A Generative/Discriminative Learning Algorithm for Image Classification *

Yi Li†, Linda G. Shapiro†‡, and Jeff A. Bilmes‡
† Department of Computer Science and Engineering
‡ Department of Electrical Engineering
University of Washington
Seattle, WA 98195-2350
{shapiro@cs,yi@cs,bilmes@ee}.washington.edu

Abstract

We have developed a two-phase generative / discriminative learning procedure for the recognition of classes of objects and concepts in outdoor scenes. Our method uses both multiple types of object features and context within the image. The generative phase normalizes the description length of images, which can have an arbitrary number of extracted features of each type. In the discriminative phase, a classifier learns which images, as represented by this fixed-length description, contain the target object. We have tested the approach by comparing it to several other approaches in the literature and by experimenting with several different data sets and combinations of features. Our results, using color, texture, and structure features, show a significant improvement over previously published results in image retrieval. Using salient region features, we are competitive with recent results in object recognition.

1. Introduction

Recognition of classes of objects in images and videos is an important problem in computer vision with applications in autonomous vehicle navigation, surveillance, aerial video analysis, and image or video retrieval systems. In the context of image annotation, image regions from various segmentations are used for recognizing object classes in images or videos [5] [10] [16] [2] [15] [6] [7]. Appearance-based object recognition, which was initially proposed for recognizing specific objects, has progressed to detection of instances of object classes [12]. Most of these systems use formal learning methodologies, such as Bayesian decision making, neural nets, support vector machines (discriminative approach) or the EM algorithm (generative approach). More recently, the learning approach has been extended by the development of interest operators [9] [4] [11] that select image windows having patterns that might be used for recognizing objects and to the ability to learn constellations of parts that make up a more complex object [11] [3] [17] [14].

Our goal in this work is to develop a classification methodology for the automatic annotation of outdoor scene images. The training data is a set of images, each labeled with a list of one or more object (or concept) classes that it contains. There is no information on the locations of these entities in the image. For each class to be learned, a classifier is trained to detect instances of that class, regardless of size, orientation, or location in the image. The solution that we propose is a generative/discriminative learning procedure that learns the object or concept classes that appear in an image from multiple segmentations of pre-annotated training images. It is significant in several respects:

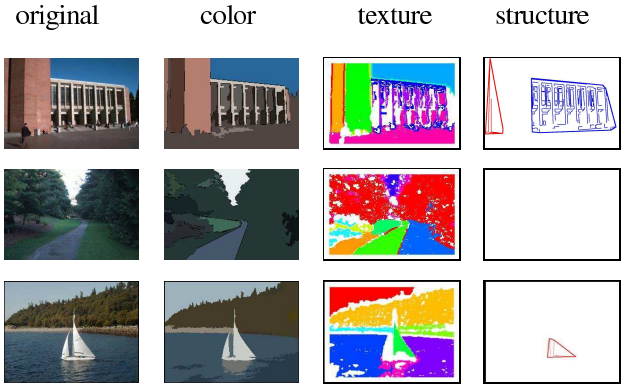

Figure 1. Abstract regions corresponding to color, texture, and structure segmentations.

*This research was supported by the National Science Foundation under Grants IRI-9711771, IIS-0097329, and IIS-0093430. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the National Science Foundation.

1. It is able to work with *any type of feature* that can be extracted from an image by some automatic segmentation process and represented by a vector of attribute values. It can work with regions from a color or texture segmentation, groups of line segments, or small windows selected by an interest operator.

2. It can work with any number of *different feature types* simultaneously. The formalism we developed for a single feature type generalizes easily to multiple feature types (Section 3.2). Thus we can use several features types together for a more powerful recognition system.

3. Like the work of Dorko and Schmid [1] and the more theoretical paper of Raina *et al* [13], our method consists of two phases: a generative phase followed by a discriminative phase. Our method is distinguished in the elegant framework we use for our discriminative phase. In particular, although each segmentation process can produce a variable number of instances of its features, our methodology produces a *fixed-length description of each image* that summarizes the feature information in a novel way. This allows the discriminative phase to be implemented by standard classifiers such as neural nets or support vector machines.

Although our work was motivated by the image annotation problem, the learning framework is general and could also be used as part of an object recognition system.

2. Abstract Regions

Our methodology allows the simultaneous use of multiple feature types for object recognition. In this paper, we will refer to the different feature types as *abstract regions*. Each type of abstract region a will have a type-a feature vector X^a containing the attribute values of that region type. Our learning methodology is general and can handle arbitrary region-based feature types. We have implemented three types of abstract regions for our studies: color, texture, and structure regions, and we have also been able to incorporate features from other systems. We will briefly describe our own features here and those of others in the discussions of our comparison experiments.

Our color regions are produced by a two-step procedure: 1) color clustering in the CIELab color space using a variant of the K-means algorithm and 2) an iterative merging procedure that merges multiple tiny regions into larger ones. The feature vector for a color region is $X^c = [L*, a*, b*]$, where $L*$ is the luminance, and $a*$ and $b*$ are the color channels. Our texture regions come from a color-guided texture segmentation process. Following the color segmentation, pairs of regions are merged if after a small dilation they overlap by more than 50%. Each of the merged regions is segmented using the same clustering algorithm on the Gabor texture coefficients. The feature vector for a texture region is $X^t = [g_1, g_2, \ldots, g_{12}]$ where the g_i's are the Gabor coefficients.

The features we use for recognizing man-made structures are called *structure features* and are obtained using the concept of a *consistent line cluster* [8]. Line segments are extracted from an image, and their color pairs (pairs of colors on opposite sides of the segment) are computed. The line segments are clustered first according to their color pairs, next according to their orientations, and finally according to their positions in the image to obtain the structure regions. The feature vector for a structure region is $X^s = [nl, L1, a1, b1, L2, a2, b2, \theta, no, mi]$ where nl is the number of lines in the region, $(L1, a1, b1)$ and $(L2, a2, b2)$ are its color pair, θ is its dominant orientation, no is the number of overlapping line segments, and mi is the maximum number of intersections of its line segments with those of another cluster.

Figure 1 illustrates the concept of abstract regions with color, texture, and structure features. The first image is of a large building. Regions such as the sky, the concrete, and the brick show up as large homogeneous regions in both color and texture segmentations. The windowed part of the building breaks up into many regions for both the color and texture segmentations, but it becomes a single structure region. The structure-finder also captures a small amount of structure at the left of the image. The second image of a park is segmented into several large regions in both color and texture. The green trees merge into the green grass on the right side in the color image, but the texture image separates them. No structure was found. In the last image of a sailboat, both the color and texture segmentations provide some useful regions that will help to identify the sky, water, trees and sailboat. The sailboat is also captured in the structure region. It is clear that no one feature type alone is sufficient to identify all objects. Therefore, a general purpose image classification system must have the ability to combine the power of mulitple features.

3. The Generative / Discriminative Learning Approach

We propose a new two-phase generative/discriminative learning approach that can learn to recognize objects using multiple feature types and variable numbers of features of each type in each image. Phase 1, the generative phase, is an unsupervised clustering step implemented with the classical EM algorithm. The clusters are represented by a multivariate Gaussian mixture model. Phase 1 also includes

an aggregation step that has the effect of normalizing the description length of images that can have an arbitrary number of regions. Phase 2, the discriminative phase, is a classification step that uses aggregated scores from the results of Phase 1 to compute the probability that an image contains a particular object class. It also generalizes to any number of different feature types in a seamless manner, making it both simple and powerful.

Our procedure for learning a specific object class o can be summarized as follows:

1. Generative Step
   (a) For each training image I_i and abstract region type a run the type-a segmentation procedure to produce a set $F_i^a = \{X_{i,r}^a | r = 1, \ldots, n_i^a\}$ of type-a feature vectors representing its regions.
   (b) Use the EM algorithm to produce an M^a-component Gaussian mixture model to approximate the feature vector distribution of the object-o training set $T^a = \cup_i \{F_i^a \mid \text{object } o \text{ appears in image } I_i\}$.
   (c) Use the Gaussian mixture models to derive a fixed-length aggregated feature vector V_i that summarizes the content of image I_i in terms of the components of the models for all feature types. (See Section 3.1 for the details.)
2. Discriminative Step
   (a) Label the aggregated feature vectors from the set of training images that contain an instance of object o with the label 1.
   (b) Label the aggregated feature vectors from the set of training images that do not contain any instances of object o with the label 0.
   (c) Train a classifier to distinguish between the classes 1 and 0. We used multi-layered perceptrons, but any standard classification algorithm could be used.

The details of our learning procedure are given below, first for the single-feature case and then for the extension to multiple types of features.

3.1. Single-Feature Case

In our framework, each object class is learned separately. Suppose that we are learning object class o and using feature type a. In Phase 1, the EM algorithm finds those clusters in the feature vector space for feature a that are most likely to appear in images containing object class o. Since the correspondence between regions and objects is unknown, all of the type a feature vectors from all the training images containing object o are used. The EM algorithm approximates the feature vector distribution by a Gaussian mixture model. Thus the probability of a particular type-a feature vector X^a appearing in an image containing object o is

$$P(X^a|o) = \sum_{m=1}^{M^a} w_m^a \cdot N(X^a; \mu_m^a, \Sigma_m^a)$$

where $N(X, \mu, \Sigma)$ is a multivariate Gaussian distribution over feature vector set X with mean μ and covariance matrix Σ, M^a is the total number of Gaussian components, and w_m^a is the weight of Gaussian component m^a. Each Gaussian component represents a cluster in the feature vector space for feature type a that is likely to be found in the images containing object class o. Figure 2a shows two positive and two negative training images for the *beach* class and the means of eight Gaussian components for the color feature learned from positive training images. Note that the mixture for object class o is trained with all regions of all images that contain o, but these images also contain many other regions from other object classes. Our discriminative step (described below) learns how to exploit this information to predict the presence of the target object.

a. Sample Training Images and Component Means

Means of 8 Color Components from EM Clustering

b. Aggregated Scores

*beach*1	0.93	0.16	0.94	0.24	0.10	0.99	0.32	0.00
*beach*2	0.66	0.80	0.00	0.72	0.19	0.01	0.22	0.02
*nonbch*1	0.43	0.03	0.00	0.00	0.00	0.00	0.15	0.00
*nonbch*2	0.15	0.77	0.18	0.02	0.28	0.49	0.12	0.47

Figure 2. a. Two positive and two negative training images for the "beach" class and the mean values for the color clusters produced by the EM clustering algorithm on the full set of "beach" training images. b. Feature vectors with aggregated scores for the two positive and two negative examples using the max aggregate function and 8 components.

Once the Gaussian components are computed, the likelihood that those components are present in each training image can be calculated. For image I_i and its type-a region r, let $X^a_{i,r}$ be the corresponding feature vector. Image I_i will produce a number of type-a region feature vectors, $X^a_{i,1}$, $X^a_{i,2}, \ldots, X^a_{i,n^a_i}$. The number n^a_i of type-a feature vectors is the same as that of the type-a regions obtained from the type-a image segmentation and varies from image to image. The joint probability of the type-a features of region r and cluster m^a is given by

$$P(X^a_{i,r}, m^a) = w^a_m \cdot N(X^a_{i,r}, \mu^a_m, \Sigma^a_m)$$

From these probabilities, we compute a summary score indicating the degree to which a component m^a explains the image I_i as:

$$P(I_i, m^a) = f(\{P(X^a_{i,r}, m^a) | r = 1, 2, \ldots, n^a_i\})$$

where f is an aggregate function that combines the evidence from each of the type-a regions in the image. We have tried max and $mean$ as aggregate functions in our experiments. Figure 2b shows the feature vectors with the aggregated scores for the positive and negative training images of Figure 2a using max as the aggregate function.

Let $I^+_1, I^+_2, \ldots,$ be positive training images (images that contain object o) and $I^-_1, I^-_2, \ldots,$ be negative training images. Our Phase 2 algorithm starts by assembling the computed values of $P(I_i, m^a)$ for each image I_i and each type-a component m^a into the following training matrix:

$$\begin{array}{c} I^+_1 \\ I^+_2 \\ \vdots \\ I^-_1 \\ I^-_2 \\ \vdots \end{array} \begin{bmatrix} P(I^+_1, 1^a) & P(I^+_1, 2^a) & \cdots & P(I^+_1, M^a) \\ P(I^+_2, 1^a) & P(I^+_2, 2^a) & \cdots & P(I^+_2, M^a) \\ \vdots & & & \\ P(I^-_1, 1^a) & P(I^-_1, 2^a) & \cdots & P(I^-_1, M^a) \\ P(I^-_2, 1^a) & P(I^-_2, 2^a) & \cdots & P(I^-_2, M^a) \\ \vdots & & & \end{bmatrix}$$

This matrix is used to train a second-stage classifier, which can implement any standard learning algorithm (support vector machines, neural networks, etc.) The classifier will learn how these aggregated scores correspond to the presence or absence of the object class. For notational purposes, let $Y^{m^a}_{I_i} = P(I_i, m^a)$ and $Y^{1^a:M^a}_{I_i} = [Y^{1^a}_{I_i}, Y^{2^a}_{I_i}, \cdots, Y^{M^a}_{I_i}]$, which is just one row of the matrix. The second-stage classifier will learn $P(o|I_i) = g(Y^{1^a:M^a}_{I_i})$ for object class o, image I_i. We use 3-layer feedforward multi-layered perceptrons (referred to as MLP). The activation function used on the hidden and output nodes is a sigmoid function. In the test stage, given a new image I_j and its feature vectors for all type-a regions, the aggregated vector $Y^{1^a:M^a}_{I_j}$ is calculated and the second-stage classifier calculates the likelihood that image I_j contains target object o based on feature type a using the learned function.

3.2. Multiple-Feature Case

To use multiple features, the generative step is run separately for each feature type, producing a separate Gaussian mixture model for each. We will denote the color feature vectors by $Y^{1^c:M^c}_{I_i}$, the texture feature vectors by $Y^{1^t:M^t}_{I_i}$, and the structure feature vectors by $Y^{1^s:M^s}_{I_i}$. To fuse these different information sources, we simply concatenate $Y^{1^c:M^c}_{I_i}$, $Y^{1^t:M^t}_{I_i}$, and $Y^{1^s:M^s}_{I_i}$ to obtain a new combined feature vector $V_i = [Y^{1^c:M^c}_{I_i}\ Y^{1^t:M^t}_{I_i}\ Y^{1^s:M^s}_{I_i}]$ for image I_i.

$$\begin{array}{c} I^+_1 \\ I^+_2 \\ \vdots \\ I^-_1 \\ I^-_2 \\ \vdots \end{array} \begin{bmatrix} \cdots & Y^{m^c}_{I^+_1} & \cdots & Y^{m^t}_{I^+_1} & \cdots & Y^{m^s}_{I^+_1} & \cdots \\ \cdots & Y^{m^c}_{I^+_2} & \cdots & Y^{m^t}_{I^+_2} & \cdots & Y^{m^s}_{I^+_2} & \cdots \\ & & & \vdots & & & \\ \cdots & Y^{m^c}_{I^-_1} & \cdots & Y^{m^t}_{I^-_1} & \cdots & Y^{m^s}_{I^-_1} & \cdots \\ \cdots & Y^{m^c}_{I^-_1} & \cdots & Y^{m^t}_{I^-_1} & \cdots & Y^{m^s}_{I^-_1} & \cdots \\ & & & \vdots & & & \end{bmatrix} =$$

$$\begin{array}{ccc} color & texture & structure \\ \begin{bmatrix} \cdots Y^{m^c}_{I^+_1} \cdots \\ \cdots Y^{m^c}_{I^+_2} \cdots \\ \vdots \\ \cdots Y^{m^c}_{I^-_1} \cdots \\ \cdots Y^{m^c}_{I^-_1} \cdots \\ \vdots \end{bmatrix} & \begin{bmatrix} \cdots Y^{m^t}_{I^+_1} \cdots \\ \cdots Y^{m^t}_{I^+_2} \cdots \\ \vdots \\ \cdots Y^{m^t}_{I^-_1} \cdots \\ \cdots Y^{m^t}_{I^-_1} \cdots \\ \vdots \end{bmatrix} & \begin{bmatrix} \cdots Y^{m^s}_{I^+_1} \cdots \\ \cdots Y^{m^s}_{I^+_2} \cdots \\ \vdots \\ \cdots Y^{m^s}_{I^-_1} \cdots \\ \cdots Y^{m^s}_{I^-_1} \cdots \\ \vdots \end{bmatrix} \end{array}$$

A classifier is then trained on these combined feature vectors to predict the existence of the target object using the same method just described for the single-feature case. The classifier will learn a weighted combination of components from different feature types that are important for recognizing the target objects and find the best weights to combine different feature types automatically.

The two-phase generative/discriminative approach has several potential advantages. It can combine any number of different feature types without any modeling assumptions. Regions from different segmentations do not have to align or to correspond in any way. Segmentations that produce a sparse set of features can be handled in the same manner as those whose features cover the entire image. Our method can learn object classes whose members have several different appearances, such as trees or grass. It can also learn high-level concepts or complex objects composed of several simpler objects, such as a football stadium, which has green turf, a pattern of white lines, and a red track around it, or a beach with sand, dark blue water, and sky. Finally,

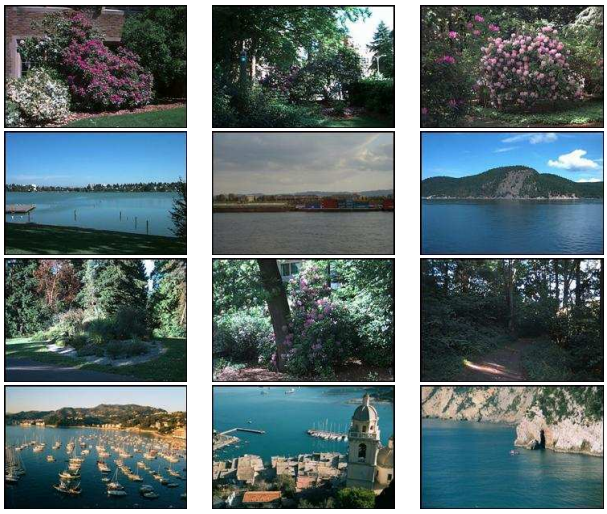

Figure 3. Highest-scoring image retrieval results for several categories of the groundtruth data set. Queries are key words. *Row 1:* **spring flowers;** *Row 2:* **water;** *Row 3:* **parks;** *Row 4:* **Italy.**

since it learns only one object at a time and does not require training images to be fully labeled, new training images with a new object label can be added to an already existent training database. A model for this new object class can be constructed, while the previously-learned models for other object classes are kept intact.

4. Experiments

Our approach was developed for image annotation in the image and video retrieval application. For this domain we ran several sets of experiments in order to 1) test our two-phase learning approach on several different image databases, 2) try several different combinations of features, and 3) compare it to previous approaches in the literature. We tested our two-phase approach on three local data sets: a groundtruth database of 1,224 outdoor scene images with multiple object and concept classes , another local database of 1,951 images of buses, small buildings, and skyscrapers, and a third database of 828 frames from aerial videos. The groundtruth database and the video frame database were hand-labeled with multiple labels per image, while the bus, building, and skyscraper images were assigned to just one category. For the image annotation task, we compared our two-phase approach to the ALIP approach of Li and Wang [6] and to the machine translation approach of Duygulu *et al.* [2] using their databases. For the object recognition domain, we compared our approach to the work of Fergus *et al* [3] and that of Dorko and Schmidt [1] using the database of airplane, face, and motorbike images from their work.

tree (97.3)	**Italy** (99.9)	sky (95.1)
bush (91.6)	grass (98.5)	**Iran** (89.3)
spr. flowers (90.3)	**sky** (93.8)	house (88.6)
flower (84.4)	rock (88.8)	**building** (80.1)
park (84.3)	**boat** (80.1)	boat (71.7)
sidewalk (67.5)	**water** (77.1)	bridge (67.0)
grass (52.5)	**European** (56.3)	**water** (13.5)
pole (34.1)	**house** (5.3)	**tree** (7.7)

Figure 4. Classifier-produced likelihood scores from the groundtruth data set. For each image, the boldface labels under it are human annotations, and the nonbold labels are other high-scoring categories.

4.1. Performance on Groundtruth Data Set

We are interested in images in which the target object can be anywhere in the image and is not necessarily the main theme of the image. For example, we want to recognize the category *tree* in images containing trees whose main theme is *house*, *beach*, or *flower*. For this purpose we have constructed a groundtruth image set containing 1,224 images and growing. The set includes our own images and those contributed by other researchers around the world. The whole image set is free for research purpose and is fully labelled. The URL is http://www.cs.washington.edu/research/imagedatabase/groundtruth/

In the groundtruth image set there are 31 elementary object categories and 20 high-level concepts represented in this database. Our qualitative experiments were image retrievals according to classifier-produced likelihood values for each of the 51 classes. Figure 3 shows some of the images that received the highest likelihood scores for each of four categories: *spring flowers*, *water*, *parks*, and *Italy*. Figure 4 shows three representative images from the groundtruth set and their likelihood scores. In our quantitative experiments, the recognition threshold for the output of the MLP classifier was varied to obtain a set of ROC curves to display the percentage of true positives vs. false positives for each object class. The measure of performance

bus (100.0)	**building** (100.0)	**skyscraper** (99.9)
building (58.1)	bus (2.79)	building (6.8)
skyscraper (1.1)	skyscraper (0.04)	bus (0.0)

Figure 5. Classifier-produced likelihood scores from the structure image set. Boldface labels are human-identified categories.

for each class was the percentage of the whole area under its ROC curve, which ranges from 0 to 100 and which we will henceforth call a *ROC score*. Table 1 shows the ROC scores in ascending order for these categories obtained using color, texture, and structure features. In general, the lower scores are obtained for object classes that have both high variance in appearance and insufficient samples in the database to learn those variations. We have no feature expressly designed for recognizing people, so they are recognized mostly by context and the performance is low.

4.2. Performance of the Structure Feature

To more thoroughly investigate the performance of the structure feature, we created a database of 1,951 images from freefoto.com including 1,013 images of buses, 609 images of buildings, and 329 images of skyscrapers. For these experiments we used the 10 attributes for the structure feature given in Section 2. We tested the structure feature alone and combined with the color segmentation feature. Figure 5 shows some images from the structure set and their likelihood scores for the three possible labels. Table 2 shows the ROC scores for the three categories. While the structure feature did a pretty good job of identifying the categories, the addition of the regions from a color segmentation of the whole image improved the identification of the building category.

4.3. Performance on Aerial Video Frames

We also applied our learning framework to recognize objects in aerial video frames. While tracking can detect objects in motion, our object recognition system can provide information about the static objects, such as forest, road, and field, which are also important in video analysis. The aerial image set contains 828 video frames. We chose a set of 10 objects that appeared in at least 30 images for our experiments. Several combinations of color, texture and struc-

Object Class	ROC Score	Object Class	ROC Score
street	60.4	stone	87.1
people	68.0	hill	87.4
rock	73.5	mountain	88.3
sky	74.1	beach	89.0
ground	74.3	snow	92.0
river	74.7	lake	92.8
grass	74.9	frozen lake	92.8
building	75.4	japan	92.9
cloud	75.4	campus	92.9
boat	76.8	barcelona	92.9
lantern	78.1	geneva	93.3
australia	79.7	park	94.0
house	80.1	spring flowers	94.4
tree	80.8	columbia gorge	94.5
bush	81.0	green lake	94.9
flower	81.1	italy	95.1
iran	82.2	swiss mountains	95.7
bridge	82.7	sanjuans	96.5
car	82.9	cherry tree	96.9
pole	83.3	indoor	97.0
yellowstone	83.7	greenland	98.7
water	83.9	cannon beach	99.2
indonesia	84.3	track	99.6
sidewalk	85.7	football field	99.8
asian city	86.7	stadium	100.0
european city	87.0		

Table 1. Groundtruth Experiments

	bus	building	skyscraper
structure	90	79	89
structure + color	92	85	93

Table 2. Structure Experiments (ROC scores)

ROC Score (%)	airplane	car	dirt road	field	forest	building	paved road	people	runway	tree
cs	81.2	81.6	86.8	77.2	83.3	82.4	79.9	**83.9**	92.9	77.5
st	83.5	68.8	70.1	68.2	71.3	78.2	66.9	49.7	80.3	61.0
cs+st	90.1	78.9	86.4	**77.5**	86.4	83.7	81.5	83.9	93.9	77.5
cs+ts	78.4	81.1	**89.5**	74.2	86.7	80.8	79.8	83.8	**94.4**	**80.6**
cs+ts+st	**91.1**	**82.3**	88.1	74.1	**87.6**	**84.9**	**87.5**	79.7	93.6	77.1

Table 3. Learning performance on aerial video image set. "cs" stands for "color segmentation", "ts" stands for "texture segmentation", and "st" stands for "structure".

runway (99.9)	**car** (94.3)	car (97.9)
field (98.7)	**dirt road** (91.7)	**forest** (94.2)
car (96.2)	**field** (16.17)	**paved road** (85.0)
		dirt road (72.4)
		tree (68.8)

Figure 6. Classifier-produced likelihood scores from the aerial video image set. Boldface labels are human annotations; nonbold labels are other high-scoring categories.

ture features were tested within our learning framework. Sample results are shown in Figure 6. The ROC scores are given in Table 3.

4.4. Comparison to the ALIP Algorithm

We measured the performance of our system on the benchmark image set used by SIMPLIcity [16] and ALIP [6]. We chose ALIP (which outperformed SIMPLIcity) for our comparison, because it uses local features, employs a learning framework, and provides a set of 1000 labeled images for training and testing. The image set contains 10 categories (100 images each) from the COREL image database.

In ALIP, image feature vectors are extracted from multiple resolution wavelets, and objects are represented by 2D multiple-resolution hidden Markov models. We applied different combinations of color, texture, and structure features in our framework; the number of correctly categorized images are shown in Table 4. The performance of our system is similar to ALIP using only color feature, significantly exceeds ALIP's performance with color and structure features combined, and achieves even better performance with the combination of color, texture, and structure. This experiment shows the power of our learning framework and also the benefit of combining several different image features.

4.5. Comparison to Machine Translation

We also compared our two-phase learning approach to that of Duygulu *et al.* [2]. In this work, image regions were treated as one language and the object labels as another, so the task of annotating images can be viewed as machine translation. Using their region-based, 33-attribute feature

Categorization Accuracy (%)	MEAN	African	beach	buildings	buses	dinosaurs	elephants	flowers	food	horses	mountains
ALIP	**63.6**	52	32	64	46	100	40	90	68	60	84
color	**64.2**	69	44	43	60	88	53	85	63	94	43
color+struct	**75.4**	79	48	70	85	89	64	87	77	92	63
color+text.+struct	**80.3**	74	64	78	95	93	69	91	85	89	65

Table 4. Comparison to ALIP

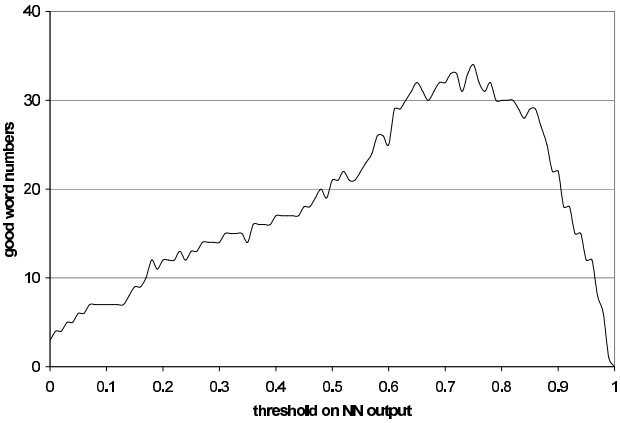

Figure 7. The number of good words vs. the threshold. Three of the words appeared in more than 15% of the total images, so that even when the threshold was set to 0, there were still 3 good words.

vectors, we extracted 3 color attributes to form a color feature vector and 12 texture attributes to form a texture feature vector and combined them in our Phase 2 learning step. From the 5000 Corel images provided, 4500 images were used for training and 500 for testing.

In [2] the evaluations were based on recall-precision pairs from varying a minimum-probability threshold that controls whether a region predicts a word or not. Performance was characterized by the number of "good words" with recall greater than 0.4 and precision greater than 0.15; a high of 14 of the 371 keywords was achieved. We selected 81 keywords, each having at least 50 corresponding images. In our experiments, we varied from 0 to 1 the threshold that determines from our MLP output whether an image is positive or negative. Our results are shown in Figure 7. The number of good words from our approach was much higher than that from [2], which is a further endorsement of our generative/discriminative learning algorithm.

	Fergus	Dorko/Schmid	Ours
airplanes	90.2%	96.0%	96.6%
faces	96.4%	96.8%	96.5%
motorbikes	92.5%	98.0%	99.2%

Table 5. Comparisons to results of [3] and [1]

4.6. Comparison to Salient Features Work

In order to test the validity of our approach in the object recognition domain, we applied it to the airplane, motorbike, and face data sets of Fergus *et al.* [3] using the same entropy-based salient regions [4]. The data set we used contains 1074 airplane images, 826 motorbike images, 450 face images, and 900 background images. For each object category, half of the positive images were used for training and half for testing as in [3]. Fergus' approach used the EM algorithm to find constellations of parts and required no negative training images. Our discriminative stage requires negative images, so we added half of the background images to the training set and left the other half for testing. About 100-300 salient regions were detected in each image, and SIFT features [9] were used to represent each by a length-128 feature vector. This representation is from the recent work of Dorko and Schmid [1] and differs from that of [3]. A comparison of our experimental results to those of [3] and [1] are shown in Table 5. Since our algorithm was designed to handle general object classes in outdoor images, our approach does not explicitely learn spatial configurations that might be helpful for recognizing objects made of distinctive parts. However, the results show that our approach can achieve better performance than [3] and similar performance to [1] without using the explicit spatial information.

5. Conclusions and Future Work

We have described a new two-phase generative/discriminative learning algorithm for object and concept recognition. The generative phase normalizes the description length of images, which in general will have an arbitrary number of abstract region features. The discriminative step learns which images, as represented by this fixed-length description, contain the target object. We have experimented with several different combinations of features on several different image data sets. We have compared our new method to the ALIP approach [6] and to the machine translation approach [2] with favorable results. We have also shown that our system's performance exceeds that of Fergus [3] and is similar to that of Dorko and Schmid [1] when we use the salient-region features. In future work we have more experiments planned to compare different variants of our approach. We are also working on a probabilistic mechanism for identifying the regions within an image where the target object is likely to lie.

References

[1] G. Dorko and C. Schmid. Object class recognition using discriminative local features. Technical Report 5497, INRIA Rhone-Alpes, February 2005.

[2] P. Duygulu, K. Barnard, J. F. G. de Freitas, and D. A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fi xed image vocabulary. In *ECCV02*, volume 4, pages 97–112, 2002.

[3] R. Fergus, P. Perona, and A. Zisserman. Object-class recognition by unsupervised scale-invariant learning. *CVPR*, 2:264–271, 2003.

[4] T. Kadir and M. Brady. Saliency, scale and image description. *IJCV*, 45(2):83–105, 2001.

[5] S. Kumar, A. C. Loui, and M. Hebert. An observation-constrained generative approach for probabilistic classifi cation of image regions. *Image and Vision Computing*, 21:87–89, 2003.

[6] J. Li and J. Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *PAMI*, 25(9):1075–1088, September 2003.

[7] Y. Li, J. Bilmes, and L. G. Shaprio. Object class recognition using images of abstract regions. *ICPR*, pages 40–43, 2004.

[8] Y. Li and L. G. Shapiro. Consistent line clusters for building recognition in CBIR. In *Proceedings of the International Conference on Pattern Recognition*, pages 952–956, 2002.

[9] D. G. Lowe. Object recognition from local scale-invariant features. *ICCV*, pages 1150–1157, 1999.

[10] O. Maron and A. L. Ratan. Multiple-instance learning for natural scene classifi cation. In *ICML*, 1998.

[11] K. Mikolajczyk and C. Schmid. An affi ne invariant interest point detector. *ECCV*, 2002.

[12] A. Mohan, C. Papageorgious, and T. Poggio. Example-based object detection in images by components. *PAMI*, 23(4):349–361, 2001.

[13] R. Raina, Y. Shen, A. Y. Ng, and A. McCallum. Classifi - cation with hybrid generative/discriminative models. *NIPS*, 16, 2002.

[14] J. Sivic and A. Zisserman. Video google: A test retrieval approach to object matching in videos. *ICCV*, 2003.

[15] N. Vasconcelo. Exploiting group structure to improve retrieval accuracy and speed in image databases. In *International Conference on Image Processing*, 2002.

[16] J. Z. Wang, J. Li, and G. Wiederhold. SIMPLIcity: Semantics-sensitive integrated matching for picture LIbraries. *IEEE PAMI*, 23(9):947–963, 2001.

[17] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. *ECCV*, pages 18–32, 2000.