A Bayesian Hierarchical Model for Learning Natural Scene Categories

Li Fei-Fei
California Institute of Technology
Electrical Engineering Dept.
Pasadena, CA 91125, USA
feifeili@vision.caltech.edu

Pietro Perona
California Institute of Technology
Electrical Engineering Dept.
Pasadena, CA 91125, USA
perona@vision.caltech.edu

Abstract

We propose a novel approach to learn and recognize natural scene categories. Unlike previous work [9, 17], it does not require experts to annotate the training set. We represent the image of a scene by a collection of local regions, denoted as codewords obtained by unsupervised learning. Each region is represented as part of a "theme". In previous work, such themes were learnt from hand-annotations of experts, while our method learns the theme distributions as well as the codewords distribution over the themes without supervision. We report satisfactory categorization performances on a large set of 13 categories of complex scenes.

1. Introduction

The ability to analyze and classify accurately and rapidly the scene in which we find ourselves is highly useful in everyday life. Thorpe and colleagues found that humans are able to categorize complex natural scenes containing animals or vehicles very quickly [12]. Li and colleagues later showed that little or no attention is needed for such rapid natural scene categorization [6]. Both of these studies posed a serious challenge to the conventional view that to understand the context of a complex scene, one needs first to recognize the objects and then in turn recognize the category of the scene [14].

Can we recognize the context of a scene without having first recognized the objects that are present? A number of recent studies have presented approaches to classify indoor versus outdoor, city versus landscape, sunset versus mountain versus forest using global cues (e.g. power spectrum, color histogram information) [3, 11, 15]. Oliva and Torralba further incorporated the idea of using global frequency with local spatial constraints [9]. The key idea is to use intermediate representations before classifying scenes: scenes are first labelled with respect to local and global properties by human observers. Similarly, Vogel and Schiele also used an intermediate representation obtained from human observers in learning the semantic context of a scene [17].

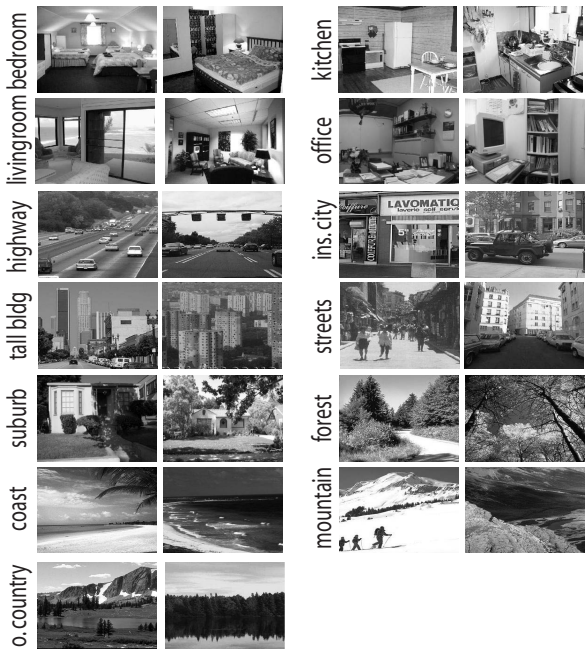


Figure 1. Our dataset consists of 13 categories, the largest natural scene category dataset to date. Detailed description of the dataset is in Section 3.

A main requirement of such approaches is the manual annotation of "intermediate" properties. In [9], human subjects are instructed to rank each of the hundreds of training scenes into 6 different properties (e.g. ruggedness, expansiveness, roughness, etc). In [17], human subjects are asked to classify $59,582$ local patches from the training images into one of 9 different "semantic concepts" (e.g. water, foliage, sky, etc.). Both cases involve tens of hours of manual labor. These works clearly point to the usefulness of these intermediate representations and motivate us to think of methods for learning such representations directly from the data: both because hand-annotating images is tedious

and expensive, and because expert-defined labels are somewhat arbitrary and possibly sub-optimal.

Much can also be learnt from studies for classifying different textures and materials [10, 5, 16]. Traditional texture models first identify a large dictionary of useful textons (or codewords). Then for each category of texture, a model is learnt to capture the signature distribution of these textons. We could loosely think of a texture as one particular intermediate representation of a complex scene. Again, such methods yield a model for this representation through manually segmented training examples. Another limitation of the traditional texture model is the hard assignment of one distribution for a class. This is fine if the underlying images are genuinely created by a single mixture of textons. But this is hardly the case in complex scenes. For example, it is not critical at all that trees must occupy 30% of a suburb scene and houses 60%. In fact, one would like to recognize a suburb scene whether there are many trees or just a few.

The key insights of previous work, therefore, appear to be that using intermediate representations improves performance, and that these intermediate representations might be thought of as textures, in turn composed of mixtures of textons, or codewords. Our goal is to take advantage of these insights, but avoid using manually labeled or segmented images to train the system, if possible at all. To this end, we adapt to the problems of image analysis recent work by Blei and colleagues [1], which was designed to represent and learn document models. In this framework, local regions are first clustered into different intermediate themes, and then into categories. Probability distributions of the local regions as well as the intermediate themes are both learnt in an automatic way, bypassing any human annotation. No supervision is needed apart from a single category label to the training image. We summarize our contribution as follows.

- Our algorithm provides a principled approach to learning relevant intermediate representations of scenes automatically and without supervision.
- Our algorithm is a principled probabilistic framework for learning models of textures via codewords (or textons) [5, 16, 10]. These approaches, which use histogram models of textons, are a special case of our algorithm. Given the flexibility and hierarchy of our model, such approaches can be easily generalized and extended using our framework.
- Our model is able to group categories of images into a sensible hierarchy, similar to what humans would do.

We introduce the generative Bayesian hierarchical model for scene categories in Section 2. Section 3 describes our dataset of 13 different categories of scenes and the experimental setup. Section 4 illustrates the experimental results. We discuss in Section 5 our results and future directions.

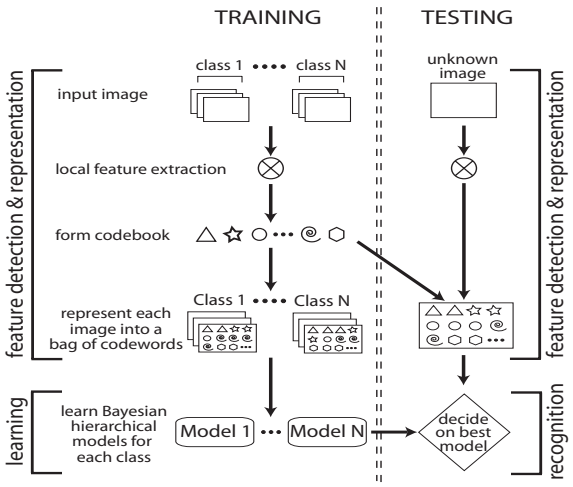


Figure 2. Flow chart of the algorithm.

2. Our Approach

Fig.2 is a summary of our algorithm in both learning and recognition. We model an image as a collection of local patches. Each patch is represented by a codeword from a large vocabulary of codewords (Fig.4). The goal of learning is to achieve a model that best represents the distribution of these codewords in each category of scenes. In recognition, therefore, we first identify all the codewords in the unknown image. Then we find the category model that fits best the distribution of the codewords of the particular image.

Our algorithm is modified based on the *Latent Dirichlet Allocation* (LDA) model proposed by Blei et al. [1]. We differ from their model by explicitly introducing a category variable for classification. Furthermore, we propose two variants of the hierarchical model (Fig.3(a) and (b)).

2.1 Model Structure

It is easier to understand the model (Fig.3(a)) by going through the generative process for creating a scene in a specific category. To put the process in plain English, we begin by first choosing a category label, say a mountain scene. Given the mountain class, we draw a probability vector that will determine what intermediate theme(s) to select while generating each patch of the scene. Now for creating each patch in the image, we first determine a particular theme out of the mixture of possible themes. For example, if a "rock" theme is selected, this will in turn privilege some codewords that occur more frequently in rocks (e.g. slanted lines). Now the theme favoring more horizontal edges is chosen, one can draw a codeword, which is likely to be a horizontal line segment. We repeat the process of drawing both the theme and codeword many times, eventually forming an entire bag of patches that would construct a scene of mountains. Fig.3(a) is a graphical illustration of the generative model. We will call this model the Theme Model 1.

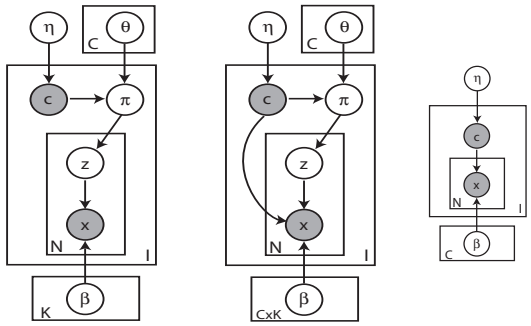


Figure 3. **(a)** Theme Model 1 for scene categorization that shares both the intermediate level themes as well as feature level codewords. **(b)** Theme Model 2 for scene categorization that shares only the feature level codewords; **(c)** Traditional texton model [5, 16].

Fig.3(b) is a slight variation of the model in Fig.3(a). We call it the Theme Model 2. Unless otherwise specified, the rest of the paper will focus on Theme Model 1. Now we are ready to show the mathematical details of the formulation of this model and how we learn its parameters.

2.1.1 The Theme Models

We begin with some notations and definitions for the Theme Model 1 in Fig.3(a). We will contrast explicitly the use of terminology with both [1] and the texture studies [5, 16].

- A *patch* x is the basic unit of an image, defined to be a patch membership from a dictionary of codewords indexed by $\{1, \ldots, T\}$. The t^{th} codeword in the dictionary is represented by a T-vector x such that $x^t = 1$ and $x^v = 0$ for $v \neq t$. In Fig.3(a), x is shaded by common convention to indicate that it is an observed variable. Nodes in the graph that are unobserved have no shading. The equivalent of an image in [1] is a "document". And a codeword (or patch) in our model is a "word" in [1]. In texture and material literature, a codeword is also referred as a "texton" [5, 16].
- An *image* is a sequence of N patches denoted by $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$, where x_n is the n^{th} patch of the image.
- A *category* is a collection of I images denoted by $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_I\}$. In [1], this is equivalent to a "corpus".

We can now write down the process that generates an image i formally from the model.

1. Choose a category label $c \sim p(c|\boldsymbol{\eta})$ for each image, where $c = \{1, \ldots, C\}$. C is the total number of categories. $\boldsymbol{\eta}$ is a C-dimensional vector of a multinomial distribution;
2. Now for this particular image in category c, we want to draw a parameter that determines the distribution of the intermediate themes (e.g. how "foliage", "water", "sky" etc. are distributed for this scene). This is done by choosing $\boldsymbol{\pi} \sim p(\boldsymbol{\pi}|c, \boldsymbol{\theta})$ for each image. $\boldsymbol{\pi}$ is the parameter of a multinomial distribution for choosing the themes. $\boldsymbol{\theta}$ is a matrix of size $C \times K$, where $\boldsymbol{\theta}_{c\cdot}$ is the K-dimensional Dirichlet parameter conditioned on the category c. K is the total number of themes.
3. for each N patches x_n in the image
   - Choose a theme $z_n \sim \text{Mult}(\boldsymbol{\pi})$. z_n is a K-dim unit vector. $z_n^k = 1$ indicates that the k^{th} theme is selected (e.g. "rock" theme).
   - Choose a patch $x_n \sim p(x_n|z_n, \boldsymbol{\beta})$, where $\boldsymbol{\beta}$ is a matrix of size $K \times T$. K is again the number of themes and T is the total number of codewords in the codebook. Therefore we have $\boldsymbol{\beta}_{kt} = p(x_n^t = 1|z_n^k = 1)$.

A $K-$dimensional Dirichlet random variable $\boldsymbol{\pi}$ has the property such that $\pi_i \geq 0, \sum_{i=1}^K \pi_i = 1$. It is a conjugate distribution of a multinomial distribution. Since the themes $\boldsymbol{z}$ are best described as a discrete variable over the multinomial distribution, Dirichlet distribution becomes the natural choice to describe distribution of $\boldsymbol{\pi}$ [2]. It has the following probability density:

$$Dir(\boldsymbol{\pi}|\boldsymbol{\theta}_{c\cdot}) = \frac{\Gamma\left(\sum_{i=1}^K \theta_{ci}\right)}{\prod_{i=1}^K \Gamma(\theta_{ci})} \pi_i^{(\theta_{ci}-1)} \ldots \pi_K^{(\theta_{cK}-1)} \quad (1)$$

Given the parameters $\boldsymbol{\theta}$, $\boldsymbol{\eta}$ and $\boldsymbol{\beta}$, we can now write the full generative equation of the model. It is the joint probability of a theme mixture $\boldsymbol{\pi}$, a set of N themes $\boldsymbol{z}$, a set of N patches $\boldsymbol{x}$ and the category c is

$$\begin{aligned} p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\pi}, c|\boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\beta}) &= p(c|\boldsymbol{\eta}) p(\boldsymbol{\pi}|c, \boldsymbol{\theta}) \cdot \\ &\quad \prod_{n=1}^N p(z_n|\boldsymbol{\pi}) p(x_n|z_n, \boldsymbol{\beta}) \quad (2) \\ p(c|\boldsymbol{\eta}) &= \text{Mult}(c|\boldsymbol{\eta}) \quad (3) \\ p(\boldsymbol{\pi}|c, \boldsymbol{\theta}) &= \prod_{j=1}^C Dir(\boldsymbol{\pi}|\boldsymbol{\theta}_{j\cdot})^{\delta(c,j)} \quad (4) \\ p(z_n|\boldsymbol{\pi}) &= \text{Mult}(z_n|\boldsymbol{\pi}) \quad (5) \\ p(x_n|z_n, \boldsymbol{\beta}) &= \prod_{k=1}^K p(x_n|\boldsymbol{\beta}_{k\cdot})^{\delta(z_n^k, 1)} \quad (6) \end{aligned}$$

As Fig.3(a) shows, Theme Model 1 is a hierarchical representation of the scene category model. The Dirichlet parameter $\boldsymbol{\theta}$ for each category is a category-level parameters, sampled once in the process of generating a category of scenes. The multinomial variables $\boldsymbol{\pi}$ are scene-level variables, sampled once per image. Finally, the discrete theme variable $\boldsymbol{z}$ and patch $\boldsymbol{x}$ are patch-level variables, sampled every time a patch is generated.

If we wish to model the intermediate themes for each category without sharing them amongst all categories, we would introduce a link between the class node c to each patch x_n, such that $x_n \sim p(x_n|z_n, \boldsymbol{\beta}, c)$, where there are C different copies of $\boldsymbol{\beta}$, each of the size $K \times T$, where $\beta_{kt}^c = p(x_n^t|z_n^k = 1)$. The generative equations above (Eq.2-6) are hence changed according to this dependency on c. Due to space limitation, we shall omit deriving the learning and inference rules for this second theme model. We will release a technical note with this paper online for completeness.

2.1.2 Bayesian Decision

Before we show how we could proceed to learn the model parameters, let us first look at how decisions are made given an unknown scene. An unknown image is first represented by a collection of patches, or codewords. We keep the notation $\boldsymbol{x}$ for an image of N patches. Given $\boldsymbol{x}$, we would like to compute the probability of each scene class

$$p(c|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\eta}) \propto p(\boldsymbol{x}|c, \boldsymbol{\theta}, \boldsymbol{\beta}) p(c|\boldsymbol{\eta}) \propto p(\boldsymbol{x}|c, \boldsymbol{\theta}, \boldsymbol{\beta}) \tag{7}$$

where $\boldsymbol{\theta}$, $\boldsymbol{\beta}$ and $\boldsymbol{\eta}$ are parameters learnt from a training set. For convenience, the distribution of $p(c|\boldsymbol{\eta})$ is always assumed to be a fixed uniform distribution in which $p(c) = 1/C$. Therefore we will omit to estimate $\boldsymbol{\eta}$ from now on. Then the decision of the category is made by comparing the likelihood of $\boldsymbol{x}$ given each category: $c = \arg\max_c p(\boldsymbol{x}|c, \boldsymbol{\theta}, \boldsymbol{\beta})$. The term $p(\boldsymbol{x}|c, \boldsymbol{\theta}, \boldsymbol{\beta})$ is in general obtained by integrating over the hidden variables $\boldsymbol{\pi}$ and $\boldsymbol{z}$ in Eq.2.

$$p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{\beta}, c) = \int p(\boldsymbol{\pi}|\boldsymbol{\theta}, c) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n|\boldsymbol{\pi}) p(x_n|z_n, \beta) \right) d\boldsymbol{\pi} \tag{8}$$

Unfortunately Eq.8 is not tractable due to the coupling between $\boldsymbol{\pi}$ and $\boldsymbol{\beta}$ [1]. However, a wide range of approximate inference algorithms can be considered, including Laplace approximation, variational approximation and MCMC method [1]. In the following section, we briefly outline the variational method based on Variational Message Passing (VMP) [18], a convenient framework to carry out variational inferences.

2.1.3 Learning: Variational Inference

In learning, our goal is to maximize the log likelihood term $\log p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{\beta}, c)$ by estimating the optimal $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$. Using Jensen's inequality, we can bound this log likelihood in the following way.

$$\begin{aligned} \log p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{\beta}) &\geq \int \sum_{\boldsymbol{z}} q(\boldsymbol{\pi}, \boldsymbol{z}) \log p(\boldsymbol{\pi}, \boldsymbol{z}, \boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{\beta}) d\boldsymbol{\theta} - \\ &\quad \int \sum_{\boldsymbol{z}} q(\boldsymbol{\pi}, \boldsymbol{z}) \log q(\boldsymbol{\pi}, \boldsymbol{z}) \\ &= E_q \left[\log p(\boldsymbol{\pi}, \boldsymbol{z}, \boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{\beta}) \right] - E_q \left[\log q(\boldsymbol{\pi}, \boldsymbol{z}) \right] \end{aligned}$$

where $q(\boldsymbol{\pi}, \boldsymbol{z}|\gamma, \phi)$ could be any arbitrary variational distribution. By letting $L(\gamma, \phi; \boldsymbol{\theta}, \boldsymbol{\beta})$ denote the RHS of the above equation, we have:

$$\begin{aligned} \log p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{\beta}) &= L(\gamma, \phi; \boldsymbol{\theta}, \boldsymbol{\beta}) + \\ &\quad KL(q(\boldsymbol{\pi}, \boldsymbol{z}|\gamma, \phi) \parallel p(\boldsymbol{\pi}, \boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\beta})) \end{aligned} \tag{9}$$

where the second term on the RHS of the above equation stands for the Kullback-Leibler distance of two probability densities. By maximizing the lower bound $L(\gamma, \phi; \boldsymbol{\theta}, \boldsymbol{\beta})$ with respect to γ and ϕ is the same as minimizing the KL distance between the variational posterior probability and the true posterior probability.

Given Eq.9, we first estimate the variational parameters γ and ϕ. Substituting the variational lower bound as a surrogate for the (intractable) marginal likelihood, we can then in turn estimate the model parameters $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$. The iterative algorithm alternates between the following two steps till convergence. We will soon publish a technical note with detailed derivations on our website.

1. (E-step) For each class of images, optimize values for the variational parameters γ and ϕ. The update rules are

$$\gamma_i = \boldsymbol{\theta}_i + \sum_{n=1}^{N} \phi_{ni} \tag{10}$$

$$\phi_{ni} \propto \beta_{i\nu} \exp \left[\Psi(\gamma_i) - \Psi(\sum_{j=1}^{K} \gamma_j) \right] \tag{11}$$

   where i is the image index, n the patch index and $\Psi(\cdot)$ a digamma function.

2. (M-step) Maximize the reuslting lower bound on the log likelihood with respect to the model parameters $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$. We can do this by finding the maximum likelihood estimates with expected sufficient statistics computed in the E-step [1, 8].

2.1.4 A Brief Comparison

We can compare this hierarchical model with a traditional texton model for texture recognition, for instance [5, 16]. Fig.3(c) is a graphical representation of a traditional texton model. We see here that for a given class of textures or materials, only a single multinomial parameter $\boldsymbol{\beta}$ is associated with the class. In other words, to generate an image, all patches are drawn from a single "theme". This might be fine when the training data are "pure" textures segmented manually. Since there is no "contaminations" of other "themes", the single mixture learnt from the codewords might suffice. As shown by [5], this framwork may be further extended by training different models for the same category of textures under different lighting and view point conditions. This again requires manual separations of data and labelling of the segmented textures. In Section 4, we will show empirically that by explicitly modelling the intermediate themes in these complex scenes, our model achieve better recognition performances than the traditional "texton" model in Fig.3(c).

2.2 Features & Codebook

In the formulation of the theme model, we represent each image as a collection of detected patches, each assigned a membership to a large dictionary of codewords. We show now how these patches are obtained and memberships assigned.

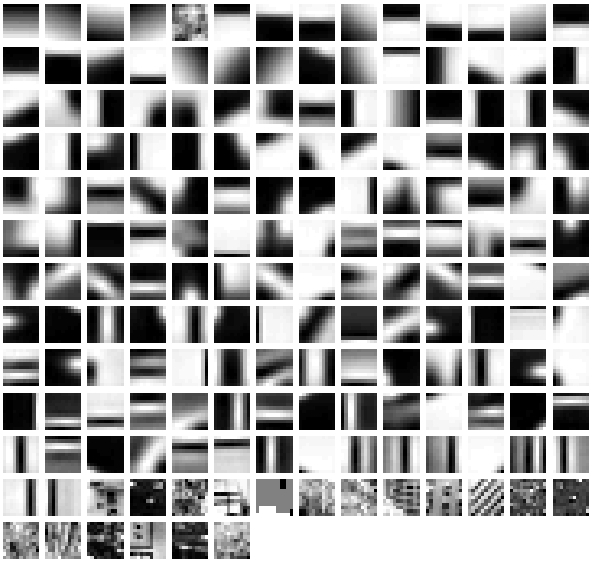


Figure 4. A codebook obtained from 650 training examples from all 13 categories (50 images from each category). Image patches are detected by a sliding grid and random sampling of scales. The codewords are sorted in descending order according to the size of its membership. Interestingly most of the codewords appear to represent simple orientations and illumination patterns, similar to the ones that the early human visual system responds to.

.

2.2.1 Local Region Detection and Representation

While most previous studies on natural scene categorization have focused on using global features such as frequency distribution, edge orientations and color histogram [3, 11, 15], recently it has been shown local regions are very powerful cues [17]. Compared to the global features, local regions are more robust to occlusions and spatial variations. We have tested four different ways of extracting local regions.

1. *Evenly Sampled Grid.* An evenly sampled grid spaced at 10×10 pixels for a given image. The size of the patch is randomly sampled between scale 10 to 30 pixels.
2. *Random Sampling.* 500 randomly sampled patches for a given image. The size of the patch is also randomly sampled between scale 10 to 30 pixels.
3. *Kadir & Brady Saliency Detector.* Roughly $100 \sim 200$ regions that are salient over both location and scale are extracted using the saliency detector [4]. Scales of each interest point are between 10 to 30 pixels.
4. *Lowe's DoG Detector.* Roughly $100 \sim 500$ regions that are stable and rotationally invariant over different scales are extracted using the DoG detector [7]. Scales of each interest point vary between 20 to 120 pixels.

We have used two different representations for describing a patch: normalized 11×11 pixel gray values or a $128-$dim SIFT vector [7]. Table 1 compares and contrasts the experimental results of the model based on different local region detectors and representations.

2.2.2 Codebook Formation

Given the collection of detected patches from the training images of all categories, we learn the codebook by performing k-means algorithm [5]. Clusters with too small a number of members are further pruned out. Codeswords are then defined as the centers of the learnt clusters. Fig.4 shows the 174 codewords learnt from the gray value pixel intensities.

3. Dataset & Experimental Setup

Our dataset contains 13 categories of natural scenes (Fig.1): highway ([9], 260 images), inside of cities ([9], 308 images), tall buildings ([9], 356 images), streets ([9], 292 images), suburb residence (241 images), forest ([9], 328 images), coast ([9], 360 images), mountain ([9], 374 images), open country ([9], 410 images), bedroom (174 images), kitchen (151 images), livingroom (289 images) and office (216 images). The average size of each image is approximately 250×300 pixels. The 8 categories that are provided by Oliva and Torralba were collected from a mixture of COREL images as well as personal photographs [9]. The rest of the 5 categories are obtained by us from both the Google image search engine as well as personal photographs. It is also worth noting that 4 (coast, forest, open country and mountain) of the categories are similar to the 4 of the 6 categories reported in [17]. But unlike them, we only use grayscale images for both learning and recognition. We believe that this is the most complete scene category dataset used in literature thus far.

Each categories of scenes was split randomly into two separate sets of images, N (100) for training and the rest for testing. A codebook of codewords was learnt from patches drawn from a random half of the entire training set. A model for each category of scenes was obtained from the training images. When asked to categorize one test image, the decision is made to the category label that gives the highest likelihood probability. A confusion table is used to illustrate the performance of the models. On the confusion table, the x-axis represents the models for each category of scenes. The y-axis represents the ground truth categories of scenes. The orders of scene categories are the same in both axes. Hence in the ideal case one should expect a completely white diagonal line to show perfect discrimination power of all category models over all categories of scenes. Unless otherwise specified, all performances in Section 4 are quoted as the average value of the diagonal entries of the confusion table. For a 13-category recognition task, random chance would be 7.7%. Excluding the preprocessing time of feature detection and codebook formation, it takes a few minutes (less than 10) to obtain 13 categories of models (100 training images for each category) on a 2.6 Ghz machine.

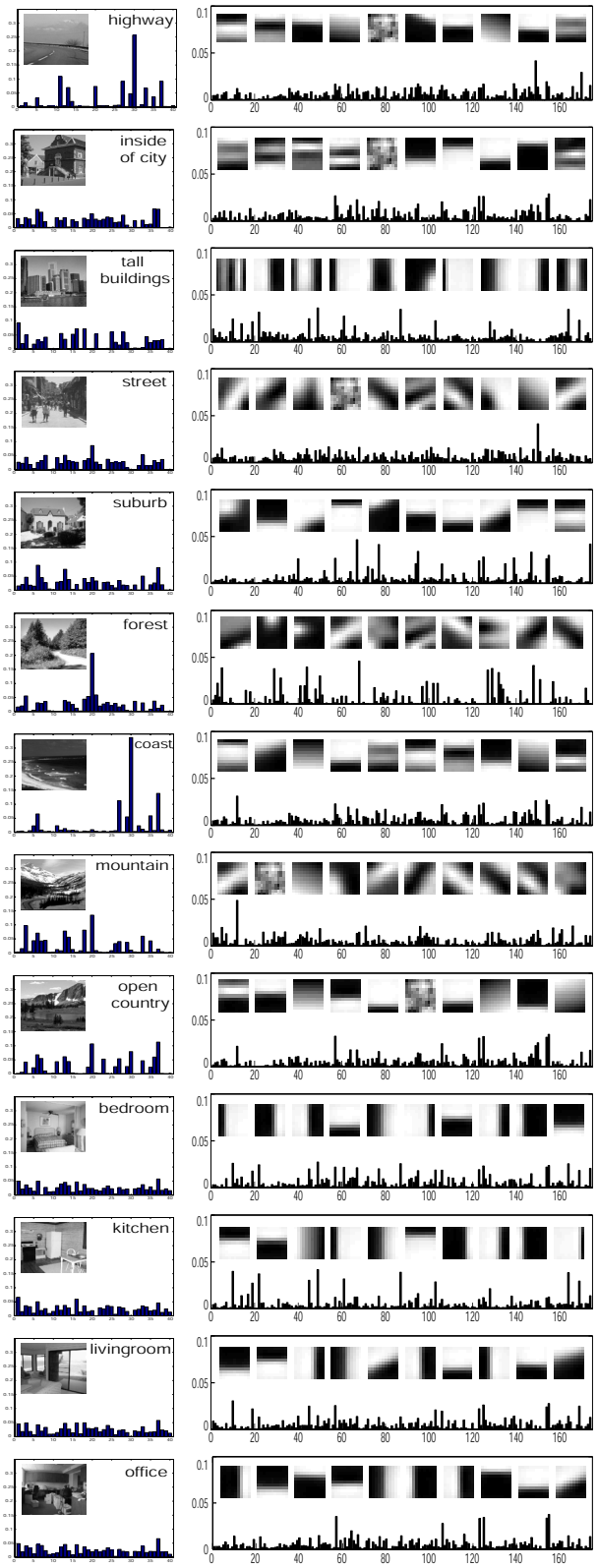


Figure 5. Internal structure of the models learnt for each category. Each row represents one category. The left panel shows the distribution of the 40 intermediate themes. The right panel shows the distribution of codewords as well as the appearance of 10 codewords selected from the top 20 most likely codewords for this category model.

Figure 6. Examples of testing images for each category. Each row is for one category. The first 3 columns on the left show 3 examples of correctly recognized images, the last column on the right shows an example of incorrectly recognized image. Superimposed on each image, we show samples of patches that belong to the most significant set of codewords given the category model. Note for the incorrectly categorized images, the number of significant codewords of the model tends to occur less likely. (This figure is best viewed in color.)

4. Results

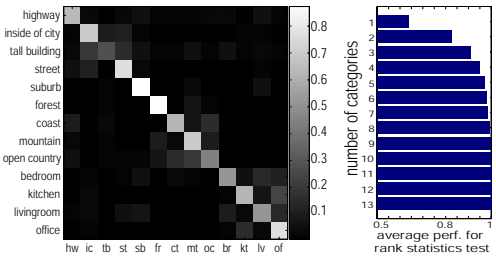


Figure 7. Left Panel. Confusion table of Theme Model 1 using 100 training and 50 test examples from each category, the grid detector and patch based representation. The average performance is 64.0%. **Right Panel.** Rank statistics of the confusion table, which shows the probability of a test scene correctly belong to one of the top N most probable categories. N ranges from 1 to 13.

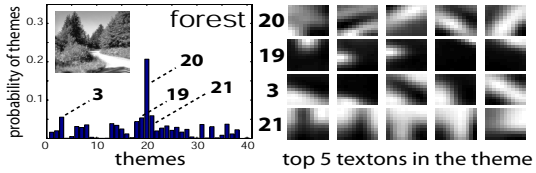


Figure 8. Example of themes for the forest category. **Left Panel** The distribution of all 40 themes. **Right Panel** The 5 most likely codewords for each of the 4 dominant themes in the category. Notice that codewords within a theme are visibly consistent. The "foliage" (#20, 3) and "tree branch" (#19) themes appear to emerge automatically from the data.

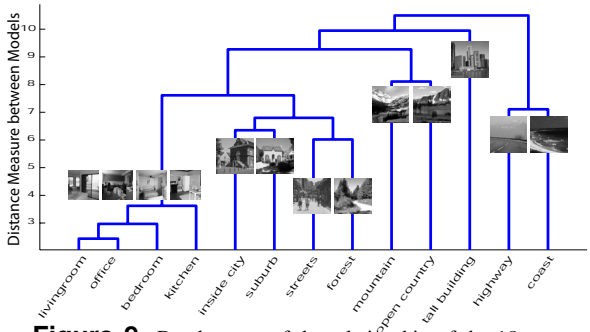


Figure 9. Dendrogram of the relationship of the 13 category models based on theme distribution. y-axis is the pseudo-euclidean distance measure between models.

Fig.7 is an overview of the performance of the Theme Model 1. We used a total number of 40 themes. A closer look at the confusion table (Fig.7(a)) reveals that the highest block of errors occurs among the four indoor categories: bedroom, livingroom, kitchen and office. Another way to evaluate the performance is to use the rank statistics of the categorization results (Fig.7(b)). Using both the best and second best choices, the mean categorization result increases to 82.3%.

Both Fig.5 & 8 demonstrate some of the internal structure of the models learnt for each category. Take the "highway" category as an example in Fig.5. The left panel shows

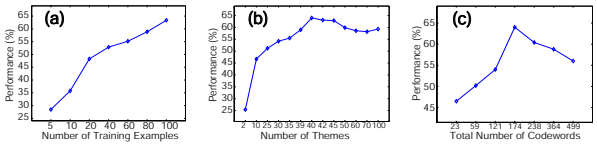
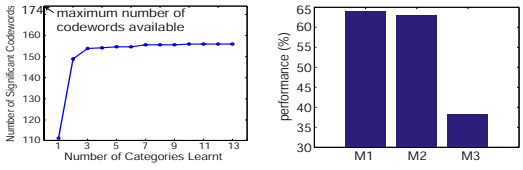


Figure 10. (a) Number of training examples vs. performance. **(b)** Number of themes vs. performance. **(c)** Number of codewords vs. performance. All performances are quotes from the mean of the confusion table.



(a) (b)

Figure 11. (a) Number of significant codewords as a function of the number of categories learnt. "Significance" is defined as 90% of the probabilistic weight. **(b)** Performance comparison among Theme Model 1 (M1), Theme Model 2 (M2) and the traditional texton model (M3, e.g. [16].)

the average distribution of the 40 intermediate themes for generating highway images. In the right panel, we show that after a large number of samplings (1000), the average distribution of all codewords for generating highway images. Clearly, this distribution of codewords (174, Fig.4) is very much influenced by the distribution of themes. We show in the right panel 10 of the top 20 codewords that are most likely to occur in highway images. Note that horizontal lines dominate the top choices. This is to be constrasted, for instance, to the likely codewords for the tall building category. We see that most of the top-choice codewords are vertical edges in the case of tall buildings. The 4 indoor categories all tend to have sharp horizontal and vertical edges. This is quite revealing of the scene statistics for these man-made, indoor structures. The distribution of both the themes and the codewords of the four indoor categories further indicates the confusion among these four categories. Fig.6 then shows some testing image examples.

We can further establish some relationship among the categories by looking at the model distances among them (see the dendrogram in Fig.9). When the distribution of the themes are close, the categories would also be close to each other on the dendrogram. For example, the closest categories are the 4 indoor environments.

Fig.10 illustrates 3 different aspects of the algorithm:

Descriptor	Grid	Random	Saliency [4]	DoG [7]
11 × 11 Pixel	64.0%	47.5%	45.5%	N/A
128-dim Sift	65.2%	60.7%	53.1%	52.5%

Table 1. Performance comparison given different feature detectors and representations. The performance is quoted from the mean of the confusion table similar to that of Fig.7. SIFT representation seems to be in general more robust than the pixel grayvalue representation. The sliding grid, which yields the most number of patches, out performs all other detectors.

performances versus the number of training examples (a), of themes (b) and of codewords in the codebook (c). Fig.11(a) shows that by sharing the resources of codewords and intermediate themes, the number of significant codewords for learning more and more new models tend to level off quickly [13]. Table 1 shows how different feature detection and representation influences the performance.

5. Summary & Discussion

We have proposed a Bayesian hierarchical model to learn and recognize natural scene categories. The model is an adaptation to vision of ideas proposed recently by [1] in the context of document analysis . While previous schemes [9, 17] require a detailed manual annotation of the images in the training database, our model can learn characteristic intermediate "themes" of scenes with no supervision, nor human intervention and achieves comparable performance to [17] (see Table 2 for details.).

	# of categ.	training # per categ.	training requirements	perf. (%)
Theme Model 1	13	100	unsupervised	76
[17]	6	~ 100	human annotation of 9 semantic concepts for $60,000$ patches	77
[9]	8	$250 \sim 300$	human annotation of 6 properties for thousands of scenes	89

Table 2. Comparison of our algorithm with other methods. The average confusion table performances are for the 4 comparable categories (forest, mountain, open country and coast) in all methods. We use roughly $1/3$ of the number of training examples and no human supervision than [9]. Fig.10(a) indicates that given more training examples, our model has the potential of achieve higher performances.

Our model is based on a principled probabilistic framework for learning automatically the distribution of codewords and the intermediate-level themes, which might be thought to be akin to texture descriptions. Fig.11(b) shows that this model outperforms the traditional "texton models" where only a fixed codeword mixing pattern is estimated for each category of scenes [16]. One way to think about our model is as a generalization of the the "texton models" [5, 16] for textures, which require samples of "pure" texture to be trained. By contrast, our model may be trained on complete scenes and infer the intermediate "themes" from the data. In the future, it is important to further explore this relationship between the "themes" to meaningful textures such as the semantic concepts suggested by [9, 17]. In addition, we provide a framework to share both the basic level codewords as well as intermediate level themes amongst different scene categories. Similarly to what [13] found, the number of features to be learnt increases sub-linearly as the number of new categories increases.

We tested our algorithm on a diverse set of scene types, introducing a number of new categories (13 here, as opposed to 4+4 in [9] and 6 in [17]). The lackluster performances for the indoor scenes suggest that our model is not complete. At a minimum, we need a richer set features: by using different cues as well as a hierarchy of codewords, we might be able to form much more powerful models for these difficult categories.

Acknowledgment. We would like to thank Chris Bishop, Tom Minka, Silvio Savarese and Max Welling for helpful discussions. We also thank Aude Oliva and Michael Fink for providing parts of the dataset.

References

[1] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[2] A. Gelman, J.B. Carlin, Stern H.S., and Rubin D.B. *Bayesian Data Analysis*. Chapman Hall/CRC, 1995.

[3] M. Gorkani and R. Picard. Texture orientation for sorting photos at a glance. In *Int. Conf. on Pattern Recognition*, 1994.

[4] T. Kadir and M. Brady. Scale, saliency and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.

[5] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV*, 43(1):29–44, June 2001.

[6] F.F. Li, R. VanRullen, C. Koch, and P Perona. Natural scene categorization in the near absence of attention. *Proc Natl Acad Sci USA*, 99(14):9596–9601, 2002.

[7] D. Lowe. Object recognition from local scale-invariant features. In *Proc. International Conference on Computer Vision*, pages 1150–1157, 1999.

[8] T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. In *Proc. of the 18th Conference on Uncertainty in Artificial Intelligence.*, pages 352–359, 2002.

[9] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *Int. Journal of Computer Vision.*, 42, 2001.

[10] J. Portilla and E.P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *IJCV*, 40(1):49–70, October 2000.

[11] M. Szummer and R. Picard. Indoor-outdoor image classification. In *Int. Workshop on Content-based Access of Image and Vedeo Databases*, Bombay, India, 1998.

[12] S. Thorpe, D. Fize, and C. Marlot. Speed of processing in the human visual system. *Nature*, 381:520–522, 1996.

[13] A. Torralba, K. Murphy, and W.T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *Proc. of the 2004 IEEE CVPR.*, 2004.

[14] A. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12:97–136, 1980.

[15] A. Vailaya, M. Figueiredo, A. Jain, and H. Zhang. Image classification for content-based indexing. *IEEE Trans. on Image Processing.*, 10, 2001.

[16] M. Varma and A. Zisserman. Texture classification: are filter banks necessary? In *CVPR03*, pages II: 691–698, 2003.

[17] J. Vogel and B. Schiele. A semantic typicality measure for natural scene categorization. In *DAGM'04 Annual Pattern Recognition Symposium*, Tuebingen, Germany, 2004.

[18] J. Winn. *Variational Message Passing and its applications.* PhD thesis, University of Cambridge, 2003.