# Using dependent regions for object categorization in a generative framework

Gang Wang[1,2] Ye Zhang[2] Li Fei-Fei[1]

[1] Dept. of Electrical and Computer Engineering
University of Illinois Urbana-Champaign (UIUC)
405 N. Mathews Ave. Urbana, IL 61801, USA
feifeili@uiuc.edu

[2] Dept. of Information Engineering
Harbin Institute of Technology
Harbin, China

## Abstract

*"Bag of words" models have enjoyed much attention and achieved good performances in recent studies of object categorization. In most of these works, local patches are modeled as basic building blocks of an image, analogous to words in text documents. In most previous works using the "bag of words" models (e.g. [4, 20, 7]), the local patches are assumed to be independent with each other. In this paper, we relax the independence assumption and model explicitly the inter-dependency of the local regions. Similarly to previous work , we represent images as a collection of patches, each of which belongs to a latent "theme" that is shared across images as well as categories. We learn the theme distributions and patch distributions over the themes in a hierarchical structure [22]. In particular, we introduce a linkage structure over the latent themes to encode the dependencies of the patches. This structure enforces the semantic connections among the patches by facilitating better clustering of the themes. As a result, our models for object categories tend to be more discriminative than the ones obtained under the independent patch assumption. We show highly competitive categorization results on both the Caltech 4 and Caltech 101 object category datasets. By examining the distributions of the latent themes for each object category, we construct an object taxonomy using the 101 object classes from the Caltech 101 datasets.*

## 1.Introduction

Generic object recognition is a central research topic in computer vision in recent years. Humans are known to effortlessly recognize many objects and object categories. But the task remains extremely challenging for computers and robots. On the individual object level, variations in lighting, geometric transformations, occlusion and clutter pose many challenges for efficient learning and robust recognition. In addition to these difficulties, recognizing object categories also needs to overcome the great intra-class variability among the different members. Beyond categorizing objects into distinct groups, the question of inter-category relationships remains largely unexplored.

In recent years, a rich palette of diverse ideas has been proposed for object categorization. One type of popular approaches emphasizes on the part-based structure of the objects, such as the "constellation model" (e.g.[8, 6, 24]), a recent $k$-fans model [3] and local shape correspondences [1]. There are also some discriminative part-based models that aim to find the boundaries among categories (e.g.[10, 18, 23]). While part-based model aspires to represent a rigorous geometric relationship among the different parts, it suffers a computational difficulty of having to search among exponentially large number of hypotheses to solve the correspondence problem [8]. Recently, "bag of words" models have made great progress in object categorization while avoiding this problem. Initial"bag of words" model starts with modeling some signature distributions of textons (or codewords) [15, 4] without any "latent themes". Inspired by recent text modeling progress, some researchers (e.g. [7, 20]) show lately that using intermediate themes can improve recognition performance greatly. In these models, the algorithms learn both the probability distributions of the codewords as well as the intermediate themes.

"Bag of words" models, however, assume local patches of an image are independent with each other. Though this assumption simplifies computations greatly, it does not take into account useful information encoded in the inter-relationships among the patches. For example, for the object category of the side view of cars, the "wheel" and "window" tend to occur together in a same image. If we neglect the dependency, we may confuse car wheel with the motorbike wheel, as illustrated in (Fig.1)(a). But if we know there is a "window" near by (Fig.1(b)), we can easily distinguish the two different kinds of wheels, and in turn enhance the accuracy of recognizing the "car" category over the "motorbike" category. This observation prompts us to relax the independence assumption toward modeling more accurately the inter-dependency of the patches of an image.

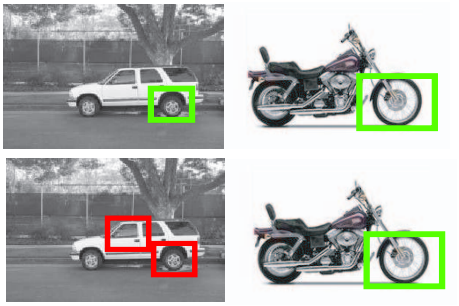


Figure 1. Dependency between features carries useful information for object recognition. **Top row.** A car wheel can be easily confused with a motorbike's wheel, hence providing little information toward discriminating between the "car" and the "motorbike." **Bottom row.** Knowing the existence of a "window", a car wheel can be more easily distinguished, and therefore enhancing the discriminability between the "car" and the "motorbike."

Inspired by a recent model for text grammar [12], we incorporate the dependency relationship between patches through a structure called *linkage*. The linkage structure serves to store information on the strength of the dependencies among local patches. During the clustering procedure, this information becomes very useful to hold together patches that are highly related (such as the "window" and "wheel" of a car). The linkage structure assumes that patch dependencies form an acyclic, planar graph, where two related patches are linked by a graph edge.

Beyond the problem of object categorization, we are also interested in discovering the relationships among the object categories. In some previous papers (e.g. [7, 23]), the authors have tried to find the similarity among different classes of images. Here we aim to group the object categories into a semantic hierarchical structure and construct a crude taxonomy for the object categories using the Caltech 101 database [5]. We cluster categories by the latent theme distributions.

Our model is an extension of the hierarchical Dirichlet process (HDP) proposed by [22]. HDP is a non-parametric Bayesian model that infers the number of latent themes from training data. It assumes a hierarchical structure such that data in different groups can share the same themes. In a recent work [21], Sudderth et al have extended the HDP model by considering the relative spatial locations of local patches to describe objects. Our approach differs from theirs by emphasizing on the semantic dependency among patches, rather than explicitly modeling the geometric locations of parts. We therefore name our model the dependent Hierarchical Dirichlet process (DHDP).

Under the generative framework, we also propose a semi-supervised variation of DHDP that requires the labeling of local patches in a few images for each category. When there are plenty of training images, this small amount of labor could help to achieve higher performance accuracy and a better characterization of the object categories.

We carry out recognition experiments on the Caltech 4 categories and the Caltech 101 dataset. Our approach obtains one of the state-of-the-art recognition result on the Caltech 101 dataset. By exploring the distributions of the latent themes and their inter-relationship, we make an attempt to form a taxonomy structure of the Caltech 101 object categories that reflects semantically meaningful structures.

In the Section 2, we introduce the DHDP model and the inference algorithm. Section 3 details the implementation procedure of our system. We show the experimental results on the two datasets in Section 4. Finally we conclude this work in Section 5.

## 2. Our Approach

Similarly to ([4, 7, 20]), we model an image as a collection of local patches. Each patch is represented by a codeword selected from a large dictionary of codewords. A latent theme is assigned to each local patch. Through the linkage structure, dependent local patches are more likely to share the same themes. We use Markov chain Monte Carlo sampling scheme to perform inference. For each class, we could sample the posterior distribution and obtain a probability matrix as well as a latent theme distribution. Given an unknown image, an object category is assigned to it according to the highest likelihood probability given each category model. Furthermore, the theme distributions are used to form the taxonomy structure of 101 object categories. In the following subsections, we first introduce in details the latent theme model DHDP. Then we show the parameters sampling scheme and Bayesian decision. Further more, we introduce a semi-supervised variation of DHDP which just needs to label several images for each class.

### 2.1. Dependent hierarchical Dirichlet process (DHDP)

We first define some notations for the model.

- A *patch* $x$ is the basic unit of an image, each patch is defined by a codeword member of the visual dictionary of codewords indexed by $\{1, \ldots, T\}$.
- An *image* is a collection of $N$ patches denoted by $\mathbf{x} = (x_1, x_2, \ldots, x_N)$, where $x_n$ is the $n^{\text{th}}$ patch of the image.
- A *category* is a collection of $I$ images denoted by $\mathbf{D} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_I)$.

Fig.2(a) shows the graphical model depiction of the dependent Hierarchical Dirichlet Process (DHDP). We also compare and contrast them with the plain HDP model by [22] and the LDA model by [2] in Fig.2(b) and (c). In DHDP, we have a probability measure $H$ in the measure space $\Theta$, and a positive real number $\gamma$. $\theta$ denotes a parameter taking values in the measure space

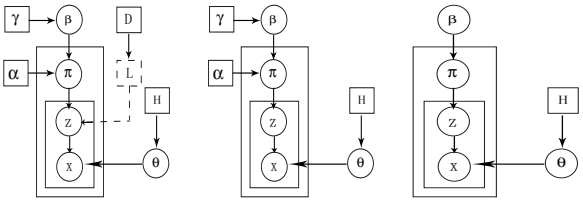



Figure 2. Graphical model depiction of DHDP **(a)**, HDP **(b)** and LDA **(c)** models. DHDP extends the "bag of words" models ((b) and (c)) by a linkage structure "L", denoted in dashed box in (a).

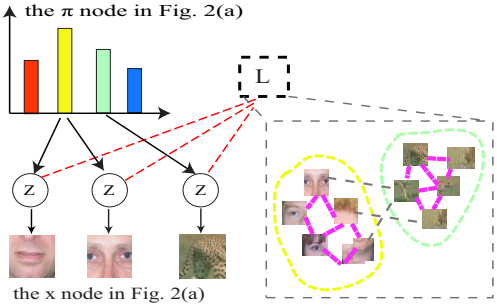



Figure 3. A "zoomed-in" version of Fig.2(a). "L" denotes the linkage structure for the corpus, each pair of codewords (shown as image patches) are linked by a graph edge: solid magenta lines indicate the two patches are highly dependent, while the dashed gray lines indicate the two patches are weakly dependent. The highly dependent patches are grouped together illustratively in the figure (the yellow and cyan enclosures under the "L" structure), showing the fact that they are highly likely to be clustered into the same theme in the sampling procedure. This figure is best viewed in colors.

.

with prior $H$, $\theta_i|H \sim H$. $\theta_i$s correspond to the set of latent themes that are shared among different categories. A Dirichlet process $G_0 \sim DP(\gamma, H)$ is a distribution over measures on $\Theta$, which can be denoted by a stick-breaking construction as Eq.1. $G_0$ is an unobserved variable in the graphical model. We can better elucidate the model with variable $\beta$ from following:

$$\begin{aligned} G_0 &= \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k}, \beta_k^{'}|\gamma, H = Beta(1,\gamma) \\ \beta_k &= \beta_k^{'} \Pi_{l=1}^{k-1}(1-\beta_l^{'}) \end{aligned} \quad (1)$$

where $\beta_k$ denotes the probability for drawing $\theta_k$. We associate each image with a Dirichlet process $G_j$, which acts as the prior of mixture models in different images. To enable the patches in different images to share the latent themes $\theta_k$, hierarchical Dirichlet process forces $G_j$ to be drawn from $G_0$, which has the following property:

$$G_j = \sum_{k=1}^{\infty} \pi_{jk} \delta_{\theta_k}, \pi_j|a_0, \beta \sim DP(a_0, \beta) \quad (2)$$

By this structure, mixture components drawn from $G_j$ can share the corpus level themes. $x_{ji}$ is the $i$th patch in $j$th image. The mixture component or the latent theme drawn is denoted by factor $z_{ji}$, and $z_{ji}$ is drawn from $\pi_j$, which is also affected by the linkage:

$$z_{ji}|\pi_j, L \sim (\pi_j, L) \quad (3)$$

Then $x_{ji}$ is generated by the following likelihood:

$$x_{ji}|z_{ji}, \theta_k \sim F(\theta_{z_{ji}}) \quad (4)$$

where $F(\theta_{z_{ji}})$ denotes distribution of $x_{ji}$ given $\theta_{z_{ji}}$.

Fig.3 can be viewed as a "zoomed-in" version of Fig.2(a). Here we show on the variables directly involved in the generation of the latent themes and the local patches. "L" denotes the linkage structure for the corpus, each pair of codewords (shown as image patches in the figure) are linked by a graph edge: solid magenta lines indicate the two patches are highly dependent, while the dashed gray lines indicate the two patches are weakly dependent. The highly dependent patches are grouped together illustratively in the figure, showing the fact that they are highly likely to be clustered into the same theme in the sampling procedure. In most situations, relevant patches in the same object category are dependent. Our model, therefore, can cluster the patches from a class into the same themes, thus enhancing the discriminability for categorization.

The clustering property of above process can be better understood in the metaphor "acquaintance Chinese restaurant franchise". We adopt this metaphor from [22]. It assumes that there is a set of restaurants (each restaurant denoting an image). These restaurants share the same dishes, which correspond to the corpus level mixture components (also the latent themes). Customers in each restaurant correspond to the patches in an image. In our dependent model, we want to cluster the dependent patches into the same theme. Metaphorically, it means that acquainted customers order the same dish. The dependency information between two patches is encoded by the "acquaintance coefficient" between their factors. We will show how to obtain its values in Section 2.2. $\phi_{ji}$ is associated with one $\varphi_{jt}$, which denotes the table that $\phi_{ji}$ takes in the $j$th restaurant. In the image modeling problem, $\varphi_{jt}$ could be interpreted as the image level mixture component. There are many $\phi_{ji}$s associated with one $\varphi_{jt}$ because there are usually many customers at a table. Let $n_{jt}$ be the number of the $\phi_{ji}$ associated with $\varphi_{jt}$. $\varphi_{jt}$is also associated with one $\theta_k$, the dish ordered by the table $\varphi_{jt}$. Let $m_{jk}$ be the number of $\varphi_{jt}$ associated with $\theta_k$ in the $j$th restaurant. Then we have $m_k = \sum_j m_{jk}$, the number of $\varphi_{jt}$ associated with over $\theta_k$ all $j$ .

Our algorithm differs from the standard "Chinese restaurant franchise" by introducing the dependency structure between patches. Given $\phi_{j1}, \ldots, \phi_{ji-1}$, $\phi_{ji}$

chooses a table with the following likelihood:

$$\phi_{ji}|\phi_{j1},\ldots,\phi_{ji-1},a_0,G_0 \sim \sum_{t=1}^{T_j} n_{jt}\prod_{q=1}^{n_{jt}}(1+C(\phi_{ji},\phi^q_{\varphi_{jt}}))\delta_{\varphi_{jt}}+a_0G_0 \quad (5)$$

where $T_j$ is the number of tables with customers. For $1 \le m < i$, $C(\phi_{ji},\phi^q_{\varphi_{jt}})$ denotes the acquaintance coefficient between $\phi_{ji}$ and the former customer at this table.

Since acquaintance share the same dish, the dish ordered by table $\varphi_{jt}$ is strongly affected by their acquaintances in all the restaurants. Assuming that there are $p$ customers at $\varphi_{jt}$, then the dish they order has the following likelihood:

$$\varphi_{jt}|\varphi_{11},\varphi_{12},\ldots,\varphi_{21},\ldots,\varphi_{jt-1},\gamma,H \sim \sum_{k=1}^{K} m_k\prod_{p=1}^{P}\prod_{q=1}^{m_k}(1+C(\phi^p_{\varphi_{jt}},\phi^q_{\theta_k}))\delta_{\theta_k}+\gamma H \quad (6)$$

where $K$ denotes the number of dishes that have been ordered by former customers, and $m_k$ is the number of customers that have ordered dish $\theta_k$. By this process, acquaintances aptly order the same dish.

## 2.2. Parameter Estimation

In this section we describe a Markov Chain Monte Carlo sampling scheme for the DHDP model based on "acquaintance Chinese restaurant franchise". When given a patch $x$, this scheme will sample a theme for it. Our goal is to obtain a probability matrix and the theme distribution for every category. During training, we assume that we know the category label of each trained image. Therefore the theme assignment results for each patch in the same category is used to generate the posterior probability matrix. In addition, we have a probability matrix for the corpus that is updated accordingly for the sampling of next patch. The theme distribution for a given category of images is modeled as the ratio of the number of times it is sampled and the total number of samples during the training process. Before introducing the sampling algorithm, we first define "acquaintance coefficient" between two patches by:

$$C(w_1,w_2)=\frac{R(w_1,w_2,I)}{R(w_1)+R(w_2)}-\frac{R(w_1,w_2,I)}{R(w_1)+R(w_2)^\lambda} \quad (7)$$

where $R(w_1,w_2,I)$ is the number of times they appear in the same image. And $R(w_1)+R(w_2)$ denotes the total number of times they appear in the corpus. $\frac{R(w_1,w_2,I)}{R(w_1)+R(w_2)^\lambda}$ is a penalty factor which prevents the patches appear rarely to become highly dependent with each other. $\lambda$ is an experimental parameter set to be slightly bigger than 1. We now show how to sample the posterior probability given a patch $x$. Similarly to [22], rather than dealing with the $\phi_{ji}$s and $\varphi_{jt}$s directly, we sample their index variables $t_{ji}$ (imagine it as the index of the table which $x$ takes in the $j$th restaurant) and $k_{jt}$ (imagine it as the index of the dish which $x$ orders).

**Sampling t.** The probability that $x$ picks the table $t$ is proportional to $n_{jt}$ and its dependency with other customers associated with $t$. And the probability that it chooses on a new table is proportional to $a_0$. Before sampling $t$, we first generate a new sample for $k_{jt^{new}}$ because $t$ may take on a new value:

$$k_{jt^{new}}|k \sim \sum_{k=1}^{K} m_k\delta_k+\gamma\delta_{k^{new}} \qquad \delta_{k^{new}} \sim H \quad (8)$$

Following Eq.5, the sampling of $t_{ji}$ is given by:

$$\begin{cases} a_0 f(x_{ji}|\theta_{k_{jt}}) & \text{if } t=t^{new} \\ n_{jt}^{-i}\prod_{q=1}^{n_{jt}}(1+C(x_{ji},x_t^q))f(x_{ji}|\theta_{k_{jt}}) & \text{if } t \text{ is previously used} \end{cases}$$

If the sampled value of $t_{ji}$ is $t^{new}$, we insert the temporary values of $k_{jt^{new}}$ and $\theta_{jt^{new}}$ into the data structure; otherwise these temporary variables are discarded.

**Sampling k.** Before sampling $k_{jt}$, we also first generate a new mixture parameter $\delta_{k^{new}} \sim H$. By Eq.6, the likelihood of setting $k_{jt}=k$ is given by the following formula:

$$\begin{cases} \gamma\prod_{i:t_{ji}=t} f(x_{ji}|\theta_k) & \text{if } k=k^{new} \\ m_k^{-t}\prod_p\prod_q^{m_k}(1+C(x_{jt}^p,x_k^q))\prod_{i:t_{ji}=t} f(x_{ji}|\theta_k) & \text{if } k \text{ is previously used} \end{cases}$$

## 2.3. A semi-supervised variation of DHDP

In the previous section, we have discussed DHDP, which is a weakly supervised method. However, sometimes we may be interested in using some (optional) supervised information to improve the performance. So in this section, we introduce a semi-supervised variation of DHDP. We first construct the visual codewords as the standard DHDP, then label the distinctive parts of each category, such as the "eye" and "mouth" for "face". Since we have clustered the similar patches in the codewords construction stage, we can simple label a handful of images and know the exact codewords associated with such distinctive patches. We then assign themes for these labeled codewords before training. In the training procedure, for the unlabeled patches, we train them as the DHDP, but if they "sit" at the same table with a labeled patch, they will be sampled to the "labeled" theme; and the labeled patches are sampled to their associated themes with probability 1. Combined with the linkage structure, we can expect this semi-supervised method to cluster the patches sensibly to the categories very well.

This method is suitable for the situation that there are many training images, in which we could just label several images and get the distinctive information of this image class.

### 2.4. Recognition and taxonomy formation

We have trained a probability matrix $p(x_j \mid \theta_i^c)$ for every object category $c$, where $\theta_i^c$ denotes the mixture components, and $x_j$ denotes the codewords from visual vocabulary. Assuming that an unknown testing image $I$ has $M$ local patches, we first represent the local patches as codewords $x_m$,$m = \{1, \ldots, M\}$, and calculate the probability $p(I \mid c)$ for each class :

$$p(I \mid c) = \prod_{m=1:M} p(x_m \mid c)$$
$$= \prod_{m=1:M} (\sum_i p(x_m \mid \theta_i^c) p(\theta_i^c \mid c)) \quad (9)$$

Categorization decision can be then made by choosing the category model that yields the highest probability.

$$c = argmax_c p(I \mid c) \quad (10)$$

Compared to the HDP model, the DHDP better models of the latent themes that are shared across the object categories. Much of the inter-category relationships among different object categories can be captured by these theme distributions. Theme distribution for class $c$ is denoted as a vector $p(\theta_i \mid c)$ , where $\theta_i$ denotes all the themes that have been sampled in the whole procedure for the corpus. Using these theme distributions, we construct the taxonomy of a set of object categories (in our case the Caltech 101 database) in a recursive clustering procedure using the k-means algorithm (see Fig.12).

## 3. Implementation

Algorithm.1 is a summary of the algorithm in both learning and recognition. Each image is represented by local regions. We extract these local image regions (approximately $30 \sim 40$ local regions per image) using the saliency region detector [13] . Each local patch is resized to $48 \times 48$ pixels, and further divided into four $24 \times 24$ sub-regions. Each sub-region is then denoted by a 18-dim gradient bins similar to SIFT descriptor [16]. Concatenating these four sub-region descriptors together, we obtain a 72-dim vector for each local patch. For computational efficiency, we represent each local patch as a 15-dim vector by obtaining the first 15 PCA coefficients for each 72-dim feature vector. The PCA-basis is constructed by using the local regions detector on the images of the "background" category of the Caltech 101 dataset. We then use a K-means algorithm to cluster these descriptors from all training images of all categories to form a "codeword dictionary". Codewords are then defined as the centers of the learnt clusters.

**Algorithm 1** The learning and recognition processes.

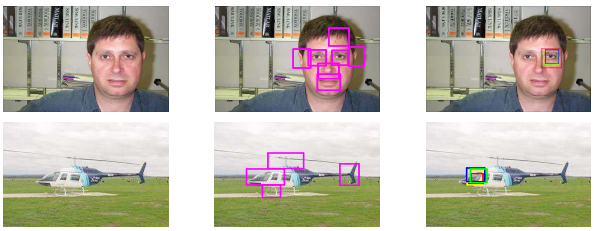
**Learning**

1. Extract local regions from the training images
2. Cluster the extracted local regions from all training images to form a visual codeword dictionary
3. Represent each image as a bag of codewords
4. Compute the "acquaintance coefficient" between every pair of codewords
5. (optional) For the semi-supervised variation, label some distinctive regions for each class
6. For each class, learn a probability matrix and a theme distribution vector using the dependent hierarchical Dirichlet process

**Recognition**

1. Extract local regions from the testing image
2. Represent the testing image as a bag of codewords
3. Categorize the image by Bayesian decision

We have mentioned in Section 2.3 that we also develop an optional semi-supervised step for learning the DHDP algorithm. For the semi-supervised variation, the critical operation is the labeling procedure. We choose a small set of training images from each category. The labeling procedure is as follows. For each image (e.g. Fig.4(a)), we locate a subset of detected local patches that perceptually characterize the object category (e.g. Fig.4(b)). Ideally, for a visual distinctive local region, we should assign one codeword to the region based on its local appearance. But sometimes a shift of a few pixels could result into different codeword assignments for the same general region. To avoid this problem, we first obtain multiple patches (usually 4-5) in the vicinity of the same local region (e.g. Fig.4(c)). For each of these patches, we assign a codeword. We perform the same operation in the same region across the small set of training images to be labeled. Now we have a collection of counts of different codewords that have been assigned to the perceptually similar region. By ranking the frequency of these codewords, we obtain $2 \sim 3$ most frequently occurred codewords for this region. The training procedure of the semi-supervised method then follows Section 2.3



(a) (b) (c)

Figure 4. Examples of the labeling procedure for the face and helicopter categories. For each category (each row), **Column (a)** shows an unlabeled training image. **Column (b)** shows the image with labeled regions. Finally, **Column (c)** shows the several sub regions used for labeling one of the regions in the center column. This figure is best viewed in colors.

In the sampling procedure,$a_0$ in Eq.5 is set as 0.1, and $\gamma$ in Eq.6 is set as 1.The parameter $\lambda$ in Eq.7 is set as 1.2.

We iterate 100 rounds for each patch using MCMC. We will detail the experimental procedures in each of the experiments below (Section 4).

## 4. Experiment and Results

### 4.1. Exp. 1: Caltech 4 datasets

Our first experiment is carried out on the popular Caltech 4 datasets. We do both the DHDP and semi-supervised DHDP experiments on this database. We randomly select 100 training images from each category to construct a dictionary of 1200 codewords. For the semi-supervised DHDP experiment, we label 10 images and get about 10 labeled codewords for each category. We assign the same theme for the 10 codewords in the same class, hence obtaining 4 labeled themes in the training set. For either the DHDP or semi-supervised DHDP experiment, it costs us about 1 hour to train the model. There are 7 themes inferred by the DHDP model and 6 by the semi-supervised DHDP. We show their performance in Table 1, and compare them with [8] and [11] in Fig.5.

| | air | face | leo | motor |
|---|---|---|---|---|
| a. | 96.1/98.1 | 1.6/0 | 2.3/1.9 | 0/0 |
| f. | 0/0 | 97.8/100 | 2.2/0 | 0/0 |
| l. | 0/0 | 0/0 | 100/100 | 0/0 |
| m. | 1.3/0 | 1.6/0 | 0.4/2.0 | 96.7/98.0 |

Table 1. The confusion matrix of DHDP model and semi-supervised DHDP model for the Caltech 4 dataset. The rows denote the ground-truth category label. The columns denote the learnt models. This convention remains the same for all the confusion tables in the rest of this paper. In each lattice, the left value is the performance of DHDP and the right one is from semi-supervised DHDP. We obtain an overall performance of 97.7% by averaging over the diagonal entries for DHDP, and an overall performance of 99.0% for the semi-supervised DHDP.

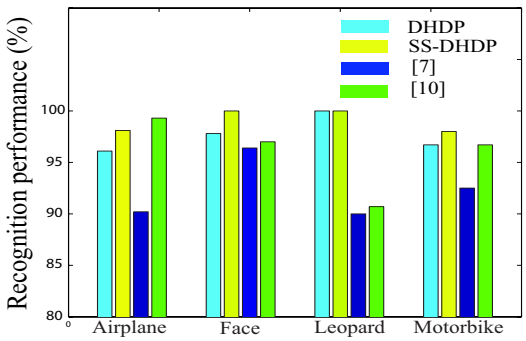



Figure 5. Performance comparison of the DHDP and semi-supervised DHDP models to [8] and [11]. In [8], they achieved performances of 90.2%, 96.4%, 90.0%, 92.5% across the diagonal; in [11], the performances are 99.3%, 97%, 90.7% and 96.7% across the diagonal.

In Fig.6, we show the top five patches that are most strongly linked with other patches in this image using the DHDP model. In Fig.7, we illustrate the latent themes corresponding to the local patches, each different color indicates a different theme. In Fig.8, we compare the theme distribution of DHDP with the "bag of words" model HDP, we see that the DHDP cluster the patches from the same class on one or two themes very well, and different classes are clustered on different themes; while the theme distribution of HDP is more distributed and less discriminative.

Figure 6. The five most dependent patches for each category. Each row represents an object category. This figure is best viewed in color.

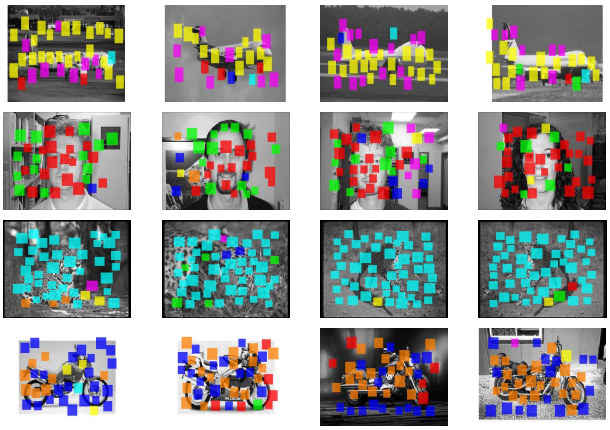


Figure 7. Local patches from different themes are colored coded with DHDP model. We can find each class clustered on one or two themes very well. This figure is best viewed in color.

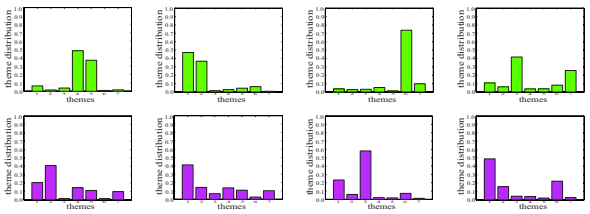



Figure 8. The theme distribution of DHDP and HDP. The top row shows the theme distribution of DHDP on the 4 categories, and the bottom row shows the theme distribution of HDP. We can find DHDP cluster the patches from the same class on one or two themes very well, and different classes are clustered on different themes; while the theme distribution of HDP is not very clustered, and intersects among different categories. So DHDP is quite more discriminative for categorization

### 4.2. Exp. 2: Caltech 101 datasets

In the second experiment, we test our algorithm on a large number of object categories by using the Caltech 101 dataset (approximately 30 $\sim$ 800 images per category). We carry out all experiments here using only the DHDP model without the semi-supervised variation. We randomly select 30 images from each category to construct a dictionary with 2200 codewords. After the training process, the algorithm learns a total of 27 latent themes.

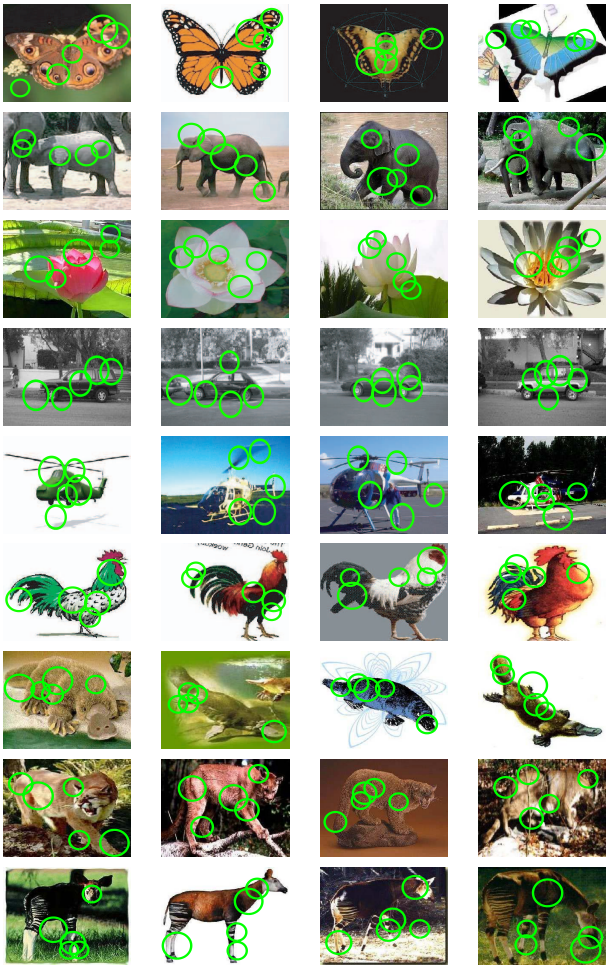


Figure 9. The five most dependent patches for each category. Each row represents an object category. This figure is best viewed in color.

In Fig.9, we show the 5 most dependent patches in the testing images for 9 categories similar to Fig.6. The performance of DHDP is shown in Fig.10 with different number of training images. The number of testing images is fixed to 30 for each category. Fig.10 is the same as that of [25], in which we compare recognition performances of a large number of methods recently obtained on the Caltech 101 dataset ([1, 5, 9, 11, 14, 17, 19, 25]). As it shows, our performance is one of the best reported results using the Caltech 101 dataset. Compared with other methods, the performance of DHDP changes

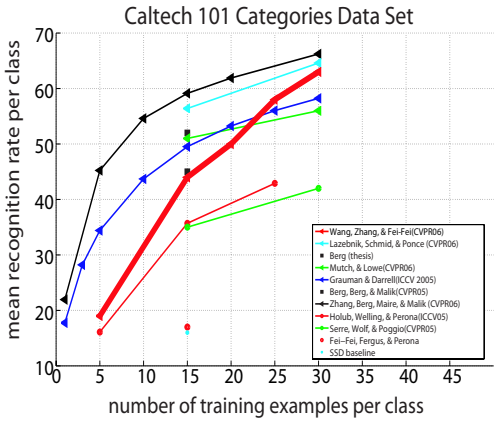



Figure 10. The performances of DHDP and other recent methods ([1, 5, 9, 11, 14, 17, 19, 25]). The figure shows that the DHDP model result is among the best. Compared to other methods, DHDP performances changes more sharply as the number of training images increases. It is possibly due to the fact that our model uses the dependence among patches. When the training number is small, the dependency among the object patches cannot overcome the noise brought by the dependency among background patches. As the training number increases, the useful dependency start to play a main role and the performance becomes much better.

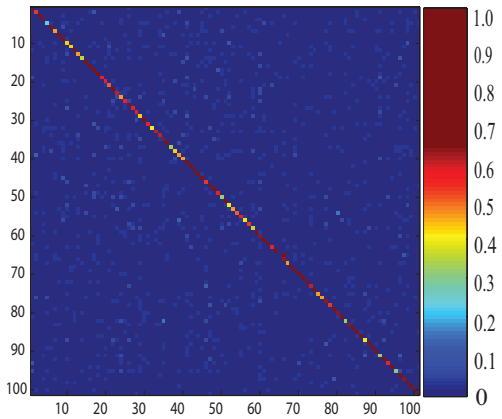
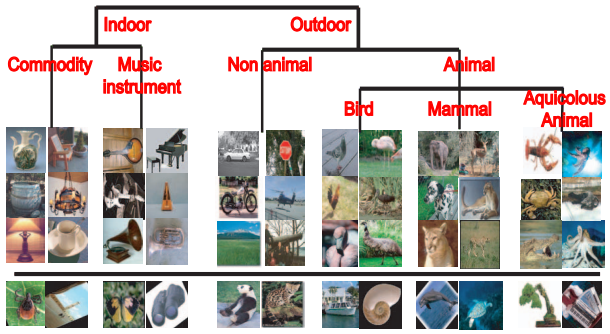


Figure 11. The confusion matrix of the Caltech 101 when using DHDP with 30 training images.

sharply with the number of training images. This is possibly due to the fact that our model uses the dependency among patches. When the training number is small, the dependency among the object patches cannot overcome the noise which is brought by the dependency among background patches. As the training number increases, the usefulness of the dependency structure starts to play a more important role and the performance becomes much better. When the training image number reaches 30, we obtain a performance as high as 63% (see Fig.11 for the confusion table and Fig.10 for comparisons).

Given the theme distribution vectors learnt the 101 categories, we cluster these vectors into two clusters using simple K-means, and find that the two clusters can be named as "indoor" and "outdoor" according to the objects categories they contain. For each cluster, we go

on to partition them until the sub clusters cannot be further partitioned. We name the clusters at each level and construct our taxonomy in Fig.12.




**Commodity**: Wheelchair Chair Chandelier Windsor_chair Ceiling_fan Lamp Umbrella Scissors Watch Headphone Wrench Cup Ewer Barrel Inline_skate Cellphone Pizza Revolver Minaret Menorah Saxophone Anchor Pyramid Scorpion Brain Garfield Starfish Tick Triobite
**Music instrument**: Metronome Mandolin Grand_piano Euphonium Electric_guitar Gramophone Butterfly Binocular Stapler Laptop Camera
**Non animal**: Car-side Helicopter Airplanes Motorbike Cannon Schooner Pagoda Background Lotus Sunflower Straberry Joshua_tree Water_lilly Yin_yang Dollar_Bill Soccer_ball Stop_sign Buddha Faces Sea_horse Okapi Panda Hedgehog Beaver Lobster Wild_cat Stegosauruns
**Bird**: Rooster Flamingo Flanmingo_head Pigeon Ibis Emu Rhino llama Ferry Kangaroo Platypus Nautilus
**Mammal**: Elephant Cougar_body Cougar_face Dalmatian Leopard Dolphin Snoopy Hawksbill Gerenuk
**Aquicolous Animal**: Octopus Crocodile_head Crab Crocodile Crayfish Bonsai Brontosaurus Accordion

Figure 12. **Top.** The taxonomy of Caltech101 obtained by learning a DHDP model. Note that this model is learnt by grayscale pixel values as descriptors and half of each class as training images. **Bottom.** The 101 categories objects are grouped into 6 semantically meaningful super-categories at the leaf level of the taxonomy tree. The gray categories indicate error.

## 5. Conclusion

In this paper, we propose a novel Bayesian model that extends the conventional "bag of words" by utilizing the dependency information among local patches for recognition. We test our algorithm on the Caltech 4 dataset and the Caltech 101 object categories. Our recognition performances are on par with the best performances of these datasets reported to date. Using the latent theme distribution of each category, we can find the semantic similarity among different categories. We therefore construct a hierarchical taxonomy system for the Caltech 101 categories, which discloses the relevancy among the categories. It will be fruitful in the future to explore how such models can be extended to handle larger degrees of variations in the objects.

## References


[1] A.C. Berg, T.L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *Proc. Computer Vision and Pattern Recognition*, 2005.

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

[3] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *Proc. Computer Vision and Pattern Recognition*, 2005.

[4] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, and Bray C. Visual categorization with bags of keypoints. In *Proc. European Conference on Computer Vision*, 2004.

[5] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *Workshop on Generative-Model Based Vision*, 2004.

[6] L. Fei-Fei, R. Fergus, and P. Perona. One-Shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, in press.

[7] L. Fei-Fei and P. Perona. A Bayesian hierarchy model for learning natural scene categories. *CVPR*, 2005.

[8] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. *IJCV(submitted)*, 2005.

[9] Kristen Grauman and Trevor Darrell. Pyramid match kernels: Discriminative classification with sets of image features (version 2). Technical Report CSAIL-TR-2006-020, MIT, 2006.

[10] A. Holub and P. Perona. A discriminative framework for modeling object classes. In *Int. Conf. on Computer Vision*, 2005.

[11] A. Holub, M. Welling, and P. Perona. Combining generative models and fisher kernels for object class recognition. In *CVPR*, 2005.

[12] G.Wu G.Cao. J.Gao, J.Nie. dependence language model for information retrieval. In *SIGIR*, 2004.

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

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

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

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

[17] Jim Mutch and David Lowe. Multiclass object recognition using sparse, localized hmax features. In *CVPR*, 2006.

[18] A. Quattoni, M. Collins, and T. Darrell. Conditional random fields for object recognition. In *NIPS*, 2004.

[19] T. Serre, L. Wolf, and T. Poggio. object recognition with features inspired by visual cortex. In *CVPR*, 2005.

[20] J. Sivic, B.C. Russell, A. Efros, A. Zisserman, and W.T. Freeman. Discovering object categories in image collections. In *Proc. International Conference on Computer Vision*, 2005.

[21] E. Sudderth, A. Torralba, W.T. Freeman, and A. Willsky. Describing visual scenes using transformed dirichlet processes. In *NIPS*, 2005.

[22] Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 2006.

[23] 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.

[24] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proc. European Conference on Computer Vision*, volume 2, pages 101–108, 2000.

[25] Hao Zhang, Alexander C. Berg, Michael Maire, and Jitendra Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, 2006.