Learning multi-label scene classification☆

Matthew R. Boutell[a], Jiebo Luo[b,*], Xipeng Shen[a], Christopher M. Brown[a]

[a]*Department of Computer Science, University of Rochester, Rochester, NY 14627, USA*
[b]*Research and Development Laboratories, Eastman Kodak Company, 1700 Dewey Avenue, Rochester, NY 14650-1816, USA*

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 semantic 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.
© 2004 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.

Keywords: 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. 2a). Various classification methods have been developed to provide different operating characteristics, including linear discriminant functions, artificial neural networks (ANN), k-nearest-neighbor (k-NN), radial basis functions (RBF) 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]. Architecture may belong to multiple genres as well. 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 [5–18], categorizing images into semantic classes such as *beaches*, *sunsets* or *parties*. Semantic scene classification finds application in many areas, including content-based indexing and organization and content-sensitive image enhancement.

Many current digital library systems allow a user to specify a query image and search for images "similar" to it, where
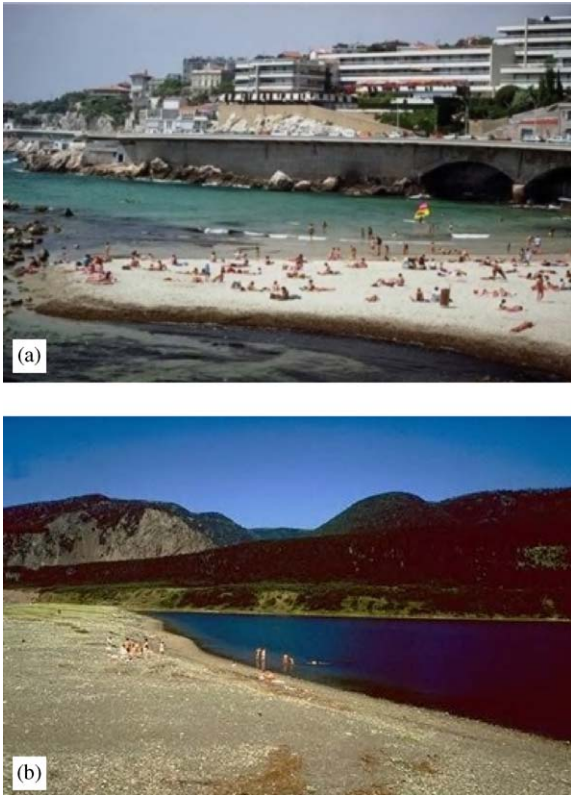

Fig. 1. Examples of multi-label images.

similarity is often defined only by color or texture properties. This the so-called "query by example" process has often proved to be inadequate [19]. Knowing the category of a scene helps narrow the search space dramatically, reducing the search space, and simultaneously increasing the hit rate and reducing the false alarm rate.

Knowledge about the scene category can find also application in context-sensitive image enhancement [16]. While an algorithm might enhance the quality of some classes of pictures, it can degrade others. Rather than applying a generic algorithm to all images, we could customize it to the scene type (allowing us, for example, to retain or enhance the brilliant colors of sunset images while reducing the warm-colored cast from tungsten-illuminated scenes).

In the scene classification domain, many images may belong to multiple semantic classes. Fig. 1(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 a *full* member of each class (due to multiplicity). Fig. 1(b) (beach and mountains) is similar.

Much research has been done on scene classification recently, e.g., [5–18]. Most systems are exemplar-based, learning patterns from a training set using statistical pattern recognition techniques. A variety of features and classifiers

have been proposed; most systems use low-level features (e.g., color, texture). 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 retrieval 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 that two separate queries be conducted respectively and the intersection of the retrieved images be taken. In a content-sensitive image 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. 2b). 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 $\chi \to B$. 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

$$P(c|x) > T, \quad \forall c \in C.$$

Clearly, the mathematical formulation and its physical meaning are distinctively different from those used in classic pattern recognition. Few papers address this problem (see Section 2), and most of these are specialized for text classification or bioinformatics. 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 that can be tailored to applications needing different error forgiveness.

It is worth noting that multi-label classification is different from fuzzy logic-based classification. Fuzzy logics are used as a *means* to cope with ambiguity in the feature space between multiple classes for a given sample, not as the end for achieving multi-label classification. The fuzzy membership stems from ambiguity and often a de-fuzzification step
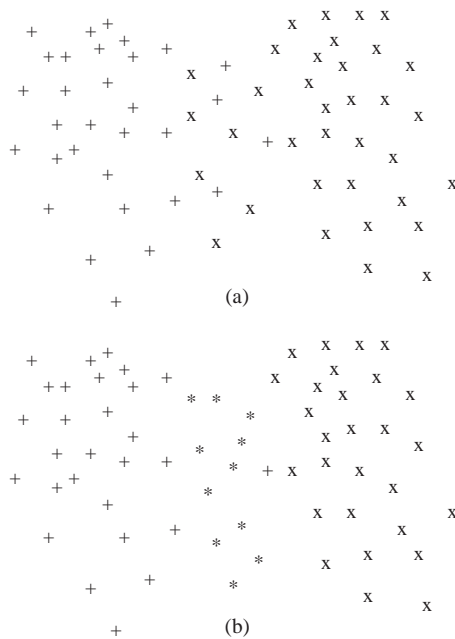
Fig. 2. Figure (a) is the typical pattern recognition problem. Two classes contain examples that are difficult to separate in the feature space. (b) is the multi-label problem. The * data belongs to both of the other two classes simultaneously.

is eventually used to derive a crisp decision (typically by choosing the class with the highest membership value). For example, a foliage scene and a sunset scene may share some warm, bright colors, therefore there is confusion between the two scene classes in the selected feature space if color features are used; fuzzy logic would be suitable for solving this problem.

In contrast, multi-label classification is a unique *problem* in that a sample may possess multiple properties of multiple classes. The content for different classes can be quite distinct: for example, there is little confusion between beach (sand, water) and city (buildings).

The only commonalty between fuzzy-logic classification and multi-class classification is the use of membership functions. However, there is *correlation* between fuzzy membership functions: when one membership takes low values, the other also takes low values or high values and vice versa [20]. On the other hand, the membership functions in multi-label case are largely *coincidence* (e.g., resort on the beach). In practice, the sum of fuzzy memberships usually is normalized to 1, while no such constraints apply to the multi-class problem (e.g., a beach resort scene is both a beach scene and a city scene, each with certainty).

With these differences aside, it is conceivable that one could use the learning strategies described in this paper *in combination with* a fuzzy classifier in a similar way as they were used with the pattern classifiers in this study.

In this paper, we first review past work related to multi-label classification. In Section 3, we describe our training models and testing criteria. Section 4 contains the proposed evaluation methods. Section 5 contains the experimental results obtained by applying our approaches to multi-labeled scene classification. We conclude with a discussion and suggestions for future work.

2. Related work

The sparse literature on multi-label classification is primarily geared to text classification or bioinformatics. For text classification, Schapire and Singer [3] proposed Boos-Texter, extending AdaBoost to handle multi-label text categorization. However, they note that controlling complexity due to overfitting in their model is an open issue. McCallum [2] 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 [21]. He finds that SVM classifiers achieve higher accuracy than others. However, he does not discuss multi-label training models or specific testing criteria. In bioinformatics, Clare and King [4] extended the definition of entropy to include multi-label data (gene expression in their case), but they used a decision tree as their baseline algorithm algorithm. As they stated, they chose a decision tree because of the sparseness of the data and because they needed to learn accurate rules, not a complete classification. However we desire to use Support Vector Machines for their high accuracy in classification.

A related approach to image classification consists of segmenting and classifying image *regions* (e.g., sky, grass) [22,23]. 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 [22]). 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., [5–18]). While many have taken this approach, none handle the multi-label problem. Furthermore, none of the approaches discussed above can be used directly for scene classification.

The main contribution of this work is an extensive comparative study of possible approaches to training and testing multi-label classifiers. 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 efficient in using training data and effective in classifying multi-labeled data; (2) various classifying 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*. α-evaluation can be used to evaluate multi-label classification performance 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, in particular, those that produce real-valued output, such as ANN and RBF.

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. 2b).

3.1. Training models with multi-label data

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 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 "single-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 "ignore").

Table 1
Experimental data

Class	Training Images	Testing Images	Total
Beach	194	175	369
Sunset	165	199	364
Fall foliage	184	176	360
Field	161	166	327
Beach+Field	0	1	1
Fall foliage+Field	7	16	23
Mountain	223	182	405
Beach+Mountain	21	17	38
Fall foliage+Mountain	5	8	13
Field+Mountain	26	49	75
Field+Fall foliage+Mountain	1	0	1
Urban	210	195	405
Beach+Urban	12	7	19
Field+Urban	1	5	6
Mountain+Urban	1	0	1
Total	1211	1196	2407

See Section 5.1 for details of ground truth labeling and split into training and testing sets.

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 (the '*' class) and build a model for it. We call the model trained by this method *MODEL-n* (n stands for "new" class). However, one important problem with this approach is that the data belonging to multiple classes are usually too sparse to build usable models. Table 1 shows the number of various images in our training data. 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.

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. 3. The area A belongs to both the '+' and 'x' classes. When classifying a testing image in area A, the models of '+' and 'x' are 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*. Experiments show that cross-training
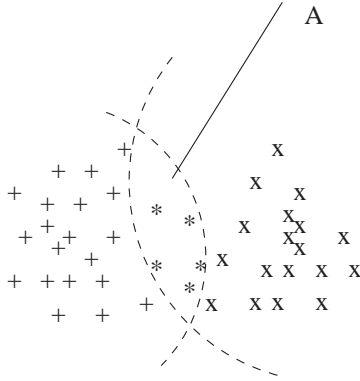
Fig. 3. Illustration of cross-training.

is effective in classifying multi-label images. We call the model obtained using this approach as *MODEL-x* (*x* stands for "**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 [24,25] and text [21] 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 the 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 basic class models.

To simplify our discussion, we use the SVM as an example classifier [26]. In the one-vs-all approach, one classifier is trained for each of the N base classes and each outputs a score for a test example [27]. These outputs can be mapped to pseudo-probabilities using a logistic function [28]; thus the magnitude of each can be considered a measure of confidence in the example's membership in the corresponding class.

Whereas for standard 2-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 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-level 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 that labels the image x with single class c_1,
E_2: Event that labels the image x with multiple classes c_1 *and* c_2
Our decision is

$$E = \arg \max_i \ p(E_i \mid dif)$$

$$= \arg \max_i \ p(E_i) \cdot p(dif \mid E_i).$$

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 as follows.

DIF_1: the set of differences between the top-two SVM scores for each correctly labeled *single-class* training image.
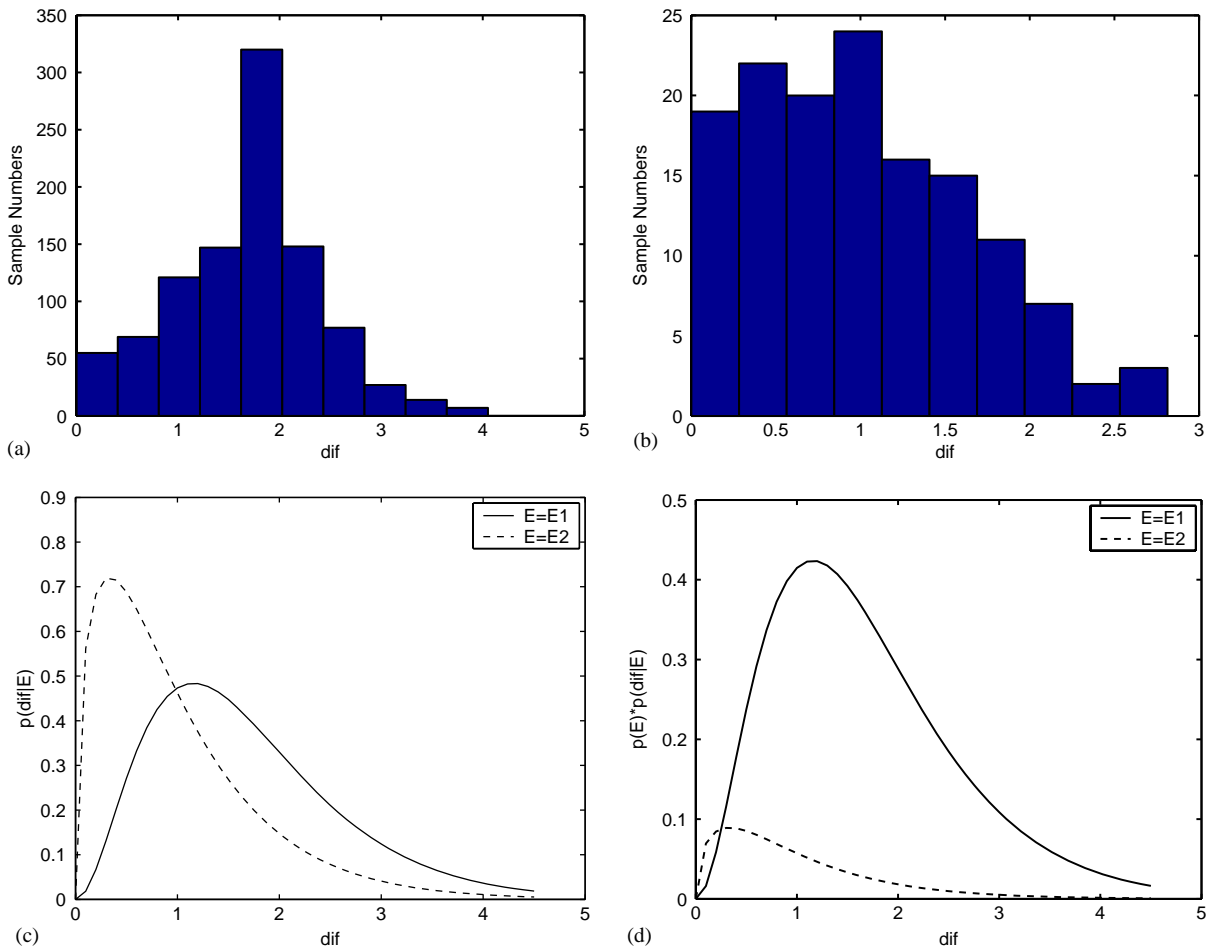
Fig. 4. Histogram and distribution graph for threshold determination in C-Criterion. (a) DIF_1 histogram; (b) DIF_2 histogram; (c) Curves of $p(dif \mid E_1)$ and $p(dif \mid E_2)$; (d) Curves of $p(E_1) * p(dif \mid E_1)$ and $p(E_2) * p(dif \mid E_2)$.

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

We then fit Gamma distributions to the two sets, because the data is non-negative and it appears to be the best fit.

Fig. 4 shows the histograms and distributions of the two difference sets in our experiments. Fig. 4(c) shows the two distributions obtained by fitting Gamma distributions to the histograms in our experiment. Fig. 4(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, $E = E_2$.

Choosing T_x as the decision threshold provably minimizes the decision error in the model. Given an arbitrary threshold T, the decision error is the shaded area in Fig. 5. The area of the shaded region is minimized only when T is the crossing
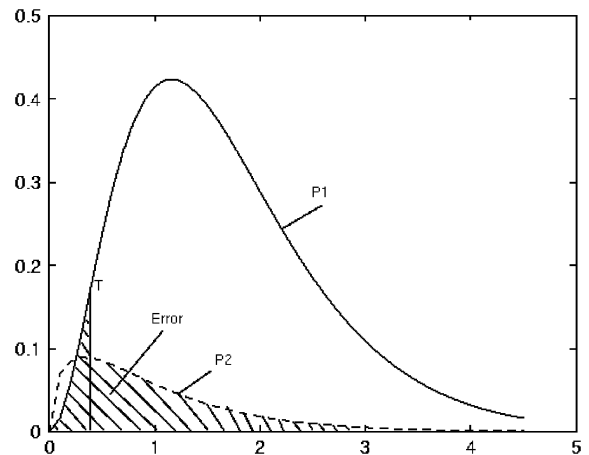

Fig. 5. Illustration of the decision error of using threshold T.

point of the two curves (i.e. $p(E_1) * p(dif | E_1) = p(E_2) * p(dif | E_2)$). The proof follows.

Let $p_1(x)$ and $p_2(x)$ denote two distributions having the following property:

$p_1(x) > p_2(x) \quad when \quad x > T_0,$

$p_1(x) = p_2(x) \quad when \quad x = T_0,$

$p_1(x) < p_2(x) \quad when \quad x < T_0.$

Given a threshold T, for any input x,

if $x > T$, we decide that x is generated from model 1;
if $x \leqslant T$, we decide that x is generated from model 2.

Our claim is that

$T = T_0$ can minimize the decision error.

Proof. Given arbitrary thresholds $T_1 > T_0$ and $T_2 < T_0$, we will show that error E_1 and E_2 obtained by using T_1 and T_2, respectively, are both greater than E_0, the error obtained by using T_0.

- *Using T_1:*

$$E_1 - E_0 = \left(\int_0^{T_1} p_1(x)\,dx + \int_{T_1}^{\infty} p_2(x)\,dx \right)$$

$$- \left(\int_0^{T_0} p_1(x)\,dx + \int_{T_0}^{\infty} p_2(x)\,dx \right)$$

$$= \int_{T_0}^{T_1} (p_1(x) - p_2(x))\,dx$$

$$> 0.$$

- *Using T_2:*

$$E_2 - E_0 = \left(\int_0^{T_2} p_1(x)\,dx + \int_{T_2}^{\infty} p_2(x)\,dx \right)$$

$$- \left(\int_0^{T_0} p_1(x)\,dx + \int_{T_0}^{\infty} p_2(x)\,dx \right)$$

$$= \int_{T_2}^{T_0} (p_2(x) - p_1(x))\,dx$$

$$> 0.$$

This shows that the C-Criterion provides the best trade-off between the performance of the classifier on single-label images and multi-label images. We note our two assumptions: (1) the testing data and the training data have the same distribution and (2) the cost of mis-labeling single-label images is the same as the cost of mis-labeling multi-label ones. We also assume in this discussion that the base classifiers are calibrated, which is the case in the proposed application to scene classification, because the same features and equal numbers of examples are used for each classifier.

4. Evaluating multi-label classification results

Evaluating the performance of multi-label classification is different from evaluating performance of classic single-label classification. Standard evaluation metrics include precision, recall, accuracy, and F-measure [29]. In multi-label classification, the evaluation is more complicated, because a result can be fully correct, partly correct, or fully incorrect. Take an example belonging to classes c_1 and c_2. We may get one of the following results:

1. c_1, c_2 (correct),
2. c_1 (partly correct),
3. c_1, c_3 (partly correct),
4. c_1, c_3, c_4 (partly correct),
5. c_3, c_4 (incorrect).

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

Schapire and Singer [3] used three kinds of measures, 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 good for the 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. Precision is a measure that can be used to assess the system as a whole. It is borrowed from information retrieval (IR) [30]:

$$precision_S(h) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{|Y_i|} \sum_{l \in Y_i}$$

$$\times \frac{|\{l' \in Y_i | rank_h(x_i, l') \leqslant rank_h(x_i, l)\}|}{rank_h(x_i, l)},$$

where h is the classifier, S is the training set, m is the total number of testing data, Y_i is the ground truth labels of an testing data example, x_i is a testing data example, $rank_h(x_i, l)$ is the rank of label l in the prediction ranking list output from h for x_i.

We propose two novel kinds of general evaluation methods for multi-label classification systems.

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 prediction 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:

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

$$(\alpha \geqslant 0, 0 \leqslant \beta, \gamma \leqslant 1, \beta = 1 | \gamma = 1).$$

Table 2
Examples of scores as a function of β and γ when the true label is $\{C_1, C_2\}$ and $\alpha = 1$

Prediction (P)	$\beta = \frac{1}{4}, \gamma = 1$	$\beta = \frac{1}{4}, \gamma = 1$	$\beta = 1, \gamma = 1$	$\beta = 1, \gamma = \frac{1}{4}$	$\beta = 1, \gamma = \frac{1}{4}$
C_1, C_2	1	1	1	1	1
C_1 (1 miss)	0.875	**0.750**	0.500	**0.500**	0.500
C_1, C_2, C_3 (1 false pos.)	0.667	**0.667**	0.667	**0.833**	0.917

Table 3
Example of alpha-evaluation scores as a function of α when the true label is $\{C_1, C_2\}$

Prediction (P)	$\beta = \gamma = 1$				
	$\alpha = 0$	$\alpha = \frac{1}{2}$	$\alpha = 1$	$\alpha = 2$	$\alpha = \infty$
C_1, C_2	1	1	1	1	1
C_1	1	0.71	0.50	0.25	1
C_1, C_3, C_4	1	0.50	0.25	0.06	1
C_3, C_4	0	0	0	0	0

The constraints on β and γ are chosen to constrain the score to be non-negative. The more familiar parameterization, constraining $\gamma = 2 - \beta$, yields negative scores, causing a need to bound the scores below by zero explicitly.

These parameters allow false positives and misses to be penalized differently, allowing the evaluation measure to be customized to the application. Table 2 contains examples showing the effect of β and γ upon the score of an example with true label $\{C_1, C_2\}$.

Setting $\beta = \gamma = 1$ yields the simpler formula:

$$score(P_x) = \left(\frac{|Y_x \cap P_x|}{|Y_x \cup P_x|} \right)^\alpha \quad (\alpha \geqslant 0).$$

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 $\alpha = \infty$, $score(P_x) = 1$ only when the prediction is fully correct and 0 otherwise (most conservative); when $\alpha = 0$, $score = 1$ except when the answer is fully incorrect (most aggressive). In the single-label case, the score also reduces to 1 if the prediction is correct or 0 if incorrect, as expected. Table 3 shows some examples of the effect of α on the score.

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

- *Recall rate* of a *multi-label* class C:

$$recall_C = \frac{1}{|D_C|} \sum_{x \in D_C} score(P_x),$$

where

$$D_C = \{x \mid C = Y_x\}.$$

- *Precision* of a *multi-label* class C:

$$precision_C = \frac{1}{|D_C|} \sum_{x \in D_C} score(P_x),$$

where

$$D_C = \{x \mid C = P_x\}.$$

- *Accuracy* 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 P and Q [31], 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. Let C be the set of base classes.

Then *base-class recall* and *precision* on data set, D, are defined as follows:

- $Recall(c) = \frac{\sum_{x \in D} H_x^c}{\sum_{x \in D} \tilde{Y}_x^c}$,

- $Precision(c) = \frac{\sum_{x \in D} H_x^c}{\sum_{x \in D} \tilde{P}_x^c}$.

- $Accuracy_D = \frac{\sum_{x \in D} \sum_{c \in C} H_x^c}{max\left(\sum_{x \in D} \sum_{c \in C} \tilde{Y}_x^c, \sum_{x \in D} \sum_{c \in C} \tilde{P}_x^c\right)}$.

Intuitively, base-class recall is the fraction of true instances of a label classified correctly, while base-class precision is the fraction of predicted instances of a label that are correct. As an example, for the data set containing five samples shown in Table 4, $Recall(C_1) = \frac{2}{3}$, while $Precision(C_1) = \frac{2}{4}$.

This evaluation measures the performance of the system based on the performance on each base class, which is

Table 4
A toy data set consisting of five samples

True labels	Predicted labels
C_1, C_2	C_1, C_3
C_1	C_1
C_4	C_1, C_3
C_1, C_3	C_3
C_2	C_1

For true and predicted label sets shown, $Recall(C_1) = \frac{2}{3}$ and $Precision(C_1) = \frac{2}{4}$.

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, scene classification finds application in many areas, including content-based image analysis and organization and content-sensitive image enhancement. We now describe our baseline classifier and features and present the results.

5.1. Classification system and features

Color information has been shown to be fairly effective in distinguishing between certain types of outdoor scenes [18]. 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 as features. These features are commonly used in the scene classification literature [18,24,25], but may not necessarily be optimal for the problem.

With color images, it is usually advantageous to use a more perceptually uniform color space such that perceived color differences correspond closely to Euclidean distances in the color space selected for representing the features. For example in image segmentation, luminance-chrominance decomposed color spaces were used by Tu and Zhu [32] and Comaniciu and Meer [33] to remove the nonlinear dependency along RGB color values. In this study, we use a CIE L*U*V*-like space, referred to as Luv (due to the lack of a true white point calibration), similar to [32,33]. Both the CIE L*a*b* and L*U*V* spaces have good approximate perceptual uniformity, but the L*U*V* has lower complexity in its mapping.

After conversion to Luv space, the image is divided into 49 blocks using a 7×7 grid. We compute the first and second moments (mean and variance) of each band, corresponding to a low-resolution image and to computationally inexpensive texture features, respectively. The end

Table 5
Average *base-class* recall, precision, and 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

Model	Criterion	Recall	Precision	Accuracy
s	T1-Criterion	75.0	80.4	72.0
	A-Criterion	100.0	18.1	18.7
	P-Criterion	61.9	**87.1**	58.9
	T-Criterion	75.5	80.1	72.5
	C-Criterion	77.6	78.0	74.9
i	T1-Criterion	74.3	79.8	71.6
	A-Criterion	100.0	18.1	18.7
	P-Criterion	60.8	88.5	57.8
	T-Criterion	75.0	79.5	72.3
	C-Criterion	77.3	77.1	74.6
x	T1-Criterion	**75.7**	**81.4**	**72.9**
	A-Criterion	100.0	18.1	18.7
	P-Criterion	**64.4**	87.0	**63.5**
	T-Criterion	**77.1**	**80.9**	**74.9**
	C-Criterion	**79.0**	79.2	**76.7**

result is a $49 \times 2 \times 3 = 294$-dimension feature vector per image.

We use a Support Vector Machine (SVM) [26] as a classifier. The software we used is SVMFu [34]. SVM classifiers have been shown to give better performance than other classifiers like Learning Vector Quantization (LVQ) on similar problems [24,25]. We use a Gaussian kernel, creating an RBF-style classifier. The sign of the output corresponds to the class and the magnitude corresponds to the confidence in classification. As a baseline, we used the one-vs-all approach [27]: for each class, an SVM is trained to distinguish that class of images from the rest, test images are classified using each SVM and then labeled with the class corresponding to the SVM which gave the highest score.

We then extended the SVM classifier to multi-label scene classification using the training and testing methods described in Section 3.

For training and testing, we used the set of images shown in Table 1. These 2400 images consist of Corel stock photo library and personal images. The images were originally chosen so that each primary class (according to *Model-s*) contained 400 images, i.e. equal priors. Our framework does not currently incorporate prior probabilities.

Each class was split randomly into independent sets of 200 training and 200 testing images. The images were later re-labeled with multiple labels by three human observers. After re-labeling, approximately 7.4% of the images belonged to multiple classes. An artifact of this process is that for some classes, there are substantially more training than testing images and vice-versa.

Table 6
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.
Beach	85.0	69.4	80.0	72.1	83.0	71.2
Sunset	89.4	92.7	90.5	91.4	89.4	93.2
Fall foliage	91.5	83.2	88.5	80.8	91.0	84.3
Field	77.6	86.4	79.3	85.8	80.2	89.2
Mountain	53.1	64.5	56.3	63.4	60.5	65.1
Urban	68.6	72.1	69.6	69.2	69.6	72.0

Table 7
α-Accuracy of *Model-s*, *Model-i* and *Model-x* for multi-label classification for original and mirror data sets

Model	Crit.	Original set accuracy (α-value)				Mirror set accuracy (α-value)			
		$\alpha = 0$	$\alpha = 1.0$	$\alpha = 2.0$	$\alpha = \infty$	$\alpha = 0$	$\alpha = 1.0$	$\alpha = 2.0$	$\alpha = \infty$
s	T1	80.3	76.3	74.3	72.3	79.5	75.6	73.7	71.7
	A	100.0	18.1	3.50	0	100	18.1	3.50	0
	P	66.0	62.3	60.5	58.7	67.0	63.2	61.3	59.4
	T	80.7	76.3	74.0	71.8	80.3	75.8	73.5	71.2
	C	82.5	76.3	73.2	70.2	82.2	76.0	72.9	69.9
i	T1	79.7	75.8	73.8	71.8	79.7	75.8	73.8	71.8
	A	100.0	18.1	3.50	0	100.0	18.1	3.50	0
	P	64.7	61.3	59.6	57.9	64.7	61.3	59.6	57.9
	T	80.3	75.9	73.7	71.5	80.3	75.9	73.7	71.5
	C	82.5	75.9	72.6	69.3	82.5	75.9	72.6	69.3
x	T1	81.2	77.2	75.2	**73.2**	80.0	76.0	74.0	72.0
	A	100.0	18.1	3.50	0	100	18.1	3.50	0
	P	68.0	64.3	62.5	60.6	72.3	67.6	65.2	62.9
	T	81.8	**77.4**	**75.3**	73.1	82.4	77.3	74.8	72.3
	C	**83.4**	77.4	74.4	71.4	84.2	77.5	74.3	71.1

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 5 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, yields the best results regardless of the criterion used.

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

Table 7 shows the α-accuracy of *Model-s*, *Model-i* and *Model-x*, with the highest accuracy at each α-value given in bold font. For all four α values, *Model-x* obtained the

Table 8
Accuracy of *Model-s*, *Model-i* and *Model-x* on both single- and multi-label test cases

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

For multi-label case, we use *T*-Criterion. See text for caveats in comparing accuracy in single- to multi-label cases.

highest accuracy. In the most progressive situation, i.e. $\alpha=0$, C-Criterion obtains the highest accuracy, and for all other cases, T-Criterion obtains the highest accuracy.

We also include the results on another dataset, the *mirror set*. This set is obtained by augmenting the original training set with mirror images of each multi-label image. Mirroring an image in the horizontal direction (assuming correct orientation) does not change the classification of an image. We also add multi-label mirror images on the testing set. We assume that the mirror images are classified independently of the original images (which should be true, due to lack of symmetry in the classifier: most of the training images are *not* mirrored). Of course, if the training and testing multi-label images are correlated, this independence assumption is violated.

This mirroring has the effect of artificially adding more multi-label images: while the original set has 177 multi-label and 2230 single-label images (7.4% multi-label images), the new set has 354 multi-label and 2230 single-label images (up to 13.7% multi-label images). We hypothesized that the increases brought about by our method would be more pronounced when a higher percentage of images contain multiple labels.

Model-x outperforms the other models in a multi-label classification task. We see that *Model-x* obtains the highest accuracy regardless of α. *Model-x*'s accuracy is statistically significantly higher than *Model-s* ($P = 0.0027$) significance level) and than *Model-i* ($P = 0.00047$). These values of *P* correspond to the 0.01 and 0.001 significance levels, respectively). Confidence in the increase is measured by $(1 - P)$.

The accuracy on the mirror set is very similar to that on the original set. As expected, the accuracy increases on forgiving values of α (where accuracy on multi-label data is higher than that on single-label data) and decreases on strict values of α, where the opposite is true. However, the changes are not substantial.

Table 8 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). The results with $\alpha = 1$ reflect this. With more forgiving values of α, multi-label classification accuracy is higher than single-label accuracy.

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.[1] The images in Fig. 6 are correctly

Fig. 6. Some images whose prediction sets are completely correct by using *Model-x* and C-Criterion: (a) real: Fall-Fol.+Field, Predicted:FallFol.+Field; (b) real:Beach+Urban, Predicted:Beach+Urban.

---

[1] For color images, see the electronic version or our technical report at `http:\www.cs.rochester.edu\trs\robotics-trs.html`.

Fig. 7. Some images whose prediction sets are subsets of their real class sets: (a) real:Beach+Mountain, Predicted:Beach; (b) real:Field+Mountain, Predicted:Field; (c) real:Field+Mountain, Predicted:Field; (d) real:Field+Mountain, Predicted:Field.

labeled by the classifiers. Among the SVM scores for Fig. 6(a), the scores corresponding to the two real classes are both positive and others are negative. For the image in Fig. 6(b), 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) [18].

In Fig. 7, 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. 7(a),

$$-0.350 - 1.34 - 0.913 - 1.355 - 0.523 - 1.212$$

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. Due to weak coloring, we can also see why the mountains in Fig. 7(b, c) were not detected.

In Fig. 8 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).

In Fig. 9, the prediction is partially correct (mountain), but also partially incorrect. The foliage is weakly colored, causing it to miss that class. It is unclear why it was also classified as a beach.

In Fig. 10, the image is labeled completely incorrectly, due to differences between the training and testing images. The atypical beach+mountain image contains little water. In addition, most of the mountain is covered in green foliage, which the classifier interpreted as a field. We emphasize that the color features appear to be the limiting feature in the classification.

Fig. 8. Some images whose real class sets are subsets of their prediction sets: (a) real:Beach, Predicted:Beach+Mountain; (b) real:Field, Predicted:Field+Mountain; (c) real:Mtn., Predicted:Urban+Mtn.+Beach; (d) real:FallFol., Predicted:FallFol.+Field.

Fig. 9. An image whose prediction set is partly correct and partly incorrect (real:Mountain+FallFoliage, Predicted:Mountain+Beach).

Fig. 10. An image whose prediction set is completely incorrect (real:Beach+Mountain, Predicted:Field).

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. In particular, we contribute the following:

- *Cross-training*, a new training strategy to build classifiers. Experimental results show that cross-training is more

efficient in using training data and more effective in classifying multi-label data.

- *C-Criterion* using threshold selected by MAP principle is effective for multi-label classification. Other classification criteria were proposed as well which may be better suited to different tasks where higher precision is more important than high recall.
- *α-Evaluation*, our novel generic evaluation metric, provides a way to evaluate multi-label classification results in a wide variety of settings. Another metric, base-class evaluation, provides a valid comparison with standard single-class recall and precision.

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, in particular, 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 $+1$s, corresponding to the multiple classes to which the example belongs. We are also investigating extensions to RBF classifiers.

Acknowledgements

References

[1] R. Duda, R. Hart, D. Stork, Pattern Classification, 2nd Edition, Wiley, New York, 2001.

[2] A. McCallum, Multi-label text classification with a mixture model trained by EM, in: AAAI'99 Workshop on Text Learning, 1999.

[3] R. Schapire, Y. Singer, Boostexter: a boosting-based system for text categorization, Mach. Learning 39 (2/3) (2000) 135–168.

[4] A. Clare, R.D. King, Knowledge Discovery in Multi-label Phenotype Data, in: Lecture Notes in Computer Science, Vol. 2168, Springer, Berlin, 2001.

[5] M. Boutell, J. Luo, R.T. Gray, Sunset scene classification using simulated image recomposition, in: International Conference on Multimedia and Expo, Baltimore, MD, July 2003.

[6] C. Carson, S. Belongie, H. Greenspan, J. Malik, Recognition of images in large databases using a learning framework, Technical Report 97-939, University of California, Berkeley, 1997.

[7] J. Fan, Y. Gao, H. Luo, M.-S. Hacid, A novel framework for semantic image classification and benchmark, in: ACM SIGKDD Workshop on Multimedia Data Mining, 2003.

[8] Q. Iqbal, J. Aggarwal, Retrieval by classification of images containing large manmade objects using perceptual grouping, Pattern Recognition 35 (2001) 1463–1479.

[9] P. Lipson, E. Grimson, P. Sinha, Configuration based scene classification and image indexing, 1997. Proc - IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico.

[10] A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, Int. J. Comput. Vision 42 (3) (2001) 145–175.

[11] A. Oliva, A. Torralba, Scene-centered description from spatial envelope properties, in: Second Workshop on Biologically Motivated Computer Vision, Tuebingen, Germany, Lecture Notes in Computer Science, Springer, Berlin, 2002.

[12] S. Paek, S.-F. Chang, A knowledge engineering approach for image classification based on probabilistic reasoning systems, in: IEEE International Conference on Multimedia and Expo, Vol. II, New York City, NY, Jul 30–Aug 2, 2000, pp. 1133–1136.

[13] N. Serrano, A. Savakis, J. Luo, A computationally efficient approach to indoor/outdoor scene classification, in: International Conference on Pattern Recognition, September 2002.

[14] J.R. Smith, C.-S. Li, Image classification and querying using composite region templates, Comput. Vision Image Understanding 75 (1999) 165–174.

[15] Y. Song, A. Zhang, Analyzing scenery images by monotonic tree, ACM Multimedia Systems J. 8 (6) 495–511 (2003).

[16] M. Szummer, R.W. Picard, Indoor–outdoor image classification, in: IEEE International Workshop on Content-based Access of Image and Video Databases, Bombay, India, 1998.

[17] A. Torralba, P. Sinha, Recognizing indoor scenes, Technical Report, AI Memo 2001-015, CBCL Memo 202, MIT, July 2001.

[18] A. Vailaya, M. Figueiredo, A. Jain, H. Zhang, Content-based hierarchical classification of vacation images, in: Proceedings of the IEEE Multimedia Systems '99, International Conference on Multimedia Computing and Systems, Florence, Italy, June 1999.

[19] A. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of early years, IEEE Trans. Pattern Anal. Mach. Intel. 22 (2000) 1349–1380.

[20] S.K. Pal, D.K.D. Majumder, Fuzzy Mathematical Approach to Pattern Recognition, Wiley, New York, 1986, p. 64.

[21] T. Joachims, Text categorization with support vector machines: learning with many relevant features, in: European Conference on Machine Learning (ECML), Springer, Berlin, 1998.

[22] N.W. Campbell, W.P.J. Mackeown, B.T. Thomas, T. Troscianko, The automatic classification of outdoor images, in: International Conference on Engineering Applications of Neural Networks, Systems Engineering Association, June 1996, pp. 339–342.

[23] X. Shi, R. Manduchi, A study on Bayes feature fusion for image classification, in: Workshop on Statistical Analysis in Computer Vision, Madison, WI, June 2003.

[24] Y. Wang, 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.

[25] A. Vailaya, H.-J. Zhang, C.-J. Yang, F.-I. Liu, A.K. Jain, Automatic image orientation detection, IEEE Trans. Image Process. 11 (2002) 746–755.

[26] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery 2 (2) (1998) 121–167.

[27] U.H.-G. Kreßel, Advances in Kernel Methods: Support Vector Learning, MIT Press, Cambridge, MA, 1999, pp. 255–268 (Chapter 15).

[28] D. Tax, R. Duin, Using two-class classifiers for multi-class classification, in: International Conference on Pattern Recognition, Quebec City, QC, Canada, August 2002.

[29] F. Sebastiani, Machine learning in automated text categorization, ACM Compu. Surveys 34 (1) (2002) 1–47.

[30] G. Salton, Developments in automatic text retrieval, Science 253 (1991) 974–980.

[31] J.C. Gower, P. Legendre, Metric and euclidean properties of dissimilarity coefficients, J. Classification 3 (1986) 5–48.

[32] Z. Tu, S.-C. Zhu, Image segmentation by data-driven markov chain monte carlo, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 657–673.

[33] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 603–619.

[34] R. Rifkin, Svmfu, http://five-percent-nation.mit.edu/SvmPu, 2000.

About the Author–MATTHEW BOUTELL received the B.S. degree (with High Distinction) in Mathematical Science from Worcester Polytechnic Institute in 1993 and the M.Ed. degree from the University of Massachusetts in 1994. Currently, he is a Ph.D. student in Computer Science at the University of Rochester. He served for several years as a mathematics and computer science instructor at Norton High School and at Stonehill College. His research interests include computer vision, pattern recognition, probabilistic modeling, and image understanding. He is a student member of the IEEE.

About the Author–JIEBO LUO received his Ph.D. degree in Electrical Engineering from the University of Rochester in 1995. He is currently a Senior Principal Research Scientist in the Eastman Kodak Research Laboratories. His research interests include image processing, pattern recognition, and computer vision. He has authored over 80 technical papers and holds 20 granted US patents. Dr. Luo was the Chair of the Rochester Section of the IEEE Signal Processing Society in 2001, and the General Co-Chair of the IEEE Western New York Workshop on Image Processing in 2000 and 2001. He was also a member of the Organizing Committee of the 2002 IEEE International Conference on Image Processing and a Guest Co-Editor for the Journal of Wireless Communications and Mobile Computing Special Issue on Multimedia Over Mobile IP. Currently, he is serving as an Associate Editor of the journal of Pattern Recognition and Journal of Electronic Imaging, an adjunct faculty member at Rochester Institute of Technology, and an At-Large Member of the Kodak Research Scientific Council. Dr. Luo is a Senior Member of the IEEE.

About the Author–XIPENG SHEN received the M.S. degree in Computer Science from the University of Rochester in 2002 and the M.S. degree in Pattern Recognition and Intelligent Systems from the Chinese Academy of Sciences. He is currently a Ph.D graduate student at the Department of Computer Science, University of Rochester. His research interests include image processing, machine learning, program analysis and optimization, speech and language processing.

About the Author–CHRISTOPHER BROWN (B.A. Oberlin 1967, Ph.D. University of Chicago 1972) is Professor of Computer Science at the University of Rochester, where he has been since finishing a postdoctoral fellowship at the School of Artificial Intelligence at the University of Edinburgh in 1974. He is coauthor of COMPUTER VISION with his Rochester colleague Dana Ballard. His current research interests are computer vision and robotics, integrated parallel systems performing animate vision (the interaction of visual capabilities and motor behavior), and the integration of planning, learning, sensing, and control.