle \mathbf{x}_1, \mathbf{W}_1 \rangle, \dots, \langle \mathbf{x}_k, \mathbf{W}_k \rangle, \dots, \langle \mathbf{x}_K, \mathbf{W}_K \rangle\};$$

**Learning:**

for ( $k = 1; k \leq K; k++$ )

$$\mathbf{E}_k = \langle \mathbf{x}_k, \mathbf{W}_k \rangle; \quad // \text{initial dictionary entry}$$

while (not converged)

$$\{NN^+\}_k = \{\}; \quad \{NN^-\}_k = \{\};$$

for (each bag  $\mathbf{B}_i \in \mathbf{D}$ )

// nearest neighbors search

$$\mathbf{p} = \arg \min_{\mathbf{B}_{ij} \in \mathbf{B}_i} d(\mathbf{E}_k, \mathbf{B}_{ij});$$

if ( $l_i == c$ ) then Add  $\mathbf{p}$  to  $\{NN^+\}_k$ ;

else Add  $\mathbf{p}$  to  $\{NN^-\}_k$ ;

$$\mathbf{x}_k = \text{Mean}(\{NN^+\}_k); \quad // \text{entry updates}$$

$$\mathbf{W}_k = \text{RCA}(\{NN^+\}_k, \{NN^-\}_k);$$

$$\mathbf{E}_k = \langle \mathbf{x}_k, \mathbf{W}_k \rangle;$$

Figure 1. pseudo-code for the IDC algorithm

**Decision list** A decision list is a variable-length sequence of decision nodes. Each node  $N$  is defined by an image feature dimension  $k_N$ , a classification threshold  $\theta_N$ , and a class label  $C_N$  for prediction. An image  $i$  is classified by node  $N$  into class  $C_N$  if

$$V_i(k_N) \leq \theta_N \quad (4)$$

where  $V_i(k_N)$  is the value of the image feature vector  $\mathbf{V}_i$  at dimension  $k_N$ . An example is classified by processing it against each node in the decision list until one of the nodes is able to classify the example.

**Training** Given all the training image features, the decision list is grown by starting with the empty list and adding decision nodes one at a time until all these training examples are correctly classified. The detailed steps of the algorithm are as follows:

1. Find the best decision node: The algorithm calls the function “*NodeFinder*” to search for the image feature dimension and corresponding threshold and class label that has the highest overall performance. The *NodeFinder* function is similar to the “*Weak\_Hypotheses\_Finder*” algorithm in [8], but it differs in several ways. The detailed introduction of the algorithm will be given in supplementary material.

2. Split the current training set based on the selected node: the correctly classified examples are removed from the training set; all the unclassified or misclassified examples are passed to the next node.

3. Repeat steps 1 & 2, adding new decision nodes to the list until the training set is empty, i.e., all the training examples have been correctly classified.

**Classification** Applying the learned decision list to a new image is similar to the training process. Its region descriptor vectors are first mapped to the image feature vector according to (3). The resulting feature vector is passed down the decision list until the image is classified by a decision node. Classification is efficient because most of the examples are classified by the nodes that appear early in the decision list.

**Bagged decision lists (BDL)** Note that although the decision lists have low bias, they can have high variance, which can lead to overfitting and poor performance. So we perform 200-fold bagging. This is accomplished by drawing 200 bootstrap replicates of the training images, and learning a separate decision list for each replicate training data set. A new image is classified by each of the 200 decision lists, which then vote to determine the overall prediction.

### 3. Experiments

Our method was tested on three families of object recognition problems: Caltech dataset [1,3,8], UIUC cars side dataset [3,8], and GRAZ dataset [8]. All the problems are binary *object present* versus *object absent* decision problems. On the test datasets, our method is compared with state-of-art generic object recognition approaches. Experiment settings are the same as in previous papers for fair comparison. The results are reported as the ROC-equal-error-rates (EERs). The results are summarized in Table 1 – Table 3. We can see that our method, IDC-BDL (Iterative Discriminative Clustering + Bagged Decision Lists), gives superior performance on most of the problems. On all problems where we obtain improvements, the differences are statistically significant at a 95% level using an unpaired test for the difference between two proportions [2]. Even on object classes where our method is not the best, its performance is comparable to other methods. In summary, the overall recognition performance of our approach matches or exceeds the state of the art.

In order to analyze the contribution of the major parts of our system, we also compare the whole IDC-BDL system to the ablated versions on the Caltech problems. The results show that both supervised dictionary learning and bagging strategy contribute to the high performance of the system. In addition, we also studied on the length of the decision lists learned by IDC-BDL. The decision lists are usually fairly short. This shows that only a small number of learned dictionary entries are sufficient for accurate classification.

**Table 1. EERs on Caltech dataset**

Dataset	IDC-BDL	[3]	[8]	[1]
Airplanes	<b>99.2</b>	93.7	88.9	98.0
Faces	98.4	91.7	93.5	<b>99.5</b>
Motorbikes	<b>98.3</b>	96.7	92.2	96.7
Leopards	<b>98.0</b>	89.0	/	/
Cars (Rear)	<b>95.5</b>	91.2	91.1	94.5

**Table 2. EERs on UIUC cars side dataset**

Dataset	Average length [confidence interval]	IDC-BDL	[3]	[8]
Cars (side)	27.3 [23.8, 31.5]	<b>92.7</b>	88.5	83.0

**Table 3. EERs on GRAZ dataset**

Dataset	Average length [confidence interval]	IDC-BDL	[8]
Bikes	14.1 [12.3, 16.2]	<b>76.5</b>	73.5
Persons	16.2 [14.1, 18.7]	<b>71.7</b>	63.0

### 4. Conclusion

This paper introduced an efficient new method to construct discriminative visual dictionaries based on bags of region descriptor vectors. The proposed system is robust to overfitting and low signal-to-noise ratio, which is shown by its competitive performance on several object recognition benchmark datasets.

### References

- [1] Y. Chen et al. MILES: Multiple-instance learning via embedded instance selection. *PAMI*, 2006.
- [2] T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 1998.
- [3] R. Fergus et al. Weakly supervised scale-invariant learning of models for visual recognition. *IJCV*, 2007.
- [4] W. T. Freeman & E. H. Adelson. The design and use of steerable filters. *PAMI*, 1991.
- [5] F. Jurie & B. Triggs. Creating efficient codebooks for visual recognition. *ICCV*, 2005.
- [6] T. Kadir, A. Zisserman, & M. Brady. An affine invariant salient region detector. *ECCV*, 2004.
- [7] K. Mikolajczyk & C. Schmid. Scale and affine invariant interest point detectors. *IJCV*, 2004.
- [8] A. Opelt et al. Generic object recognition with boosting. *PAMI*, 2006.
- [9] F. Perronnin et al. Adapted vocabularies for generic visual categorization. *ECCV*, 2006.
- [10] N. Shental et al. Adjustment learning and relevant component analysis. *ECCV*, 2002.
- [11] Q. Zhang & S. A. Goldman. EM-DD: An improved multiple-instance learning technique. *NIPS*, 2001.
- [12] W. Zhang et al. A hierarchical object recognition system based on multi-scale principal curvature regions. *ICPR*, 2006.