# Reconfigurable Models for Scene Recognition

Sobhan Naderi Parizi
School of Engineering
Brown University
sobhan@brown.edu

John Oberlin
Dept. of Computer Science
Brown University
oberlin@cs.brown.edu

Pedro F. Felzenszwalb
School of Engineering and
Dept. of Computer Science
Brown University
pff@brown.edu

## Abstract

*We propose a new latent variable model for scene recognition. Our approach represents a scene as a collection of region models ("parts") arranged in a reconfigurable pattern. We partition an image into a pre-defined set of regions and use a latent variable to specify which region model is assigned to each image region. In our current implementation we use a bag of words representation to capture the appearance of an image region. The resulting method generalizes a spatial bag of words approach that relies on a fixed model for the bag of words in each image region.*

*Our models can be trained using both generative and discriminative methods. In the generative setting we use the Expectation-Maximization (EM) algorithm to estimate model parameters from a collection of images with category labels. In the discriminative setting we use a latent structural SVM (LSSVM). We note that LSSVMs can be very sensitive to initialization and demonstrate that generative training with EM provides a good initialization for discriminative training with LSSVM.*

## 1. Introduction

Consider an image of a beach scene. We expect to see sky, water and sand in the image. Moreover, we expect to see sky at the top of the image, water somewhere in the middle and sand in the bottom. One approach for capturing this information involves using a different bag of words model for different regions in the image. This structure can be modeled by spatial pyramid matching [5]. Note however that a region in the middle of the image could contain water or sand. Similarly a region at the top of the image could contain a cloud, the sun or blue sky alone. Therefore the features observed in each region depend on a latent variable specifying which of several possible region models should be used to capture the content of the region.

We propose to model a scene as a collection of region models ("parts") arranged in a reconfigurable pattern. An image is divided into a set of pre-defined regions and we have latent variables specifying which region model should be used for each image region. The model includes parameters so that each image region has a preference over the region models that can be assigned to it. In practice we divide an image into a grid of regions and use a bag of words (BoW) representation to capture the appearance of a region. We call the resulting models *Reconfigurable BoW* models.

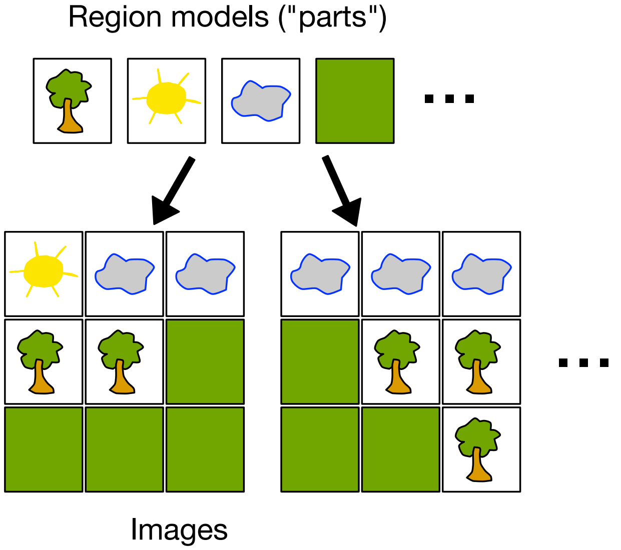

Figure 1. A Reconfigurable model for a class of outdoor scenes. We have $L$ region models that can be arranged in different ways to make up an image. Each image region has a preference over the region models that can be used to generate its content. In this example regions in the top are formed by choosing between a cloud or sun region model, while regions in the middle and bottom are formed by choosing between a tree or grass region model.

Figure 1 illustrates a model for a class of outdoor scenes composed of sky, grass and trees. We can think of the model as being defined by parts that model image regions with specific content. The latent variables specify which part should be used to capture the appearance of each region in the grid.

We compare reconfigurable BoW models to spatial BoW models that use a fixed model for the bag of words in each image region and show that reconfigurable models lead to superior results on two datasets. In particular we obtain results that are comparable to state-of-the-art methods on the MIT 67 indoor scene dataset [10], even though our implementation uses only simple local features.

The idea of modeling a scene in terms of a configuration of regions with specific properties goes back to the work in [6]. This notion has also been used recently for recognizing indoor scenes in [10] and [9]. These methods represent scenes using different kinds of deformable part models. Reconfigurable models are different from deformable models because they explicitly model the whole image. Reconfigurable models also allow the same part (region model) to be used multiple times in an image. For example a grass region model can be instantiated at multiple locations to explain a large patch of grass in an image.

Another kind of latent variable model that has been used for scene recognition involves hierarchical "topic" models [3, 2]. These models represent the features in an image using a mixture of topics associated with an image category. They are related to Reconfigurable models if we think of a region model as a topic. In the case of a reconfigurable model we assume there is a single topic in each image region. Here we train different region models for each image category but we could also share a set of region models over all categories as is often done with topic models.

The approach in [14] is closely related to ours from a technical point of view but they use only two region models for "foreground" and "background" regions while we use many different region models to model a scene.

We describe both generative and discriminative version of the reconfigurable BoW model. For the generative models we use Expectation-Maximization (EM) [1] to train model parameters. For the discriminative models we use a latent structural SVM (LSSVM) [15]. Discriminative training usually outperforms generative training but we have found that LSSVM training is much more sensitive to initialization when compared to EM training. We show that a combined approach that initializes LSSVM training using the results of EM training gives the best performance.

# 2. Generative and discriminative Models

## 2.1. Generative models

Let $x$ denote an image and $y$ denote an image class. Classification with generative models involves modeling a prior probability over classes, $p_\theta(y)$, and the probability of observing certain image features conditional on the image class $p_\theta(x|y)$. Using Bayes law we can classify an image $x$ by selecting the class $y$ with maximum probability given the observed image features

$$y^* = \operatorname*{argmax}_y p_\theta(y|x) = \operatorname*{argmax}_y p_\theta(y) p_\theta(x|y). \tag{1}$$

The parameters $\theta$ of a generative model can be estimated from a set of training examples $\{(x_1, y_1), \dots, (x_N, y_N)\}$ using a maximum likelihood criteria. Assuming the training examples are independent samples from $p_\theta(x, y)$ this leads to the following optimization problem

$$\theta^* = \operatorname*{argmax}_\theta \prod_{i=1}^{N} p_\theta(y_i) p_\theta(x_i|y_i). \tag{2}$$

One important aspect of generative models is that parameter estimation can often be decomposed into separate problems, one for each image class. Let $\theta = \{\gamma, \theta_1, \dots, \theta_M\}$ where $\gamma$ defines a discrete distribution over classes modeling $p_\theta(y)$ while $\theta_y$ defines the parameters of a generative model for the images in class $y$. That is, $p_\theta(x|y) = p_{\theta_y}(x)$. In this case maximum likelihood estimation amounts to selecting $\gamma$ based on the empirical frequencies of different classes in the training data and selecting $\theta_y$ to maximize $\prod_{i \text{ s.t. } y_i = y} p_{\theta_y}(x_i)$. Note that $\theta_y$ is estimated from training images of class $y$ alone.

In a latent variable model we have a set of unobservable values $z$ associated with each image $x$. We define $p_\theta(x|y)$ in terms of a distribution $p_\theta(z|y)$ over latent values conditional on the class of the image, and the probability of observing certain image features conditional both on the image class and the latent values $p_\theta(x|z, y)$. Then $p_\theta(x|y)$ is obtained by integrating over the latent variables

$$p_\theta(x|y) = \sum_z p_\theta(x|z, y) p_\theta(z|y). \tag{3}$$

Maximum likelihood parameter estimation with latent variable models typically leads to non-convex optimization problems. The Expectation-Maximization (EM) algorithm is a general tool for dealing with such problems.

## 2.2. Discriminative models

In contrast to the generative setting, the discriminative approach does not rely on explicit probabilistic models for the images in each class. Instead the parameters of a classifier are selected to directly minimize mistakes on the training data, often with a regularization bias to avoid overfitting. A common approach involves training a discriminant function $f_w(x, y)$ with high score if image $x$ is from class $y$, and low score otherwise. We then classify an image by selecting the class with highest score

$$y^* = \operatorname*{argmax}_y f_w(x, y). \tag{4}$$

Let $\{(x_1, y_1), \dots, (x_N, y_N)\}$ be a set of training examples. We would like to train $w$ such that $f_w(x_i, y_i) >$

$f_w(x_i, y)$ whenever $y \neq y_i$. A general max-margin approach involves an objective function

$$w^* = \underset{w}{\operatorname{argmin}} \frac{1}{2}||w||^2 + \\ C \sum_{i=1}^{N} \max_y (f_w(x_i, y) + L(y, y_i)) - f_w(x_i, y_i), \quad (5)$$

where $L(y, y') = 0$ if $y = y'$ and $L(y, y') = 1$ if $y \neq y'$.

This objective encourages the score of the correct class for each example to be above the highest score of an incorrect class plus one. Together with the regularization term, this leads to a large margin classifier.

An important class of discriminative models involves linear discriminant functions of a joint feature map

$$f_w(x, y) = w \cdot \Phi(x, y). \quad (6)$$

In this case the training problem defined by equation (5) corresponds to a structural SVM (SSVM) [11]. The resulting optimization problem is convex and can be solved using standard techniques.

We can define discriminative latent variable models using a discriminant function of the form

$$f_w(x, y) = \max_z w \cdot \Phi(x, y, z) \quad (7)$$

where $z$ is a set of latent values. In this case the training problem defined by equation (5) corresponds to a latent structural SVM (LSSVM) [15]. A popular example in computer vision is the deformable part model (DPM) for object detection described in [4]. The work in [4] considered the special case of a latent variable binary classifier (the object is present or not at each position in the image).

Unfortunately, the LSSVM training objective is non-convex. In [15], the training problem is solved using the CCCP algorithm [16], while [4] uses a coordinate descent method designed for the binary case. While these methods have been shown to work well in some applications there is increasing evidence that they can be quite sensitive to initialization. Our experiments confirm this is a significant problem for the models we consider. In contrast, the EM algorithm for generative models with latent variables seems to be less sensitive to initialization.

# 3. Basic models

Here we review two basic models for image classification. Our reconfigurable models build on these.

One difference between the discriminative approach we use and the methods that are most commonly used in the vision literature is that we use a single structural SVM for multi-class classification instead of several binary SVMs trained with a one-versus-all rule. We believe this is a more natural framework and it leads to a single discriminative training problem. In contrast, the one-versus-all approach involves one binary training problem per class where each problem involves training examples from all classes.

## 3.1. Bag of words (BoW)

A bag of words (BoW) model represents an image $x$ by an unordered collection of visual words. Suppose we have a dictionary with $K$ visual words. A bag of words $b$ is defined by a vector $[b_1, \ldots, b_K]$ where $b_k$ is the multiplicity of word $k$ in $b$. We use $|b|$ to denote the total number of words in $b$.

**Generative model** We can define a generative BoW classifier by assuming the visual words in an image come from a multinomial distribution associated with the image class.

A multinomial distribution is defined by a discrete distribution with parameters $v = \{v_1, \ldots, v_K\}$ specifying the probability of each outcome in a trial. In the multinomial model each word in a bag is generated independently. The probability of a bag $b$ (conditional on $|b|$) is given by

$$\text{mult}(b, v) = \frac{|b|!}{b_1! \cdots b_K!} \prod_{k=1}^{K} v_k^{b_k}. \quad (8)$$

To define a BoW classifier, let $\theta_y$ specify a discrete distribution over visual words associated with class $y$. Then

$$p_\theta(x|y) = \text{mult}(x, \theta_y). \quad (9)$$

We can estimate the model parameters $\theta$ from a set of training examples using a maximum likelihood criteria. The parameters $\theta_y$ simply depend on the frequencies of different visual words observed in images from class $y$.

To classify an image we combine the generative model associated with each class $p_\theta(x|y)$ with the prior probability of each class $p_\theta(y)$ as specified by equation (1).

**Discriminative model** We can define a discriminative BoW classifier using a discriminant function of the form

$$f_w(x, y) = w_y \cdot \phi(x). \quad (10)$$

Here $w = [w_1; \ldots; w_M]$ denotes a vector of model parameters where $w_y$ are parameters associated with class $y$. The function $\phi(x)$ is a (possibly non-linear) feature map of the bag of words in image $x$.[1]

We can train $w$ using equation (5). Since $f_w(x, y)$ is linear in $w$ this leads to a structural SVM. As mentioned above the optimization problem defined by structural SVMs is convex and can be solved using a variety of techniques.

[1] $\phi(x)$ should generally include a dimension with constant value to allow for a class specific bias term in $w_y$.

### 3.2. Spatial bag of words (SBoW)

Following [5] we can take spatial information into account by using a different model for the features in different regions of an image. This leads to *Spatial BoW (SBoW)* models. Here we consider the case where the image is partitioned into a fixed grid with $R$ regions. We use $r$ to denote an image region and $x_r$ to denote the bag of words in region $r$ of an image $x$.

**Generative model** In the SBoW model we capture spatial information by allowing the probability of observing a particular visual word to depend on the region where the word is observed. Let $\theta_{y,r}$ denote a discrete distribution over visual words associated with region $r$ and class $y$. Under the SBoW model we have

$$p_\theta(x|y) = \prod_{r=1}^{R} \text{mult}(x_r, \theta_{y,r}). \tag{11}$$

As in the case of a BoW model we can estimate the model parameters from a set of training examples using a maximum likelihood criteria. The parameters $\theta_{y,r}$ simply depend on the frequencies of different visual words observed in region $r$ taken over images in class $y$.

**Discriminative model** We can define a discriminative SBoW classifier using a discriminant function of the form

$$f_w(x, y) = w_y \cdot [\phi(x_1); \cdots ; \phi(x_R)]. \tag{12}$$

As in the discriminative BoW model, $w_y$ are parameters associated with class $y$, but now $w_y$ has different parameters for modeling the visual words in each image region. Since $f_w$ is still linear in $w$, we can once again train $w$ using a structural SVM.

## 4. Reconfigurable bag of words (RBoW)

Our latent variable model builds on the SBoW model. We can think of an SBoW model as a part-based approach with one part per image region. Here we augment the SBoW model to allow for reconfiguration of the parts that make up a scene. This leads to a class of *reconfigurable bag of words (RBoW)* models.

For example, an RBoW model for beach scenes might have a part modeling an image region that contains the sun and another part modeling an image region that contains a cloud. The regions at the top of a beach image could all contain the sun or a cloud. In the RBoW model we have a latent variable indicating which region model (part) should be used to explain each image region.

As in an SBoW model we assume images are partitioned into $R$ predefined regions and $x_r$ specifies the bag of words observed in region $r$ within the image $x$. In a reconfigurable BoW model we have $L$ BoW region models. A latent value $z_r$ assigns a particular region model to region $r$.

### 4.1. Generative RBoW model

In the RBoW model we assume the visual words in image region $r$ are generated independently conditional on the class label and latent value $z_r$ assigning a region model to region $r$.

Let $W_{y,l}$ be a discrete distribution over visual words associated with the $l$-th region model for class $y$. Under the RBoW model we have

$$p_\theta(x|z, y) = \prod_{r=1}^{R} \text{mult}(x_r, W_{y,z_r}) \tag{13}$$

We assume the latent values $z_r$ are independent conditional on the class label $y$ but not identically distributed. There is a different categorical distribution capturing which parts are likely to occur in each region of an image from a particular class. For each class $y$ and region $r$ let $a_{y,r} = \{a_{y,r,1}, \ldots a_{y,r,L}\}$ where $a_{y,r,l}$ is the probability that $z_r = l$ on an image from class $y$. This leads to the following distribution over the latent values

$$p_\theta(z|y) = \prod_{r=1}^{R} a_{y,r,z_r}. \tag{14}$$

Now we can express the probability of observing the features in an image $x$ conditional on an image class $y$ by integrating over the possible latent values

$$p_\theta(x|y) = \sum_{z} p_\theta(x|z, y) p_\theta(z|y). \tag{15}$$

Since the latent values are independent and the observations are independent conditional on the latent values we can compute this probability efficiently (in $O(RLK)$ time for a model with $L$ region models on an image with $R$ regions and a dictionary with $K$ visual words) as

$$p_\theta(x|y) = \prod_{r=1}^{R} \sum_{z_r} \text{mult}(x_r, W_{y,z_r}) a_{y,r,z_r}. \tag{16}$$

The parameters $\theta_y$ associated with the model for class $y$ are given by $L$ BoW region models $\{W_{y,1}, \ldots W_{y,L}\}$ and $R$ distributions over region models $\{a_{y,1}, \ldots, a_{y,R}\}$.

**Parameter estimation with EM** Suppose we have $N$ training examples $\{(x_1, y_1), \ldots, (x_N, y_N)\}$. We can estimate the parameters of an RBoW model using a maximum likelihood criteria, but since the model has latent variables maximum likelihood estimation leads to a non-convex optimization problem. We use the Expectation-Maximization (EM) algorithm to address this problem [1].

EM computes a sequence of model parameters by repeatedly alternating between two steps which are guaranteed to increase the likelihood of the data. In the E step we use the current model $\theta$ to compute the posterior probability of the latent variables in each training example. This gives us a tractable lower-bound on the likelihood function which is tangent to the actual likelihood at the current $\theta$. In the M step we update the model parameters by maximizing the lower-bound on the likelihood function. In the case of an RBoW model we obtain the following algorithm.

**Repeat until convergence**

**Step 1 (E):** For each example $i$, region $r$ and latent value $z_r$ compute $Q_{i,r,z_r} = p_\theta(z_r|x_i, y_i)$ using

$$p_\theta(z_r|x_i, y_i) = \frac{\text{mult}(x_r, W_{y,z_r})a_{y,r,z_r}}{\sum_l \text{mult}(x_r, W_{y,l})a_{y,r,l}} \tag{17}$$

**Step 2 (M):** Update $\theta$ by selecting

$$a_{y,r,l} \propto \sum_{i,y_i=y} Q_{i,r,l} \tag{18}$$

$$W_{y,l,k} \propto \sum_{i,y_i=y} \sum_r Q_{i,r,l}c_{i,r,k} \tag{19}$$

where $c_{i,r,k}$ is the number of times the $k$-th visual word was seen in region $r$ of $x_i$ and the parameters are normalized so that $\sum_l a_{y,r,l} = 1$ and $\sum_k W_{y,l,k} = 1$.

We initialize the algorithm by selecting a random latent value $z_r$ for each region $r$ within $x_i$ and setting $Q_{i,r,z_r} = 1$ while $Q_{i,r,l} = 0$ for $l \neq z_r$.

In practice we smooth the multinomial probabilities in equation (17) by raising them to a power of $1/T$. This attenuates the sharpness induced by the assumption that visual words are generated independently within a region. This is essential when using densely sampled features.

## 4.2. Discriminative RBoW model

We can define a discriminative RBoW classifier using a discriminant function of the form

$$f_w(x, y) = \max_z \sum_r A_{y,r,z_r} + B_{y,z_r} \cdot \phi(x_r). \tag{20}$$

Here $\phi(x_r)$ is a feature map. The vector $B_{y,l}$ specifies model parameters for the $l$-th region model in class $y$. The parameter $A_{y,r,l}$ specifies a score for assigning part $l$ to region $r$ in an image of class $y$.

The intuition is that for each class $y$ we attempt to explain the image $x$ by finding the best assignment $z$ of region models to the regions in $x$. An assignment $z$ has a score with two terms. The first term $\sum_r A_{y,r,z_r}$ gives preference to some region models over others for each image region. The second term $\sum_r B_{y,z_r} \cdot \phi(x_r)$ measures how well the region model $z_r$ matches the content of region $r$.

**Latent structural SVM** The score of class $y$ under a particular assignment $z$ can be expressed as

$$\sum_r A_{y,r,z_r} + B_{y,z_r} \cdot \phi(x_r) = w_y \cdot \Phi(x, z) \tag{21}$$

Here the weight vector $w_y$ is the concatenation of the parameters $A_{y,r,z_r}$ and $B_{y,z_r}$.

The vector $\Phi(x, z)$ is a sum of $R$ vectors $\psi(x, r, z_r)$, one per image region. The vector $\psi(x, r, z_r)$ equals $\phi(x_r)$ in the dimensions corresponding to $B_{y,z_r}$ within $w_y$ and 1 in the dimension corresponding to $A_{y,r,z_r}$ within $w_y$. The other entries in $\psi(x, r, z_r)$ are zero.

Let $w = [w_1; \ldots; w_M]$ denote a vector with all model parameters from all classes. Suppose we have $N$ training examples $\{(x_1, y_1), \ldots, (x_N, y_N)\}$. We can train $w$ using a latent structural SVM (LSSVM) [15]

$$w^* = \operatorname*{argmin}_w \frac{1}{2}||w||^2 + \\ C\sum_{i=1}^{N} \max_{y,z}(w_y \cdot \Phi(x_i, z) + L(y, y_i)) - \max_z w_{y_i} \cdot \Phi(x_i, z) \tag{22}$$

where $L(y, y') = 0$ if $y = y'$ and $L(y, y') = 1$ if $y \neq y'$.

Like the objective function of a structural SVM a latent structural SVM encourages the score of the correct class to be above the highest score of an incorrect class by a margin of one. The only difference is that the score is no longer linear, and instead involves a maximization over $z$.

Unfortunately the optimization problem defined by a LSSVM is not convex. Following [15] we use the CCCP [16] algorithm to find a local optimum solution. CCCP works by repeatedly alternating between two steps. The first step picks the best latent values for each training example under the current model. The second step defines a convex objective function over model parameters by replacing the maximization in the last term of the LSSVM objective with the latent values from the first step. This convex objective gives an upper-bound on the LLSVM objective function.

**Repeat until convergence**

**Step 1:** For each training example $i$ compute

$$z_i = \operatorname*{argmax}_z w_{y_i} \cdot \Phi(x_i, z) \tag{23}$$

**Step 2:** Update $w$ by optimizing the convex objective

$$w^* = \operatorname*{argmin}_w \frac{1}{2}||w||^2 + \\ C\sum_{i=1}^{N} \max_{y,z}(w_y \cdot \Phi(x_i, z) + L(y, y_i)) - w_{y_i} \cdot \Phi(x_i, z_i) \tag{24}$$

In practice we use stochastic subgradient descent to optimize the convex function in step 2.

Note that CCCP optimization for LSSVM is similar to EM in the way that it alternates estimating latent values and estimating model parameters. One important difference is that in step 1 of EM we obtain a distribution over latent values for each example while here we pick a single latent value for each example. This seems to make LSSVM optimization with CCCP much more sensitive to initialization. In practice we find that most latent values that are selected in step 1 of the initial iteration never change.

The optimization requires either an initial weight vector $w$ or initial latent values $z_i$, in which case training starts in step 2. We experimented with three different methods for selecting initial latent values. One method simply picks a random region model for each region in each image. Another method picks a particular region model for each region. In particular, we train models with 16 regions and 16 region models and assign a different initial region model for each image region. Finally, we tried using the result of EM training of a generative RBoW model to select the initial latent values. In this case we set the initial $z_i$ to be the most probable latent values under the model trained by EM.

## 5. Experiments

We evaluated our model on the *15 Scene* dataset from [5] and the *MIT 67 Indoor Scenes* dataset from [10]. We measured the performance of different models using the average of the diagonal entries of their confusion matrix.

We used densely sampled SIFT features [7] to define visual words. The visual vocabulary is created using *K-Means* clustering on a subset of SIFT features randomly sampled from training images. We set the size of the visual vocabulary to be $K = 200$ in all of our experiments.

For discriminative training we used a feature map $\phi(b)$ that normalizes the bag of words vector $b$ to have unit norm and then computes the square root of each entry.

All of the experiments with SBoW and RBoW models used a 4x4 regular grid to partition the image into $R = 16$ rectangular regions. For the RBoW models we used $L = 16$ region models for each image category. Taking $L = R$ makes it possible to initialize an RBoW model with a fixed assignment of region models to image regions, with one region model for each image region.

### 5.1. MIT 67 Indoor Scenes

The MIT dataset contains images from 67 different categories of indoor scenes. There is a fixed training and test set containing approximately 80 and 20 images from each category respectively.

| **MIT 67 Indoor Scenes** | **Method** | **Rate** | **LSSVM Objec.** |
|---|---|---|---|
| **Prev. Works** | ROI+Gist [10] | 26.5 | |
| | MM-scene [17] | 28.0 | |
| | CENTRIST [12] | 36.9 | |
| | Object Bank [13] | 37.6 | |
| | DPM [9] | 30.4 | |
| | DPM+Gist-color+SP [9] | 43.1 | |
| **BoW** | Generative | 12.80 | |
| | Discriminative | 25.17 | |
| **SBoW** | Generative | 19.46 | |
| | Discriminative | 33.99 | |
| **RBoW** | Generative | 27.66 | |
| | Discriminative Init-rand | 31.63 | 91.08 |
| | Discriminative Init-fixed | 34.99 | 83.50 |
| | Discriminative Init-EM | 37.93 | 80.30 |

Table 1. Average performance of different methods on the MIT dataset. The last column shows the final value of the LSSVM objective function for RBoW models with different initializations.

Table 1 summarizes the performance of our models and some previously published methods. To our knowledge, the state-of-the-art results on this dataset were obtained in [9] by combining scores from a deformable part model (DPM) [4] for each category, together with spatial pyramid matching [5] and color GIST descriptors [8]. To compare reconfigurable models to deformable models we also include the performance obtained in [9] using DPMs alone. Table 1 also shows the performance of our BoW and SBoW baselines. The performance gap between the BoW and SBoW approaches proves a considerable point regarding the importance of spatial information for image classification.

Table 1 includes the results of discriminative RBoW models trained with different initialization methods. As discussed in Section 4.2, CCCP requires initial latent values for each training example. *Init-rand* selects random initial region models for each image region. *Init-fixed* selects a fixed initial region model for each image region. *Init-EM* uses a generative RBoW model trained with EM, and selects the most probable latent values under the generative model to initialize LSSVM training. We have found that Init-EM gives consistently better results. This shows the importance of initialization for LSSVM training. It also shows that while generative models typically don't perform as well as discriminative models, EM training seems to be less susceptible to local optima when compared to LSSVM.

Last column of Table 1 shows the final value of LSSVM objective under each initialization method. Note that the value of the objective is consistent with the performance of the model, suggesting that developing better optimization algorithms for LSSVM should lead to better models.

Table 2 shows per-category performance of the discriminative RBoW model (initialized with EM), the discriminative baseline approaches and the DPM method from [9]. Note that even though SBoW has a better overall accuracy than BoW, it does worse in 12 classes. RBoW is able to

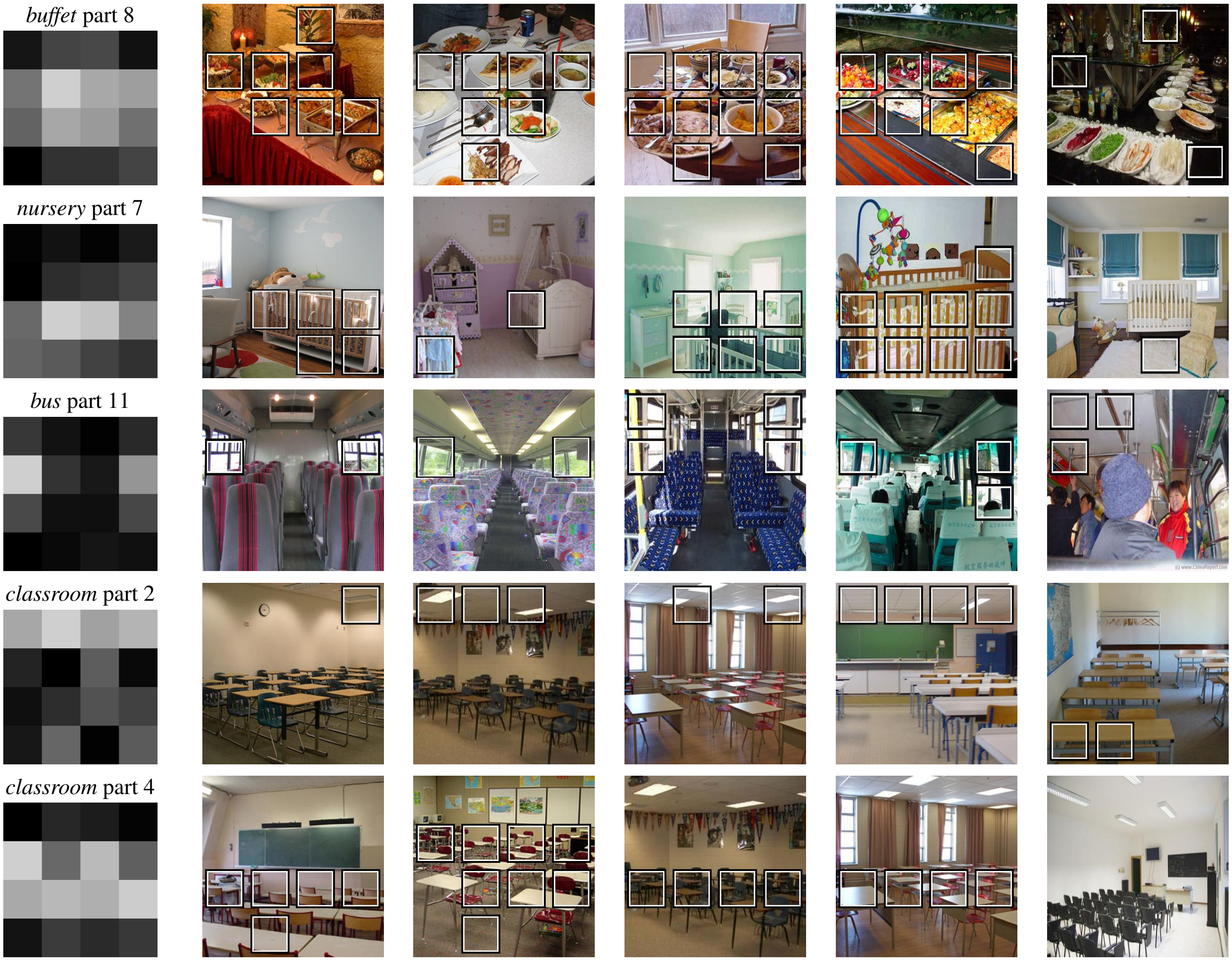

Figure 2. Some interesting region models learned for different categories using a discriminative RBoW model (Init-EM). Each row illustrates a region model for a particular category. The first column shows the preferences of different image regions for this region model ($A_{y,r,l}$ for fixed $y$ and $l$). The other columns show image regions that were assigned to this region model during classification ($z_r = l$).

recover the performance lost by SBoW in several classes, including *florist*, *gameroom* and *videostore*. The RBoW model performs significantly better than our baselines and the DPM method on several classes.

Figure 2 illustrates some interesting region models that were learned for different categories. For example, in the *buffet* class there is a model for food regions, in the *nursery* class there is a model for crib regions, while in the *classroom* class there is a model for regions with desks and another for the ceiling.

Training RBoW models is reasonably fast. Training a generative RBoW model with EM (with 16 parts and 16 image regions) on the MIT dataset takes about 10 minutes on a 2.8GHz computer with an i7 multi-core processor. Training a similar discriminative model with LSSVM on the MIT dataset takes about 10 hours. Discriminative training takes much longer than EM because step 2 of CCCP involves a large convex optimization problem. At test time our implementation can classify more than 180 images per second for the MIT dataset. The running time for classification scales linearly with the number of classes.

### 5.2. 15 Scene Dataset

The 15 Scene dataset contains 4485 images of 15 different scenes. It includes both indoor scenes (*office*, *bedroom*, *kitchen*, *living-room*, *store*) and outdoor scenes (*suburb*, *coast*, *forest*, *highway*, *inside-city*, *mountain*, *open-country*, *street*, *tall-building*, *industrial*). The dataset does not provide separate training and test sets, so we use 5 random splits and compute the mean and standard deviation of the classification performance across splits. In each split we use 100 training images for each category.

| Category | RBoW | SBoW | BoW | DPM | Category | RBoW | SBoW | BoW | DPM | Category | RBoW | SBoW | BoW | DPM | Category | RBoW | SBoW | BoW | DPM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| bowling | 85 | 85 | 55 | 35 | dentaloffice | 48 | 29 | 19 | 24 | laundromat | 36 | 41 | 9 | 45 | airport ins. | 20 | 15 | 15 | 5 |
| florist | 84 | 63 | 74 | 79 | casino | 47 | 47 | 63 | 32 | stairscase | 35 | 35 | 45 | 35 | bedroom | 19 | 14 | 0 | 5 |
| ins. subway | 81 | 62 | 57 | 62 | gameroom | 45 | 10 | 40 | 40 | bathroom | 33 | 22 | 6 | 50 | hairsalon | 19 | 24 | 5 | 43 |
| cloister | 80 | 85 | 55 | 90 | prisoncell | 45 | 40 | 35 | 40 | grocerystore | 33 | 29 | 38 | 19 | locker room | 19 | 14 | 14 | 19 |
| inside bus | 78 | 61 | 9 | 43 | trainstation | 45 | 70 | 35 | 35 | subway | 33 | 33 | 14 | 38 | warehouse | 19 | 19 | 10 | 24 |
| greenhouse | 75 | 80 | 80 | 65 | auditorium | 44 | 39 | 22 | 11 | bookstore | 30 | 20 | 30 | 45 | artstudio | 15 | 5 | 0 | 5 |
| church ins. | 74 | 79 | 53 | 63 | bar | 44 | 39 | 33 | 11 | winecellar | 29 | 29 | 29 | 14 | toystore | 14 | 5 | 0 | 9 |
| classroom | 72 | 56 | 44 | 67 | clothingstore | 44 | 33 | 11 | 33 | child. room | 28 | 28 | 11 | 6 | lobby | 10 | 10 | 5 | 30 |
| buffet | 65 | 60 | 55 | 75 | garage | 44 | 39 | 39 | 56 | dining room | 28 | 22 | 11 | 28 | poolinside | 10 | 5 | 5 | 0 |
| concert hall | 65 | 60 | 55 | 65 | corridor | 43 | 62 | 33 | 57 | gym | 28 | 6 | 0 | 22 | restaurant | 10 | 10 | 10 | 5 |
| elevator | 62 | 62 | 57 | 52 | meetingroom | 41 | 55 | 27 | 75 | lab. wet | 27 | 18 | 0 | 5 | office | 10 | 10 | 10 | 10 |
| closet | 61 | 56 | 56 | 44 | videostore | 41 | 18 | 23 | 18 | rstrnt kitchen | 26 | 26 | 0 | 4 | bakery | 5 | 5 | 0 | 11 |
| comp. room | 56 | 33 | 6 | 22 | hospitalroom | 40 | 30 | 5 | 5 | mall | 25 | 15 | 10 | 25 | operat. room | 5 | 0 | 32 | 5 |
| movietheater | 55 | 65 | 50 | 45 | kindergarden | 40 | 40 | 25 | 15 | waitingroom | 24 | 14 | 5 | 5 | livingroom | 5 | 5 | 10 | 20 |
| nursery | 55 | 50 | 75 | 60 | museum | 39 | 26 | 0 | 13 | fastfoodrstrnt | 24 | 6 | 35 | 12 | deli | 0 | 0 | 5 | 5 |
| pantry | 55 | 50 | 30 | 75 | kitchen | 38 | 43 | 14 | 29 | tv studio | 22 | 44 | 17 | 6 | jewel. shop | 0 | 9 | 0 | 5 |
| library | 50 | 45 | 40 | 0 | studiomusic | 37 | 37 | 11 | 32 | shoeshop | 21 | 32 | 21 | 16 | | | | | |

Table 2. Performance of our reconfigurable model in comparison to the baseline methods on the MIT dataset. The last column shows performance of DPM method from [9].

| **15 Scenes** | **BoW** | **SBoW** | **RBoW** | | |
|---|---|---|---|---|---|
| | | | **Init-rand** | **Init-fixed** | **Init-EM** |
| **Disc.** | 71.7 ± 0.2 | 77.7 ± 0.9 | 74.5 ± 0.4 | 78.5 ± 1.1 | 78.6 ± 0.7 |
| **Gen.** | 62.1 ± 2.4 | 74.3 ± 0.5 | 76.1 ± 0.5 | | |

Table 3. Average performance of different methods on the 15 scene dataset. We used three different initialization methods for training a discriminative RBoW model.

Table 3 compares the overall performance of RBoW to the SBoW and BoW baselines. Again we see that careful initialization is important for LSSVM training. Initialization of CCCP using a generative model trained with EM leads to the best performance, while random initialization leads to the worst performance.

## 6. Summary

Reconfigurable models represent images by a collection of regions with specific content. For each scene category we have a set of region models. The content of an image is defined by latent variables that assign a region model to each image region. The models defined here assume the latent variables are independent conditional on the image category. In the future we plan to model dependencies between the latent variables. For example in an outdoor scene we should have at most one region with a sun, and regions with water should never be above regions with sky. Our current models rely on a pre-defined partition of an image into a grid of regions. We would like to relax this assumption so that we can better capture the content of an image.

Latent variable models lead to challenging training problems, especially in the discriminative setting. Our experiments demonstrate that EM can be used as an effective method for initializing LSSVM training.

## References

[1] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. 2, 4

[2] A. Bosch, A. Zisserman, and X. Munoz. Scene classification using a hybrid generative/discriminative approach. *TPAMI*, 2008. 2

[3] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005. 2

[4] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 2010. 3, 6

[5] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 1, 4, 6

[6] P. Lipson, W. Grimson, and P. Sinha. Configuration based scene classification and image indexing. In *CVPR*, 1997. 2

[7] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 6

[8] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV*, 2001. 6

[9] M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *ICCV*, 2011. 2, 6, 8

[10] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *CVPR*, 2009. 2, 6

[11] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 2006. 3

[12] J. Wu and J. Rehg. Centrist: A visual descriptor for scene categorization. *TPAMI*, 2010. 6

[13] E. Xing, L. Li, H. Su, and L. Fei-Fei. Object bank: A high level image representation for scene classification and semantic feature sparsification. In *NIPS*, 2010. 6

[14] O. Yakhnenko, J. Verbeek, and C. Schmid. Region-based image classification with a latent svm model. *INRIA Tech. Rep. 7665*, 2011. 2

[15] C.-N. J. Yu and T. Joachims. Learning structural svms with latent variables. In *ICML*, 2009. 2, 3, 5

[16] A. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 2003. 3, 5

[17] J. Zhu, L. Li, L. Fei-Fei, and E. Xing. Large margin training of upstream scene understanding models. In *NIPS*, 2010. 6