Multi-label machine learning and its application to semantic scene classification

Xipeng Shen[a], Matthew Boutell[a], Jiebo Luo[b], Christopher Brown[a]

[a]Department of Computer Science, University of Rochester, Rochester, NY, USA 14627
[b]Electronic Imaging Products, R & D, Eastman Kodak Company, Rochester, NY, USA 14650

ABSTRACT

In classic pattern recognition problems, classes are mutually exclusive by definition. Classification errors occur when the classes overlap in the feature space. We examine a different situation, occurring when the classes are, by definition, *not* mutually exclusive. Such problems arise in scene and document classification and in medical diagnosis. We present a framework to handle such problems and apply it to the problem of semantic scene classification, where a natural scene may contain multiple objects such that the scene can be described by multiple class labels (e.g., a field scene with a mountain in the background). Such a problem poses challenges to the classic pattern recognition paradigm and demands a different treatment. We discuss approaches for training and testing in this scenario and introduce new metrics for evaluating individual examples, class recall and precision, and overall accuracy. Experiments show that our methods are suitable for scene classification; furthermore, our work appears to generalize to other classification problems of the same nature.

Keywords: pattern recognition, machine learning, image understanding, semantic scene classification, multi-label classification, multi-label training, multi-label evaluation, image organization, cross-training, Jaccard similarity

1. INTRODUCTION

In traditional classification tasks[1]:

> Classes are **mutually exclusive by definition**. Let χ be the domain of examples to be classified, Y be the set of labels, and H be the set of classifiers for $\chi \rightarrow Y$. The goal is to find the classifier $h \in H$ maximizing the probability of $h(x) = y$, where $y \in Y$ is the ground truth label of x, i.e.,
>
> $$y = \arg\max_i P(y_i|x)$$

Classification errors occur when the classes overlap in the selected feature space (Fig. 1a). Various classification methods have been developed to provide different operating characteristics, including linear discriminant functions, artificial neural networks (ANN), and support vector machines (SVM).[1]

However, in some classification tasks, it is likely that some data belongs to multiple classes, causing the actual classes to overlap *by definition.* In text or music categorization, documents may belong to multiple genres, such as *government* and *health*, or *rock* and *blues*.[2, 3] In medical diagnosis, a disease may belong to multiple categories, and genes may have multiple functions, yielding multiple labels.[4]

A problem domain receiving renewed attention is semantic scene classification,[5–8] categorizing images into semantic classes such as *beaches*, *sunsets* or *parties*, in which many images may belong to multiple semantic classes. Figure 4(a) shows an image that had been classified by a human as a beach scene. However, it is clearly both a beach scene *and* an urban scene. It is not a *fuzzy* member of each (due to ambiguity), but is *fully* a member of each class (due to multiplicity). Figure 4(b) (field and fall foliage scene) is similar.

Further author information: (Send correspondence to M.B.)
X.S., M.B., C.B.: E-mail: {xshen, boutell, brown}@cs.rochester.edu
J.L.: E-mail: luo@image.kodak.com

Much research has been done on scene classification recently[5–8] . Most systems are exemplar-based, learning patterns from a training set using statistical pattern recognition techniques. However, none addresses the use of multi-label images.

When choosing their data sets, most researchers either avoid such images, label them subjectively with the base (single-label) class most obvious to them, or consider "*beach+urban*" as a new class. The last method is unrealistic in most cases because it would increase the number of classes to be considered substantially and the data in such combined classes is usually sparse. The first two methods have limitations as well. For example, in content-based image indexing and organization applications, it would be more difficult for a user to retrieve a multiple-class image (e.g., "*beach+urban*") if we only have exclusive beach or urban labels. It may require two separate queries to be conducted respectively and the intersection of the retrieved images be taken. In a content-sensitive enhancement application, it may be desirable for the system to have different settings for beach, urban, and beach+urban scenes. This is impossible using exclusive single labels. In this work, we consider the following problem:

> The base classes are non-mutually-exclusive and may **overlap by definition** (Fig. 1b). As before, let χ be the domain of examples to be classified and Y be the set of labels. Now let B be a set of binary vectors, each of length $|Y|$. Each vector $b \in B$ indicates membership in the base classes in Y (+1 = member, -1 = non-member). H is the set of classifiers for $\chi \rightarrow B$. The goal is to find the classifier $h \in H$ that minimizes a distance (e.g., Hamming), between $h(x)$ and b_x for a newly observed example x.
>
> In a probabilistic formulation, the goal of classifying x is to find **one or more** base class labels in a set C and for a threshold T such that
>
> $$P(c|x) > T, \forall c \in C$$

Clearly, the mathematical formulation and its physical meaning are distinctively different from those used in classic pattern recognition. Few papers address this problem (see Sect. 2), and most of these concern text classification. Based on the multi-label model, we investigate several methods of training and propose a novel training method, "cross-training". We also propose three classification criteria in testing. When applying our methods to scene classification, our experiments show that our approach is successful on multi-label images even without an abundance of training data. We also propose a generic evaluation metric, which can be tailored to applications needing different error forgiveness.

2. RELATED WORK

The sparse literature on multi-label classification is primarily geared to text classification. Schapire and Singer[2] proposed BoosTexter, extending AdaBoost to handle multi-label text categorization. However, they note that controlling complexity due to over-fitting in their model is an open issue. McCallum[3] proposed a mixture model trained by EM, selecting the most probable set of labels from the power set of possible classes and using heuristics to overcome the associated computational complexity. However, his generative model is based on learning text frequencies in documents, and is thus specific to text applications. Joachims' approach is most similar to ours in that he uses a set of binary SVM classifiers as well.[9] He also finds that SVM classifiers achieve higher accuracy than others. However, he does not discuss multi-label training models or specific testing criteria.

A related approach to image classification consists of segmenting and classifying image *regions* (e.g., sky, grass).[10, 11] A seemingly natural approach to multi-label scene classification is to model such scenes using combinations of these labels. For example, if a mountain scene is defined as one containing rocks and sky and a field scene as one containing grass and sky, then an image with grass, rocks, and sky would be considered both a field scene and a mountain scene.

However, this approach has drawbacks. First, region labeling has only been applied with success to constrained environments with a limited number of predictable objects (e.g., outdoor images captured from a moving vehicle[10]). Second, because scenes consist of groups of regions, there is a combinatorial explosion in the number of region combinations. Third, scene modeling is a difficult problem in its own right, encompassing more than mere presence or absence of objects. For example, a scene with sky, water and sand could be best described as a lake or a beach scene, depending on the relative size and placement of the components.

The difficulties with the segmentation-based approach have driven many researchers to use a low-level feature, exemplar-based approach (e.g., Refs. 5, 6, 8). While many have taken this approach, none handle the multi-label problem.

The main contribution of this work is an extensive comparative study of possible approaches to training and testing multi-label exemplar-based classifiers. The novelty of our work will be summarized in the conclusion.

3. MULTI-LABEL CLASSIFICATION

In this section, we describe possible approaches for training and testing with multi-label data. Consider two classes, denoted by '+' and 'x' respectively. Examples belonging to both the '+'and 'x' classes simultaneously are denoted by '*' (see Fig. 1b).

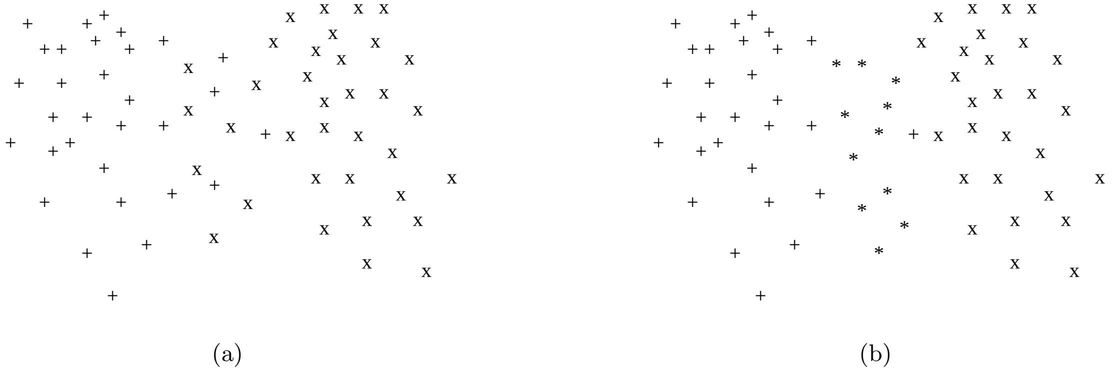


Figure 1. Figure (a) is the typical pattern recognition problem. Two classes contain examples that are difficult to separate in the feature space. Figure (b) is the multi-label problem. The * data belongs to both of the other two classes simultaneously.

3.1. Multi-label Training Models

For multi-label classification, the first question to address is that of training. Specifically, how should training examples with multiple labels be used in the training phase?

In previous work, researchers labeled the multi-label data with the one class to which the data most likely belonged, by some perhaps subjective criterion. For example, the image of hotels along a beach (Fig. 4a) would be labeled as a beach if the beach covered the majority of the image, or if one happened to be looking for a beach scene at the time of data collection. In our example, part of the '*' data would be labeled as '+', and part would be labeled as 'x' (e.g., depending on which class was most dominant). We call this kind of model *MODEL-s* (s stands for "**s**ingle-label" class).

Another possible method would be simply to ignore the multi-label data when training the classifier. In our example, all of the '*' data would be discarded. We call the model trained by this approach *MODEL-i* (i stands for "**i**gnore").

A straightforward method to achieve our goal of correctly classifying the data in each class is to consider those items with multiple labels as a new class (e.g., the '*' class) and build a model for it. We call the model trained by this method *MODEL-n* (n stands for "**n**ew" class). However, one important problem with this approach is that the data belonging to multiple classes is usually too sparse to build usable models. Table 1 shows the number of various images in our training data. (Details pertaining to this data set, such as the image source and how the set was randomly split into independent training and testing sets, are given in Ref. 12.) While the number of images belonging to more than one class comprises over 7% of the database, many combined classes (e.g., *beach+field*) are extremely small. This is an even greater problem when some scenes can be assigned to more than two classes.

Table 1. Experimental Data: BH–Beach, ST–Sunset, FE–Foliage, FD–Field, MN–Mountain, UN–Urban

Class	Training Images	Testing Images	Total
BH	194	175	369
ST	165	199	364
FE	184	176	360
FD	161	166	327
BH+FD	0	1	1
FE+FD	7	16	23
MN	223	182	405
BH+MN	21	17	38
FE+MN	5	8	13
FD+MN	26	49	75
FD+FE+MN	1	0	1
UN	210	195	405
BH+UN	12	7	19
FD+UN	1	5	6
MN+UN	1	0	1
Total	1211	1196	2407

A novel method is to use the multi-label data more than once when training, using each example as a positive example of *each* of the classes to which it belongs. In our example, we consider the ‘*’ data to belong to the ‘+’ class when training the ‘+’ model, and consider it to belong to the ‘x’ class when training the ‘x’ model. We emphasize that the ‘*’ data is not used as a negative example of either the ‘+’ or the ‘x’ classes. We call this approach “*cross-training*”. The resulting class decision surfaces are illustrated in Fig. 2. The area A belongs to both the ‘+’ and ‘x’ classes. When classifying an example in area A, the models of ‘+’ and ‘x’ are each expected to classify it as an instance of each class. According to the testing label criterion, that image will have multiple labels, ‘+’ and ‘x’. This method avoids the problem of sparse data since we use all related data that can be used for each model. Compared with the training approach of *MODEL-n*, cross-training can use training data more effectively since the cross-training models contain more training data than *MODEL-n*. We call the model obtained using this approach as *MODEL-x* (x stands for “**cross**-training”).

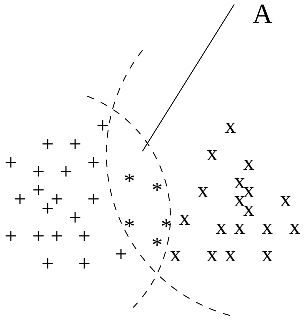


Figure 2. Illustration of cross-training

One might argue that this approach gives too much weight to examples with multiple labels. It may be so if a density estimation based classifier (e.g., ANN) is used. We recognized that it seems natural to use a neural network with one output node per class to deal with multi-label classification. However, we used SVMs in our study as they have been empirically proved to yield higher accuracy and better generalizability in scene[13, 14] and text[9] classification. Intuitively, multi-label images are likely to be those that are near the decision boundaries, making them particularly valuable for SVM-type classifiers. In practice, the sparseness of multi-label images also makes it imperative to use all such images. If there are predominant percentages of multiple images, it is possible and may be necessary to use multi-label examples by sampling according to a distribution over the labels.

3.2. Multi-label Testing Criteria

In this section, we discuss options for labeling criteria to be used in testing. As stated above, the sparseness of some class combinations prohibits us, in general, from building models of each combination (*MODEL-n*). Therefore, we only build models for the base classes. We now discuss how to obtain multiple labels from the output of the base class models.

To simplify our discussion, we use the SVM as an example classifier. In the one-vs-all approach,[15] one classifier is trained for each class and each outputs a score for a test example. Each of the N scores corresponds loosely to the likelihood of the example belonging to the N base classes (e.g., mapping via a logistic function produces posterior probabilities[16]). Whereas for standard two-class SVMs, the example is labeled as a positive instance if the SVM score is *positive*, in the one-vs-all approach, the example is labeled with the single class corresponding to the SVM that outputs the *maximum* score, even if multiple scores are positive. It is also possible that for some examples, none of the N SVM scores is positive due to the imperfectness of features.

To generalize the one-vs-all-approach to multi-label classification, we experiment with the following three labeling criteria.

- **P-Criterion:** Label input testing data by all of the classes corresponding to *positive* SVM scores. (In "P-Criterion", P stands for **p**ositive.) If no scores are positive, label that data example as "unknown".
- **T-Criterion:** This is similar to the P-Criterion, but differing in how to deal with the all-negative-score case. Here, we use the Closed World Assumption (CWA) that all examples belong to at least one of the N classes. If all the N SVM scores are negative, the input is given the label corresponding to the SVM producing the *top* (least negative) score. (T denotes **t**op.)
- **C-Criterion:** The decision depends on the *closeness* between the top SVM scores, regardless of whether they are positive or negative. (C denotes **c**lose.) Among all the SVM scores for an example, if the top M are close enough, then the corresponding classes are considered as the labels for that example. We use the *maximum a posteriori* (MAP) principle to determine the threshold for judging if the SVM scores are close enough or not. (Note that this is independent of the probabilistic interpretation of SVM scores given above).

The formalized C-criterion problem, illustrated for two classes, is as follows:

Given an example, x, we have two SVM scores s_1 and s_2 for two classes c_1 and c_2, respectively. Without loss of generality, assume that $s_1 > s_2$. Let $dif = s_1 - s_2 > 0$.

Problem: Should we label x with only c_1 or with both c_1 and c_2?

We use MAP to answer the question:

E_1: Event labeling x with c_1

E_2: Event labeling x with c_1 *and* c_2

Our decision is:

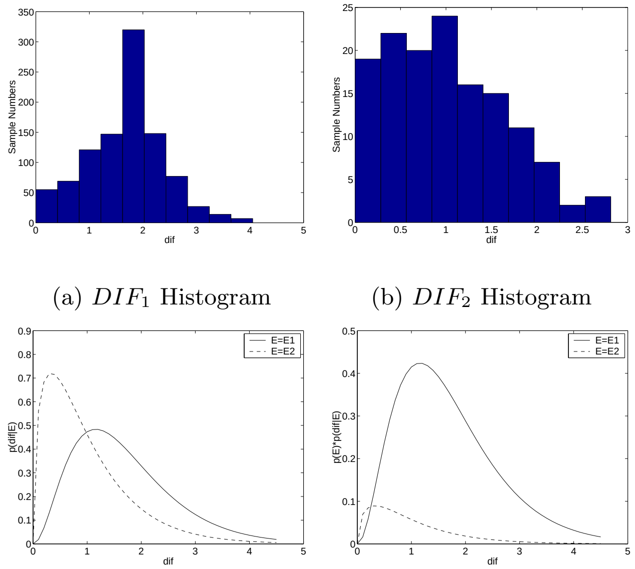
$$\begin{aligned} E &= \arg\max_i p(E_i \mid dif) \\ &= \arg\max_i p(E_i) \cdot p(dif \mid E_i) \end{aligned}$$

The probabilities of $p(dif \mid E_i)$ are calculated from the training data. We apply the SVM models obtained by cross-training to classify the training images. DIF_1 and DIF_2 stand for two difference sets below. We then fit Gamma distributions to the two sets (since the data is non-negative).

DIF_1: the set of differences between the top-two SVM scores for each correctly-labeled *single-class* training image.

DIF_2: the set of differences between the SVM scores corresponding to the multiple classes for each *multiple-class* image.

Figure 3 shows the histograms and distributions of the two difference sets in our experiments. Figure 3(c) shows the two distributions obtained by fitting Gamma distributions to the histograms in our experiment. Figure 3(d) shows the curves obtained by multiplying the distributions in (c) by $p(E_i)$. The x-axis value of the cross point, T_x, is the desired threshold. If the difference of two SVM scores is bigger than T_x, $E = E_1$. Otherwise, $E = E_2$. It can be proven mathematically that this choice of T_x minimizes the expected error in the model (details in Ref. 12).



(a) DIF_1 Histogram

(b) DIF_2 Histogram

(c) Curves of $p(dif \mid E_1)$ and $p(dif \mid E_2)$

(d) Curves of $p(E_1) * p(dif \mid E_1)$ and $p(E_2) * p(dif \mid E_2)$

Figure 3. Histogram and distribution graph for threshold determination in C-criterion

4. EVALUATING MULTI-LABEL CLASSIFICATION

Evaluating multi-label examples is different from evaluating classic single-label examples. Standard evaluation metrics include precision, recall, accuracy, and F-measure.[17] In multi-label classification, evaluation is more complicated, because a result can be fully correct, partly correct, or fully wrong. Take, for example, an example belonging to classes c_1 and c_2. We may get one of the following results:

correct	$\{c_1, c_2\}$
partly correct	$\{c_1\}$,$\{c_1$,$c_3\}$, $\{c_1$,c_3,$c_4\}$
wrong	$\{c_3, c_4\}$

The above five results are different from each other in the degree of correctness.

Schapire[2] used metrics all customized for *ranking* tasks: one-error, coverage, and precision. One-error evaluates how many times the top-ranked label is not in the set of ground truth labels. This measure is used to compare with single label classification, but is not appropriate for the (non-ranked) multi-label case. Coverage measures how far one needs, on average, to go down the list of labels in order to cover all the ground truth labels. These two measures can only reflect some aspects of the classifiers' performance in ranking.

We propose two novel general evaluation metrics for multi-label classifiers.

4.1. α-Evaluation

Suppose Y_x is the set of ground truth labels for test data x, and P_x is the set of predicted labels from classifier h. Furthermore, let $M_x = Y_x - P_x$ (missed labels) and $F_x = P_x - Y_x$ (false positive labels). In α-evaluation, each prediction is scored by the following formula:

$$score(P_x) = \left(1 - \frac{\mid \beta M_x + \gamma F_x \mid}{\mid Y_x \cup P_x \mid}\right)^{\alpha}$$

$$(\alpha \geq 0, 0 \leq \beta, \gamma \leq 1, \beta = 1 | \gamma = 1)$$

The constraints on β and γ are chosen to constrain the score to be non-negative. These parameters allow false positives and misses to be penalized differently, allowing the evaluation measure to be customized to the application. Setting $\beta = \gamma = 1$ yields the simpler formula:

$$score(P_x) = \left(\frac{\mid Y_x \cap P_x \mid}{\mid Y_x \cup P_x \mid}\right)^{\alpha} \quad (\alpha \geq 0)$$

We call α the *forgiveness rate* because it reflects how much to forgive errors made in predicting labels. Small values of α are more aggressive (tend to forgive errors), and big values are conservative (penalizing errors more harshly). In the limits, when $\alpha = \infty$, score $= 1$ only when the prediction is fully correct and 0 otherwise (most conservative); when $\alpha = 0$, score $= 1$ except when the answer is fully wrong (most forgiving). In the single-label case, the score reduces to 1 if the prediction is correct or 0 if incorrect, as expected.

Using this score, we can now define the accuracy rate on a testing data set, D:

$$accuracy_D = \frac{1}{|D|} \sum_{x \in D} score(P_x)$$

Our α-evaluation metric is a generalized version of the *Jaccard similarity* metric of two sets P and Q,[18] augmented with the forgiveness rate and with weights on $P - Q$ and $Q - P$ (misses and false positives, in our case). This evaluation formula provides a flexible way to evaluate the multi-label classification results for both conservative and aggressive tasks.

4.2. Base-class Evaluation

To evaluate recall and precision of each base class, we extend the classic definitions. As above, let Y_x be the set of true labels for example x and P_x be the set of predicted labels from classifier h. Let $H_x^c = 1$ if $c \in Y_x$ and $c \in P_x$ ("hit" label), 0 otherwise. Likewise, let $\tilde{Y}_x^c = 1$ if $c \in Y_x$, 0 otherwise, and let $\tilde{P}_x^c = 1$ if $c \in P_x$, 0 otherwise.

Then we can define *multi-label class recall* and *precision* on data set D:

$$recall_c = \frac{\sum_{x \in D} H_x^c}{\sum_{x \in D} \tilde{Y}_x^c}$$

$$precision_c = \frac{\sum_{x \in D} H_x^c}{\sum_{x \in D} \tilde{P}_x^c}$$

This evaluation measures the performance of the system based on the performance on each base class, which is consistent with the fact that the latter performance reflects the former performance.

5. EXPERIMENTAL RESULTS

We applied the above training and testing methods to semantic scene classification. As discussed in the Introduction, applications of scene classification include content-based image analysis and organization and content-sensitive image enhancement. We now describe our feature selection and baseline classifier.

5.1. Classification System and Features

Color information has been shown to be fairly effective in distinguishing between certain types of outdoor scenes.[19] Furthermore, spatial information appears to be important as well: bright, warm colors at the top of an image may correspond to a sunset, while those at the bottom may correspond to desert rock. Therefore, we use spatial color moments in Luv-space[13, 19] as features to take advantage of luminance-chrominance decomposition, dividing the image into 49 blocks using a 7x7 grid and computing the mean and variance of each band. This gives a 49 x 2 x 3 = 294-dimension feature vector per image.

We use a Support Vector Machine (SVM)[15] as a classifier. SVM classifiers have been shown to give better performance than other classifiers on similar problems.[13] We use a Gaussian kernel and extend the SVM to multi-label scene classification using the training and testing methods described in Section 3. We use the set of images shown in Table 1. Approximately 7.4% of the images belong to multiple classes.

In the next section, we compare the classification results obtained by various training models. Specifically, we compare the cross-training model *Model-x* with *Model-s* and *Model-i*, obtained by training on data labeled by the (subjectively) most obvious class and by ignoring the multi-label data, respectively (Section 3.1).

In Section 3.2, we proposed three criteria to adjudicate the scores output for each base class. We present classification results of the three models using each of the three criteria. As a comparison, we will also give the results obtained by applying a naive criterion, *T1-Criterion*, as a baseline. The *T1-criterion* is to select only the top score as the class label for an input testing image no matter how many SVM scores are positive (the normal "one-vs-all" scheme in single-label classification). An additional naive criterion, *A-Criterion*, that selects all possible classes as the class labels for every testing image, would cause 100% recall and extremely low precision and is not shown.

5.2. Results

Table 2 shows the average recall and precision rate of the six base classes for *Model-s*,*Model-i* and *Model-x* under the five testing criteria. *Model-x*, the model obtained by cross-training, yielded the best results regardless of the criterion used.

Table 2 also shows the α-accuracy of *Model-s*, *Model-i* and *Model-x*, with the highest accuracy at each α-value shown in bold. We see that *Model-x* obtained the highest accuracy regardless of α using this metric as well. *Model-x*'s accuracy is statistically significantly higher than *Model-s* (0.01 significance level) and than *Model-i* (0.001 significance level), where confidence in the increase is measured by ($1-$significance).

We also see that the C-criterion favors higher recall and the T-criterion favors higher precision. Otherwise, their performance is similar and should be chosen based on the application.

Table 3 contains the recall and precision rates of the base classes for *Model-s*, *Model-i* and *Model-x* using the C-Criterion. We see that the precision and recall are slightly higher for *Model-x* in general.

We can see that *Model-x* outperforms the other models in a multi-label classification task. Table 4 shows that for the single-label classification task (where test examples are labeled with the single most obvious class), *Model-x* also outperforms the other models using T-Criterion. This is expected because *Model-x* is a richer training set with more exemplars per class. We note that caution should be used when comparing the accuracy of the single-label and the multi-label paradigms. Multi-label classification in general is a more difficult problem, because one is attempting to classify *each* of the classes of each example correctly (as opposed to only the most obvious one). The results with $\alpha = 1$ reflect this. With more forgiving values of α, multi-label classification accuracy is higher than single-label accuracy.

Table 2. Average base-class recall and precision rates and α-accuracy of the three models (**S**ingle class, **I**gnore, and **X**-training) under 5 criteria:**T**op **1**, **A**ll, **P**ositive, **T**op negative, and **C**lose.

			Accuracy (α-value)		
Criterion	Recall	Precision	0	1	∞
Model-s					
T1	75.0	80.4	80.3	76.3	72.3
P	61.9	**87.1**	66.0	62.3	58.7
T	75.5	80.1	80.7	76.3	71.8
C	77.6	78.0	82.5	76.3	70.2
Model-i					
T1	74.3	79.8	79.7	75.8	71.8
P	60.8	88.5	64.7	61.3	57.9
T	75.0	79.5	80.3	75.9	71.5
C	77.3	77.1	82.5	75.9	69.3
Model-x					
T1	**75.7**	**81.4**	81.2	77.2	**73.2**
P	**64.4**	87.0	68.0	64.3	60.6
T	**77.1**	**80.9**	81.8	**77.4**	73.1
C	**79.0**	**79.2**	**83.4**	77.4	71.4

Table 3. Base-class (beach, sunset, foliage, field, mountain, and urban) recall and precision rates of *Model-s*, *Model-i* and *Model-x* under C-Criterion.

Class	*Model-s*		*Model-i*		*Model-x*	
	recall	prec	recall	prec	recall	prec
BH	85.0	69.4	80.0	72.1	83.0	71.2
ST	89.4	92.7	90.5	91.4	89.4	93.2
FE	91.5	83.2	88.5	80.8	91.0	84.3
FD	77.6	86.4	79.3	85.8	80.2	89.2
MN	53.1	64.5	56.3	63.4	60.5	65.1
UN	68.6	72.1	69.6	69.2	69.6	72.0

Table 4. Accuracy of *Model-s*, *Model-i* and *Model-x* on both single-label and multi-label test cases. For multi-label case, we use T-criterion. See text for caveats in comparing accuracy in single- to multi-label cases.

Model	single-label	multi-label	
		$\alpha = 1$	$\alpha = 0$
Model-s	78.3	76.3	80.7
Model-i	77.6	75.9	80.3
Model-x	79.5	77.4	81.8

6. DISCUSSIONS

As shown in Table 1, some combined classes contain very few examples. The above experimental results show that the increase in accuracy due to the cross-training model is statistically significant; furthermore, these good multi-label results are produced even without an abundance of training data.

We now analyze the results obtained by using C-criterion and cross-training. The images in Fig. 4 are correctly labeled by the classifiers. Among the SVM scores for Fig. 4(b), the scores corresponding to the two real classes are both positive and others are negative. For the image in Fig. 4(a), all of the 6 SVM scores are negative:

```
-0.182 -2.187 -1.455 -1.665 -1.090 -0.199
```

However, because the two scores corresponding to the correct classes (1-beach and 6-urban) are the top two and are very close in magnitude to each other, the C-criterion labels the image correctly.

Other images are classified somewhat correctly or completely incorrectly. We emphasize that we used color features alone in our experiments, and the results should only be interpreted in this feature space. Other features, such as edge direction histograms, may discriminate some of the classes better (e.g., mountain vs. urban).[19]

In Fig. 5, the predictions are subsets of the real class sets. Although those images are not labeled fully correctly, the SVM scores of those images show that the scores of the real classes are the top ones. For instance, in the SVM scores for the image in Fig. 5(b),

```
-0.351 -1.349 -0.913 -1.355 -0.524 -1.212
```

the top two scores (1-beach and 5-mountain) are correct, but their difference is above the threshold and the image is considered to have one label.

In Fig. 6 are images whose predicted class sets are supersets of the true class sets. It is understandable why the image on the right was classified as a mountain (as well as the true class, field).

The panoramic scene in Fig. 7(a) is labeled correctly as a mountain. The foliage is weakly colored, causing it to miss that class. It is unclear why it was also classified as a beach. Figure 7(b) is an atypical beach+mountain image containing little water. In addition, most of the mountain is covered in green foliage, which the classifier interpreted as a field, completely wrong. We emphasize that the color features appear to be the limiting feature in the classification.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an extensive comparative study of possible approaches to training and testing in multi-label classification. The key features of our work include: **(1)** a new training strategy, *cross-training*, to build classifiers. Experimental results show that this training strategy is more efficient in using training data and more effective in classifying multi-labeled data; **(2)** various classification criteria in testing. The *C-Criterion* using a threshold selected by the MAP principle is effective for multi-label classification; **(3)** Two novel evaluation metrics, base-class- and *α-evaluation*. To evaluate multi-label classification performance, α-evaluation can be used in a wide variety of settings. Advantages of our approach include simplicity and effective use of limited training data. Furthermore, these approaches seem to generalize to other problems and other classifiers, particularly those that produce real-valued output, such as neural networks (ANN) and radial basis functions (RBF).

In the scene classification experiment, our data is sparse for some combined classes. We would like to apply our methods to a task with a large amount of data for each single and multiple class. We expect the increase in performance to be much more pronounced.

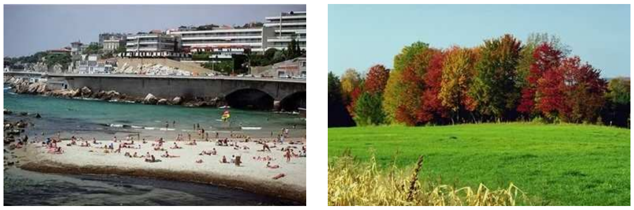
Our techniques were demonstrated on the SVM classifier, but we are interested in generalizing our methods to other classifiers. For neural networks, one possible extension is to allow the target vector to contain multiple +1s, corresponding to the multiple classes to which the example belongs. We are also investigating extensions to RBF classifiers.

ACKNOWLEDGMENTS

Boutell and Brown were supported by a grant from Eastman Kodak Company, by the NSF under grant number EIA-0080124, and by the Department of Education (GAANN) under grant number P200A000306. Shen was supported by DARPA under grant number F30602-03-2-0001. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Eastman Kodak Company or these agencies.

REFERENCES

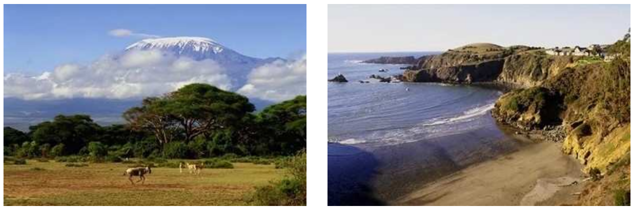
1. R. Duda, R. Hart, and D. Stork, *Pattern Classification*, John Wiley and Sons, Inc., New York, 2nd ed., 2001.
2. R. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine Learning* **39**(2/3), pp. 135–168, 2000.
3. A. McCallum, "Multi-label text classification with a mixture model trained by em," in *AAAI'99 Workshop on Text Learning*, 1999.
4. A. Clare and R. D. King, "Knowledge discovery in multi-label phenotype data," *Lecture Notes in Computer Science* **2168**, pp. 42–??, 2001.
5. Q. Iqbal and J. Aggarwal, "Retrieval by classification of images containing large manmade objects using perceptual grouping," *Pattern Recognition* **35**, pp. 1463–1479, 2001.
6. A. Oliva and A. Torralba, "Scene-centered description from spatial envelope properties," in *2nd Workshop on Biologically Motivated Computer Vision*, *Lecture Notes in Computer Science*, (Tuebingen, Germany), 2002.
7. J. Fan, Y. Gao, H. Luo, and M.-S. Hacid, "A novel framework for semantic image classification and benchmark," in *ACM SIGKDD Workshop on Multimedia Data Mining*, 2003.
8. M. Boutell, J. Luo, and R. T. Gray, "Sunset scene classification using simulated image recomposition," in *Internation Conference on Multimedia Expo (ICME)*, (Baltimore, MD), July 2003.
9. T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European Conference on Machine Learning (ECML)*, Springer, 1998.
10. N. W. Campbell, W. P. J. Mackeown, B. T. Thomas, and T. Troscianko., "The automatic classification of outdoor images," in *International Conference on Engineering Applications of Neural Networks*, pp. 339–342, Systems Engineering Association, June 1996.
11. X. Shi and R. Manduchi, "A study on bayes feature fusion for image classification," in *Workshop on Statistical Analysis in Computer Vision*, (Madison, WI), June 2003.
12. M. Boutell, X. Shen, J. Luo, and C. Brown, "Multi-label semantic scene classification," Tech. Rep. 813, University of Rochester, Rochester, NY, September 2003.
13. Y. Wang and H. Zhang, "Content-based image orientation detection with support vector machines," in *IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL2001)*, (Kauai, Hawaii USA), December 14 2001.
14. A. Vailaya, H.-J. Zhang, C.-J. Yang, F.-I. Liu, and A. K. Jain, "Automatic image orientation detection," *IEEE Transactions on Image Processing* **11**, pp. 746–755, July 2002.
15. U. H.-G. Kre*β*el, *Advances in Kernel Methods: Support Vector Learning*, ch. 15, pp. 255–268. MIT Press, Cambridge, MA, 1999.
16. D. Tax and R. Duin, "Using two-class classifiers for multi-class classification," in *International Conference on Pattern Recognition*, (Quebec City, QC, Canada), August 2002.
17. F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys* **34**(1), 2002.
18. J. C. Gower and P. Legendre, "Metric and euclidean properties of dissimilarity coefficients," *Journal of Classification* **3**, pp. 5–48, 1986.
19. A. Vailaya, M. Figueiredo, A. Jain, and H. Zhang, "Content-based hierarchical classification of vacation images," in *Proc. IEEE Multimedia Systems '99 (International Conference on Multimedia Computing and Systems)*, (Florence, Italy), June 1999.



(a) real:BH+UN, prediction:BH+UN

(b) real:FE+FD, prediction:FE+FD

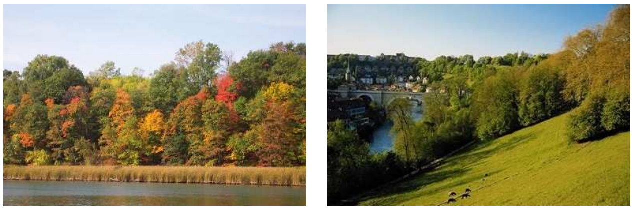
Figure 4. Some images whose prediction sets are completely right by using *Model-x* and C-criterion



(a) real:FD+MN, prediction:FD

(b) real:BH+MN, prediction:BH

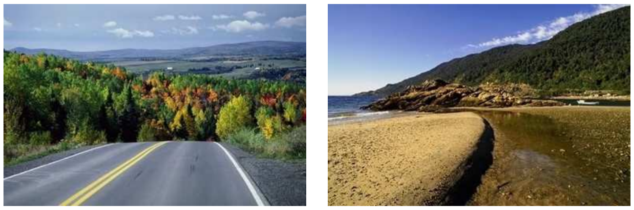
Figure 5. Some images whose prediction sets are subsets of their real class sets



(a) real:FE, prediction:FE+FD

(b) real:FD, prediction:FD+MN

Figure 6. Some images whose real class sets are subsets of their prediction sets



(a) Real:MN+FE, prediction:MN+BH

(b) real:BH+MN, prediction:FD

Figure 7. Some images whose prediction sets are either partially or completely wrong