Multi-label Semantic Scene Classification

Matthew Boutell[a] Xipeng Shen[a] Jiebo Luo[b] Chris Brown[a1]

[a]Department of Computer Science, University of Rochester
{*boutell,xshen,brown*}*@cs.rochester.edu*
[b]Electronic Imaging Products, R & D, Eastman Kodak Company
luo@image.kodak.com

Technical Report 813
September, 2003
Department of Computer Science
University of Rochester
Rochester, NY 14627

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.

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 [7]:

> 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 $\chi \rightarrow Y$.

[1]Boutell and Brown were supported by a grant from Eastman Kodak Company and 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.

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 (Figure 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) [7].

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* [14, 20]. 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 [5].

A problem domain receiving renewed attention is semantic scene classification [1, 4, 8, 10, 13, 15, 16, 17, 22, 25, 26, 27, 29, 31], 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 similarity is often defined only by color or texture properties. This so-called "query by example" process has often proved to be inadequate [24]. 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 [27]. 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. Figure 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 *fully* a member of each class (due to multiplicity). Figure 1(b) (beach and mountains) is similar.

Much research has been done on scene classification recently, e.g., [1, 4, 8, 10, 13, 15, 16, 17, 22, 25, 26, 27, 29, 31]. 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.

Figure 1: Examples 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** (Figure 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 $\chi \rightarrow B$. 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 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.

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 [20] proposed BoosTexter, 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 [14] 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 [11]. He finds that SVM classfiers achieve higher accuracy than others. However, he does not discuss multi-label training models or specific testing criteria. In bioinformatics, Clare and King [5] 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) [3, 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 [3]). 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., [1, 4, 10, 13, 15, 16, 17, 22, 25, 27, 29, 31]). 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 neural networks (ANN) and radial basis functions (RBF).

3 Multi-label Classification

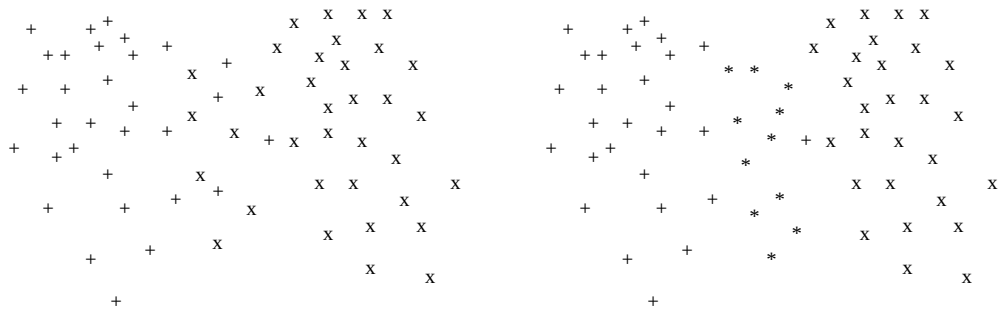
In this section, we describe possible approaches for training and testing with multi-label data. Consider two classes, denoted by '+' and 'x' respectively. Examples belonging to both the '+'and 'x' classes simultaneously are denoted by '*' (see Figure 2b).

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 "**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) (b)

Figure 2: 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.

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 "**n**ew" 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 Figure 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 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

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

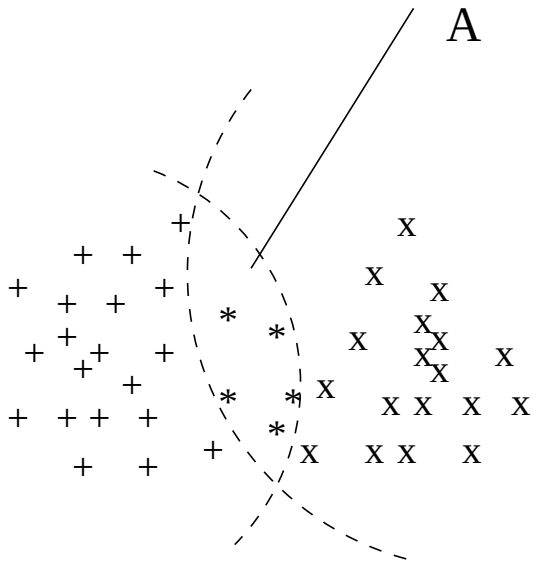


Figure 3: Illustration of Cross-Training

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 [33, 32] and text [11] 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 [2]. 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 [12]. 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-lavel 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:

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

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 as follows.

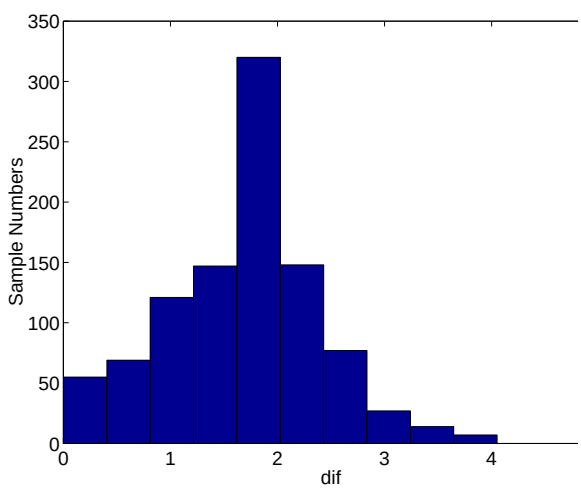
DIF_1: the set of differences between the top-two SVM scores for each correctly-labeled *single-class* training image.

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

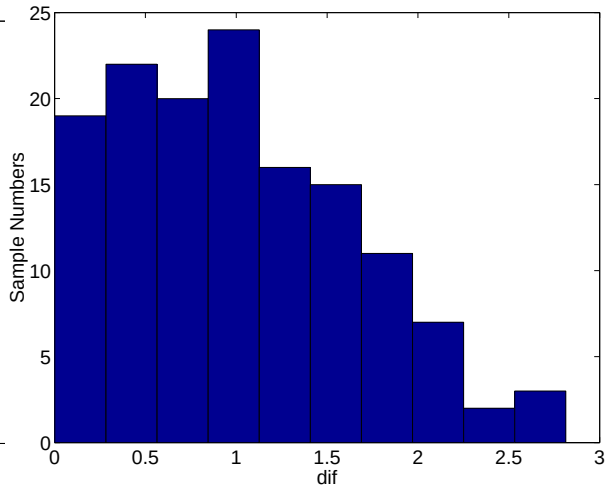
We then fit Gamma distributions to the two sets, because the data is non-negative and it appears to be the best fit.

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

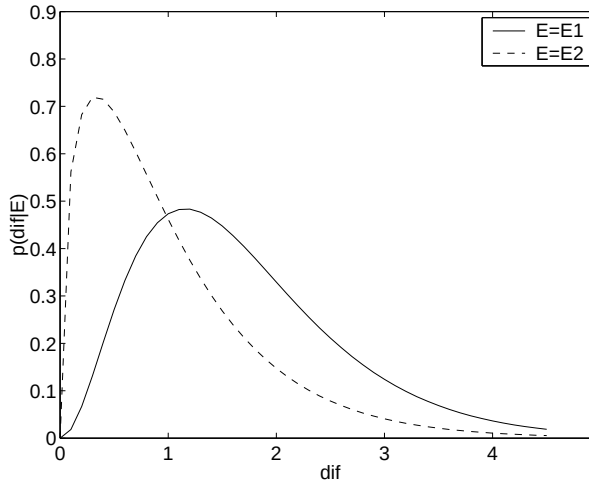
Choosing T_x as the decision threshold provably minimizes the decision error in the model. Given an arbitrary threshold T, the decision error is the shaded area in Figure 5. The area



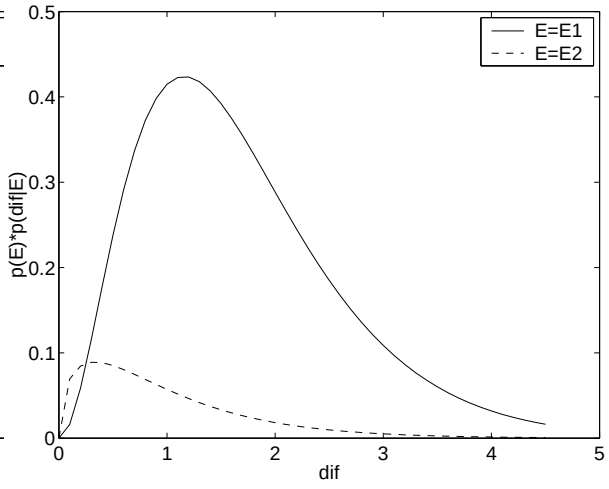
(a) DIF_1 Histogram



(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 4: Histogram and Distribution Graph for Threshold Determination in C-Criterion

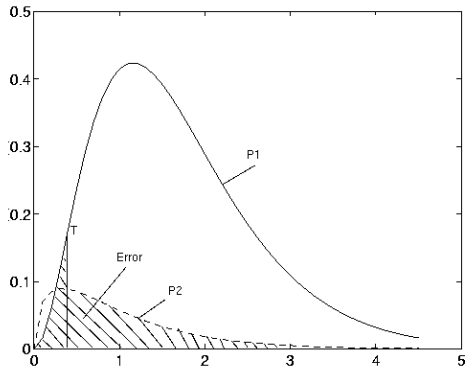


Figure 5: Illustration of the decision error of using threshold T.

of the shaded region is minimized only when T is the crossing point of the two curves (i.e. $p(E_1) * p(dif \mid E_1) = p(E_2) * p(dif \mid E_2)$). The proof follows.

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

$$\begin{cases} p_1(x) > p_2(x) & when \;\; x > T_0 \\ p_1(x) = p_2(x) & when \;\; x = T_0 \\ p_1(x) < p_2(x) & when \;\; x < T_0 \end{cases}$$

Given a threshold T, for any input x,

if $x > T$, we decide that x is generated from model 1;

if $x \leq T$, 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**:

$$\begin{aligned} E_1 - E_0 &= \left(\int_0^{T_1} p_1(x)dx + \int_{T_1}^{\infty} p_2(x)dx\right) - \left(\int_0^{T_0} p_1(x)dx + \int_{T_0}^{\infty} p_2(x)dx\right) \\ &= \int_{T_0}^{T_1} (p_1(x) - p_2(x))dx \\ &> 0 \end{aligned}$$

- **Using T_2**:

$$\begin{aligned} E_2 - E_0 &= (\int_0^{T_2} p_1(x)dx + \int_{T_2}^{\infty} p_2(x)dx) - (\int_0^{T_0} p_1(x)dx + \int_{T_0}^{\infty} p_2(x)dx) \\ &= \int_{T_2}^{T_0} (p_2(x) - p_1(x))dx \\ &> 0 \end{aligned}$$

This shows that the C-Criterion provides the best tradeoff 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.

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 [21]. In multi-label classification, the evaluation is more complicated, because a result can be fully correct, partly correct, or fully wrong. 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 (wrong)

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

In [20], Schapire and Singer 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) [19]:

$$precision_S(h) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{\mid Y_i \mid}\sum_{l\in Y_i}\frac{\mid \{l' \in Y_i \mid rank_h(x_i, l') \leq rank_h(x_i, l)\} \mid}{rank_h(x_i, l)}$$

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:

$$score(P_x) = \left(1 - \frac{\mid \beta M_x + \gamma F_x \mid}{\mid Y_x \cup P_x \mid}\right)^{\alpha} (\alpha \geq 0, 0 \leq \beta, \gamma \leq 1, \beta = 1 | \gamma = 1)$$

The constraints on β and γ are chosen to constrain the score to be non-negative. 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. 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$(P_x) = 1$ only when the prediction is fully correct and 0 otherwise (most conservative); when $\alpha = 0$, score = 1 except when the answer is fully wrong (most aggressive). In the single-label case, the score also reduces to 1 if the prediction is correct or 0 if incorrect, as expected.

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 [9], 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 multilabel 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:

- $\mathbf{Recall_c} = \dfrac{\sum\limits_{x \in D} H_x^c}{\sum\limits_{x \in D} \tilde{Y}_x^c}$
- $\mathbf{Precision_c} = \dfrac{\sum\limits_{x \in D} H_x^c}{\sum\limits_{x \in D} \tilde{P}_x^c}$

- $\textbf{Accuracy}_{\mathbf{D}} = \dfrac{\sum_{x \in D} \sum_{c \in C} H_x^c}{max\left(\sum_{x \in D} \sum_{c \in C} \tilde{Y}_x^c, \sum_{x \in D} \sum_{c \in C} \tilde{P}_x^c \right)}$

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, categorizing images into semantic classes such as *beaches*, *sunsets* or *parties*. 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 [31]. 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 [31, 33, 32] as features.

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 [30] and Comaniciu and Meer [6] 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 [30, 6]. 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 7x7 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 result is a 49 x 2 x 3 = 294-dimension feature vector per image.

We use a Support Vector Machine (SVM) [2] as a classifier. The software we used is SVMFu [18]. SVM classifiers have been shown to give better performance than other classifiers like Learning Vector Quantization (LVQ) on similar problems [33, 32]. 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 [12]: 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, and then each of which 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.

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, 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 3 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 4 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 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

Table 2: 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.

	Criterion	Recall	Precision	Accuracy
Model-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
Model-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
Model-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**

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

augmenting the original training set with mirror images of each multilabel image. Mirroring an image in the horizontal direction (assuming correct orientation) does not change the classification of an image. We also add multilabel 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 multilabel images are correlated, this independence assumption is violated.

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

Table 4: α-Accuracy of *Model-s*, *Model-i* and *Model-x* for multi-label classification for original and mirror data sets

	Crit.	Original Set Accuracy (α-value)				Mirror Set Accuracy (α-value)			
		$\alpha = 0$	$\alpha = 1.0$	$\alpha = 2.0$	$\alpha = \infty$	$\alpha = 0$	$\alpha = 1.0$	$\alpha = 2.0$	$\alpha = \infty$
Model-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
Model-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
Model-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

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 multilabel 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 5 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.

Table 5: 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	
		$\alpha = 0$	$\alpha = 1$
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 Figure 6 are correctly labeled by the classifiers. Among the SVM scores for Figure 6(a), the scores corresponding to the two real classes are both positive and others are negative. For the image in Figure 6(b), all of the 6 SVM scores are negative:

-0.182365 -2.18762 -1.45516 -1.66521 -1.0902 -0.199863

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 disciminate some of the classes better (e.g., mountain vs. urban) [31].

In Figure 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 Figure 7(a),

-0.350572 -1.34971 -0.913437 -1.35506 -0.523688 -1.21277

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 Figure 7(b, c) were not detected.

In Figure 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 Figure 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 Figure 10, the image is labeled completely wrong, 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.

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 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.

References

[1] Matthew Boutell, Jiebo Luo, and Robert T. Gray. Sunset scene classification using simulated image recomposition. In *International Conference on Multimedia Expo (ICME)*, Baltimore, MD, July 2003.

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

[3] 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*, pages 339–342. Systems Engineering Association, June 1996.

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

[5] Amanda Clare and Ross D. King. Knowledge discovery in multi-label phenotype data. *Lecture Notes in Computer Science*, 2168:42–??, 2001.

[6] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.

[7] R. Duda, R. Hart, and D. Stork. *Pattern Classification*. John Wiley and Sons, Inc., New York, 2nd edition, 2001.

[8] 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.

[9] J. C. Gower and P. Legendre. Metric and euclidean properties of dissimilarity coefficients. *Journal of Classification*, 3:5–48, 1986.

[10] Q. Iqbal and J. Aggarwal. Retrieval by classification of images containing large manmade objects using perceptual grouping. *Pattern Recognition*, 35:1463–1479, 2001.

[11] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Machine Learning (ECML)*. Springer, 1998.

[12] Ulrich H.-G. Kreβel. *Advances in Kernel Methods: Support Vector Learning*, chapter 15, pages 255–268. MIT Press, Cambridge, MA, 1999.

[13] P. Lipson, E. Grimson, and P. Sinha. Configuration based scene classification and image indexing, 1997.

[14] Andrew McCallum. Multi-label text classification with a mixture model trained by em. In *AAAI'99 Workshop on Text Learning*, 1999.

[15] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[16] 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.

[17] S. Paek and S.-F. Chang. A knowledge engineering approach for image classification based on probabilistic reasoning systems. In *IEEE International Conference on Multimedia and Expo. (ICME-2000)*, volume II, pages 1133–1136, New York City, NY, Jul 30-Aug 2 2000.

[18] Ryan Rifkin. Svmfu. http://five-percent-nation.mit.edu/SvmFu, 2000.

[19] Gerard Salton. Developments in automatic text retrieval. *Science*, 253:974–980, 1991.

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

[21] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), 2002.

[22] Navid Serrano, Andreas Savakis, and Jiebo Luo. A computationally efficient approach to indoor/outdoor scene classification. In *International Conference on Pattern Recognition*, September 2002.

[23] 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.

[24] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, December 2000.

[25] J. R. Smith and C.-S. Li. Image classification and querying using composite region templates. *Computer Vision and Image Understanding*, 75(1/2):165 – 174, July/August 1999.

[26] Y. Song and A. Zhang. Analyzing scenery images by monotonic tree. *ACM Multimedia Systems Journal*, 2002.

[27] Martin Szummer and Rosalind W. Picard. Indoor-outdoor image classification. In *IEEE International Workshop on Content-based Access of Image and Video Databases*, Bombay, India, 1998.

[28] 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.

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

[30] Z. Tu and S.-C. Zhu. Image segmentation by data-driven markov chain monte carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5), May 2002.

[31] A. Vailaya, M. Figueiredo, A. Jain, and H.J. 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.

[32] 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(7):746–755, July 2002.

[33] Yongmei Wang and Hongjiang 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.

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

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

Figure 6: Some images whose prediction sets are completely right by using *Model-x* and C-criterion

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

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

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

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

Figure 7: Some images whose prediction sets are subsets of their real class sets

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

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

(c) real:MN, prediction:UN+MN+BH

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

Figure 8: Some images whose real class sets are subsets of their prediction sets

Figure 9: An image whose prediction set is partly right and partly wrong. Real:MN+FE, prediction:MN+BH

Figure 10: An image whose prediction set is completely wrong. Real:BH+MN, prediction:FD