Int J Comput Vis
DOI 10.1007/s11263-009-0265-6

# OPTIMOL: Automatic Online Picture Collection via Incremental Model Learning

**Li-Jia Li · Li Fei-Fei**

Received: 21 July 2008 / Accepted: 18 June 2009

**Abstract** The explosion of the Internet provides us with a tremendous resource of images shared online. It also confronts vision researchers the problem of finding effective methods to navigate the vast amount of visual information. Semantic image understanding plays a vital role towards solving this problem. One important task in image understanding is object recognition, in particular, generic object categorization. Critical to this problem are the issues of learning and dataset. Abundant data helps to train a robust recognition system, while a good object classifier can help to collect a large amount of images. This paper presents a novel object recognition algorithm that performs automatic dataset collecting and incremental model learning simultaneously. The goal of this work is to use the tremendous resources of the web to learn robust object category models for detecting and searching for objects in real-world cluttered scenes. Humans contiguously update the knowledge of objects when new examples are observed. Our framework emulates this human learning process by iteratively accumulating model knowledge and image examples. We adapt a non-parametric latent topic model and propose an incremental learning framework. Our algorithm is capable of automatically collecting much larger object category datasets for 22 randomly selected classes from the *Caltech 101* dataset. Furthermore, our system offers not only more images in each object category but also a robust object category model and meaningful image annotation. Our experiments show that OPTIMOL is capable of collecting image datasets that are superior to the well known manually collected object datasets *Caltech 101* and LabelMe.

**Keywords** Internet vision · Image retrieval · Dataset collection · Object recognition · Classification · Annotation · Object localization · Semi-supervised learning · Incremental learning · Generative model · Graphical model · Topic model · Non-parametric model · Markov Chain Monte Carlo sampling · Polysemy

## 1 Introduction

Type the word "airplane" in your favorite Internet search image engine, say Google Image (or Yahoo!, flickr.com, etc.). What do you get? Of the thousands of images these search engines return, only a small fraction would be considered good airplane images ($\sim 15\%$, Fergus et al. 2005b). It is fair to say that for most of today's average users surfing the web for images of generic objects, the current commercial state-of-the-art results are far from satisfying.

This problem is intimately related to the problem of learning and modeling generic object classes in computer vision research (Fei-Fei et al. 2007; Felzenszwalb and Huttenlocher 2005; Fergus et al. 2005a, 2005b; Krempp et al. 2002; LeCun et al. 2004; Leibe and Schiele 2004). In order to develop effective object categorization algorithms, researchers rely on a critical resource: an accurate object class dataset. A good dataset serves as training data as well as an evaluation benchmark. A handful of large scale datasets currently serve such a purpose, such as *Caltech101/256* (Fei-Fei et al. 2004; Griffin et al. 2007), the UIUC car dataset (Agarwal et

L.-J. Li (✉) · L. Fei-Fei
Dept. of Computer Science, Princeton University, Princeton, USA
e-mail: jial@cs.princeton.edu

L. Fei-Fei
e-mail: feifeili@cs.princeton.edu

L. Fei-Fei
Dept. of Computer Science, Stanford University, Stanford, USA
e-mail: feifeili@cs.stanford.edu

Published online: 25 July 2009

al. 2004), LotusHill (Yao et al. 2007), LableMe (Russell et al. 2005) etc. Section 1.1 will elaborate on the strengths and weaknesses of these datasets. In short, all of them, however, have a rather limited number of images and offer no possibility of expansion other than with extremely costly manual labor.

We are therefore facing a chicken and egg problem here: Users of the Internet search engines would like better search results when looking for objects; developers of these search engines would like more robust visual models to improve these results; vision researchers are developing models for this purpose; but in order to do so, it is critical to have large and diverse object datasets for training and evaluation; this, however, goes back to the same problem that the users face.

There are few breakthroughs for this problem recently. Among the solutions, one of the major trends is to manually collect and annotate a ground truth dataset (LotusHill (Yao et al. 2007) and LableMe (Russell et al. 2005)). Due to the vast number of object classes in our world, however, manually collecting images for all the classes is currently impossible. Recently, researchers have developed approaches utilizing images retrieved by image search softwares to learn statistical models to collect datasets automatically. Yet, learning from these images is still challenging:

- Current commercial image retrieval software is built upon text search techniques using the keywords embedded in the image link or tag. Thus, retrieved image is highly contaminated with visually irrelevant images. Extracting the useful information from this noisy pool of retrieved images is quite critical.
- The intra-class appearance variance among images can be large. For example, the appearance of "wrist watches" are different than the "pocket watches" in the watch category. The ability of relying on knowledge extracted from one of them (e.g. "wrist watch") to distinguish the other (e.g. "pocket watch") from unrelated images is important.
- Polysemy is common in the retrieved images, e.g. a "mouse" can be either a "computer mouse" or an "animal mouse". An ideal approach can recognize the different appearances and cluster each of the objects separately.

In this paper, we provide a framework to simultaneously learn object class models and collect object class datasets. This is achieved by leveraging on the vast resource of images available on the Internet. The sketch of our idea is the following. Given a very small number of seed images of an object class (either provided by a human or automatically), our algorithm learns a model that best describes this class. Serving as a classifier, the algorithm can extract from the text search result those images that belong to the object class. The newly collected images are added to the object dataset, serving as new training data to improve the object model. With this new model, the algorithm can then go back to the web and extract more relevant images. Our model uses its previous prediction to teach itself. This is an iterative process that continuously gathers an accurate image dataset while learning a more and more robust object model. We will show in our experiments that our automatic, online algorithm is capable of collecting object class datasets of more images than *Caltech 101* (Fei-Fei et al. 2004) or LabelMe (Russell et al. 2005). To summarize, we highlight here the main contributions of our work.

- We propose an iterative framework that collects object category datasets and learns the object category models simultaneously. This framework uses Bayesian incremental learning as its theoretical base.
- We have developed an incremental learning scheme that uses only the newly added images for training a new model. This memory-less learning scheme is capable of handling an arbitrarily large number of images, which is a vital property for collecting large image datasets.
- Our experiments show that our algorithm is capable of both learning highly effective object category models and collecting object category datasets significantly larger than that of *Caltech 101* or LabelMe.

### 1.1 Related Works

*Image Retrieval from the Web* Content-based image retrieval (CBIR) (Zhou and Huang 2002; Deng 2001; Carson et al. 1999; Li et al. 2000; Chen et al. 2003; Jain and Vailaya 1996; Barnard and Forsyth 2001; Barnard et al. 2003; Jeon et al. 2003) has been long an active field of research. One major group of research (Barnard and Forsyth 2001; Barnard et al. 2003; Jeon et al. 2003) in CBIR treats images as a collection of blobs or blocks, each corresponding to a word or phrase in the caption (with some considerable variations). The task of such algorithms is to assign proper words and/or phrases to a new image, and hence to retrieve similar ones in a database that contains such annotations. Another group of approaches focuses on comparing the query image with exemplar images and retrieving images based on image similarity (Carson et al. 1999; Chen et al. 2003; Deng 2001). However, our work is different from the conventional frameworks of CBIR. Instead of learning to annotate images with a list of words or comparing the similarity of images, our algorithm collects the most suitable images from the web resources given a single word or phrase. One major difference between our work and the traditional CBIR is the emphasis on visual model learning. When collecting images of a particular object category, our algorithm continues to learn a better and better visual model to classify this object.

A few recent approaches in this domain are closer to our current framework. Feng and Chua propose a method to refine images returned by search engine using co-training

(Feng and Chua 2003). They employ two independent segmentation methods as well as two independent sets of features to co-train two "statistically independent" SVM classifiers and co-annotate unknown images. Their method, however, does not offer an incremental training approach to boost the training efficiency in the co-train and co-annotate process. Moreover, their approach needs user interaction at the beginning of training and also when both the classifiers are uncertain about the decision.

Berg and Forsyth (2006) develop a lightly supervised system to collect animal pictures from the web. Their system takes advantage of both the text surrounding the web images and the global feature statistics (patches, colors, textures) of the images to collect a large number of animal images. Their approach involves a training and a testing stage. In the training stage, a set of visual exemplars are selected by clustering the textual information. In the testing stage, textual information as well as visual cues extracted from these visual exemplars are incorporated in the classifier to find more visually and semantically related images. This approach requires supervision to identify the clusters of visual exemplars as relevant or background. In addition to this, there is an optional step for the user to swap erroneously labeled exemplars between the relevant and background topics in training.

Similar to Berg and Forsyth (2006), Schroff et al. (2007) also employ the web meta data to boost the performance of image dataset collection. The images are ranked based on a simple Bayesian posterior estimation, i.e. the probability of the image class given multiple textual features of each image. A visual classifier, trained on the top ranked images, is then applied to re-rank the images.

Another method close in spirit to ours is by Yanai and Barnard (2005). They also utilize the idea of refining web image result with a probabilistic model. Their approach consists of a collection stage and a selection stage. In the collection stage, they divide the images into relevant and unrelated groups by analyzing the associated HTML documents. In the selection stage, a probabilistic generative model is applied to select the most relevant images among those from the first stage. Unlike ours, their method focuses on image annotation. Furthermore, their experiments show that their model is effective for "scene" concepts but not for "object" concepts. Hence, it is not suitable for generic object category dataset collection.

While the three approaches above rely on both visual and textual features of the web images returned by search engines, we would like to focus on visual cue only to demonstrate how much it can improve the retrieval result.

Finally, our approach is inspired by two papers by Fergus et al. (2005b, 2004). They introduce the idea of training a good object class model from web images returned by search engines, hence obtaining an object filter to refine these results. Fergus et al. (2004) extends the constellation model (Weber et al. 2000; Fergus et al. 2003) to include heterogeneous parts (e.g. regions of pixels and curve segments). The extended model is then used to re-rank the retrieved result of image search engine. In Fergus et al. (2005b), the authors extend a latent topic model (pLSA) to incorporate spatial information. The learned model is then applied to classify object images and to re-rank the images retrieved by Google image search. Although these two models are accurate, they are not scalable. Without an incremental learning framework, they need to be re-learned with all available images whenever new images are added.

All the above techniques achieve better search results by using either a better visual model or a combination of visual and text models to re-rank the rather noisy images from the web. We show later that by introducing an iterative framework of incremental learning, we are able to embed the processes of image collection and model learning efficiently into a mutually reinforcing system.

*Object Classification* The recent explosion of object categorization research makes it possible to apply such techniques to partially solve many challenging problems such as improving the image search result and product images organization. Due to the vast number of object categorization approaches (Fei-Fei et al. 2007; Felzenszwalb and Huttenlocher 2005; Fergus et al. 2005a, 2005b; Krempp et al. 2002; LeCun et al. 2004; Leibe and Schiele 2004; Lowe 1999), it is out of the scope of our paper to discuss all of them. Here we will focus on two major branches that are closely related to our approach, specifically, latent topic model based on the "bag of words" representation and incremental learning of statistic models.

A number of systems based on the bag of words model representation have shown to be effective for object and scene recognition (Fergus et al. 2005b; Sivic et al. 2005; Fei-Fei and Perona 2005; Sudderth et al. 2005b; Bosch et al. 2006; Csurka et al. 2004; Sivic and Zisserman 2003). Sivic et al. (2005) apply probabilistic Latent Semantic Analysis (pLSA), a model introduced in the statistical text literature, to images. Treating images and categories as documents and topics respectively, they model an image as a mixture of topics. By discovering the topics embedded in each image, they can find the class of the image. pLSA, however, can not perform satisfactorily on unknown testing images since it is not a well defined generative model. Furthermore, the number of parameters in pLSA grows linearly with the number of training images, making the model prone to overfitting. Fei-Fei and Perona (2005) apply an adapted version of a more flexible model called Latent Dirichlet Allocation (LDA) model (Blei et al. 2003) to natural scene categorization. LDA overcomes problems of pLSA by modeling the topic mixture proportion as a latent variable regularized by its Dirichlet hyper-parameter.

The models mentioned above are all applied in a batch learning scenario. If the training data grows, as in our framework, they have to be retrained with all previous data and the new data. This is not an efficient approach, especially when learning from large datasets. Hence, we would like to apply incremental learning to our model.

A handful of object recognition approaches have applied incremental learning to object recognition tasks. The most notable ones are Krempp et al. (2002) and Fei-Fei et al. (2004). Krempp et al. (2002) use a set of edge configurations as parts, which are learned from the data. By presenting the object categories sequentially to the system, it is optimized to accommodate the new classes by maximally reusing parts. Fei-Fei et al. (2004) adopt a generative probabilistic model called constellation model (Weber et al. 2000; Fergus et al. 2003) to describe the object categories. Following Neal and Hinton's adaptation of conventional EM (Neal and Hinton 1998), a fully Bayesian incremental learning framework is developed to boost the learning speed.

Our approach combines the merits of these two branches:

- Bag of words representation enables the model to handle occlusion and rotation, which are common for web images. It is also computationally efficient, a desired property for the computation of large image dataset. On the other hand, latent topic model provides natural clustering of data, which helps solving the polysemy problem in image retrieval. We choose a nonparametric latent topic model so that the model can adjust its internal structure, specifically the number of clusters of the data, to accommodate new data.
- Given large intra-class variety of the online images, it is difficult to prepare good training examples for every subgroup of each image class. We employ an iteratively learning and classification approach to find the good training examples automatically. In each iteration, the object model is taught by its own prediction. In such iterative process, incremental learning is important to make learning in every iteration more efficient.

*Object Datasets* One main goal of our proposed work is to suggest a framework that can replace most of the current human effort in object dataset collection. A few popular object datasets exist today as the major training and evaluation resources for the community such as *Caltech 101* and LabelMe. *Caltech 101* consists of 101 object classes each of which contains 31 to 800 images (Fei-Fei et al. 2004). It was collected by a group of students spending on average three or four hours per 100 images. While it is regarded as one of the most comprehensive object category datasets now available, it is limited in terms of the variation in the images (big, centered objects with few viewpoint changes), numbers of images per category (at most a few hundred) as well as the number of categories. For a long time, datasets are collected in this way relying on extensive human labor. Similar datasets are Caltech-256 (Griffin et al. 2007), PASCAL (1), LotusHill (Yao et al. 2007) and Fink and Ullman (2007).

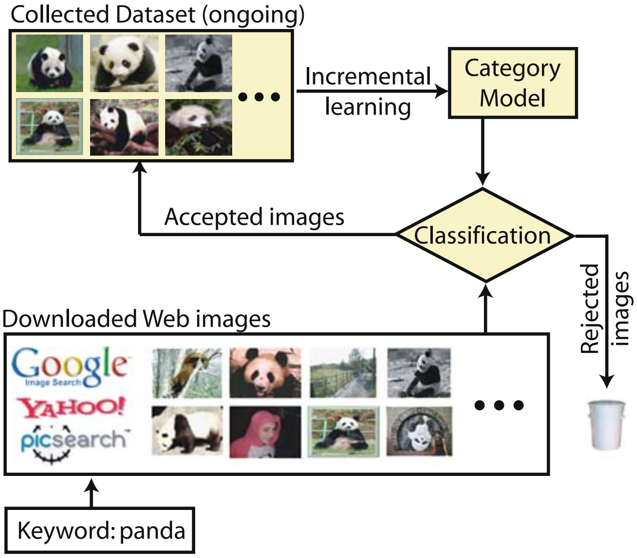

**Fig. 1** Illustration of the framework of the Online Picture collecTion via Incremental MOdel Learning (OPTIMOL) system. This framework works in an incremental way: Once a model is learned, it can be used to classify images from the web resource. The group of images classified as being in this object category are regarded as related images. Otherwise, they are discarded. The model is then updated by a subset of the newly accepted images in the current iteration. In this incremental fashion, the category model gets more and more robust. As a consequence, the collected dataset becomes larger and larger

Recently, LabelMe has offered an alternative way of collecting datasets of objects by having users upload their images and label them (Russell et al. 2005). This dataset is much more diverse than *Caltech 101*, potentially serving as a better benchmark for object detection algorithms. But since it relies on people uploading pictures and making uncontrolled annotations, it is difficult to use it as a generic object dataset. In addition, while some classes have many images (such as 20 304 images for "car"), others have too few (such as 7 images for "watch").

A few other object category datasets such as Agarwal et al. (2004) are also used by researchers. All of the datasets mentioned above require laborious human effort to collect and select the images. In addition, while serving as training and test datasets for researchers, they are not suitable for general search engine users. Our proposed work offers a first step towards a unified way of automatically collecting data useful both as a research dataset as well as for answering user queries.

## 2 General Framework of OPTIMOL

OPTIMOL has two goals to fulfill simultaneously: to automatically collect object datasets from the web and to incrementally learn object category models. We use Fig. 1 and Algorithm 1 to illustrate the overall framework. For every object category we are interested in, say, "panda", we *initialize* our image dataset with a handful of seed images. This can be done either manually or automatically.[1] With this small dataset, we begin the iterative process of model learning and dataset collection. *Learning* is done via an incremental learning process that we introduce in Sect. 3.2.3. Given the current updated model of the object class, we perform a binary *classification* on a subset of images downloaded from the web (e.g. "panda" vs. background). If an image is accepted as a "panda" image based on some statistical criteria (see Sect. 3.2.3), we *augment* our existing "panda" dataset by appending this new image. We then update our "panda" model with a subset of the newly accepted images (see Sect. 3.3.2 for details of the "cache set"). Note that the already existing images in the dataset no longer participate in this iteration of learning. In the meantime, the background model will also be updated using a constant resource of background images.[2] We *repeat* this process till a sufficient dataset is collected or we have exhausted all downloaded images.

**Algorithm 1** Incremental learning, classification and data collection

```
Download from the Web a large reservoir of images obtained by searching with keyword(s)
Initialize the object category dataset with seed images (manually or automatically)
repeat
    Learn object category model with the latest accepted images to the dataset
    Classify a subset of downloaded images using the current object category model
    Augment the dataset with accepted images
until user satisfied or images exhausted
```

## 3 Detailed System of OPTIMOL

### 3.1 Our Model

In this section, we describe the model used in OPTIMOL in detail. Specifically, Sect. 3.1.1 describes the probabilistic classification approaches, especially generative models. Section 3.1.2 introduces briefly the "bag of words" image representation combined with the latent topic model. Finally, in Sect. 3.1.3, we discuss the nonparametric latent topic model (i.e. Hierarchical Dirichlet Process (HDP)) in OPTIMOL.

#### *3.1.1 Generative Model*

Classification approaches can be grossly divided into generative models, discriminative models and discriminant functions (Bishop 2006). For generative models, such as Gaussian mixture models (Weber et al. 2000; Fergus et al. 2003; Fei-Fei et al. 2003), Markov random fields (Pawan Kumar et al. 2005), latent topic model (Sivic et al. 2005; Fei-Fei and Perona 2005; Wang et al. 2006) etc., both the input distribution and the output distribution are modeled. While for discriminative models, which include boosting (Freund and Schapire 1995; Freund and Schapire 1996), support vector machines (Boser et al. 1992), conditional random field (McCallum et al. 2000) etc., the posterior probabilities are modeled directly. The simplest approaches are called discriminant functions (e.g. Fisher's linear discriminant by Belhumeur et al. 1997), which are projections mapping the input data to class labels. Comparing to the other two approaches, generative models are able to handle the missing data problem better since all variables are jointly modeled in a relatively equal manner. When the missing data problem is encountered, the performance will not be affected dramatically. This property is desired for semi-supervised learning from Internet images where only a small amount of labeled data is provided. Here, we would like to adopt generative model given this ideal property for OPTIMOL's iterative incremental learning framework. Previous success of generative model in object recognition (Fei-Fei et al. 2003; Sivic et al. 2005) and content based image retrieval (Yanai and Barnard 2005; Fergus et al. 2005b) ensure the potential ability of generative model in our framework.

#### *3.1.2 Object Category Model*

We would like to emphasize that our proposed framework is not limited to the particular object model used in this paper. Any model that can be cast into an incremental learning framework is suitable for our protocol. Of the many possibilities, we have chosen to use a variant of the HDP (Hierarchical Dirichlet Process) (Teh et al. 2006) model based on the

[1] To automatically collect a handful of seed images, we use the images returned by the first page of Google image search, or any other state-of-the-art commercial search engines given the object class name as query word.

[2] The background class model is learnt by using a published "background" image dataset (Fergus et al. 2003; Fei-Fei et al. 2006). The background class model is updated together with the object class model. In this way, it can accommodate the essential changes of the new training data.

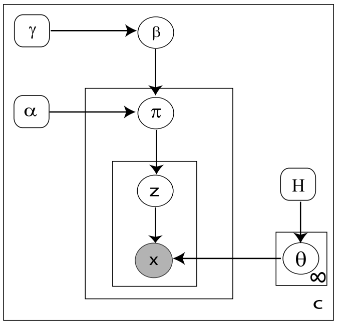

**Fig. 2** Graphical model of HDP. Each node denotes a random variable. *Bounding boxes* represent repetitions. *Arrows* indicate conditional dependency. *Dark node* indicates it is observed

“bag of words” (Sivic et al. 2005; Fei-Fei and Perona 2005; Csurka et al. 2004; Sivic and Zisserman 2003) representation of images. HDP is particular suitable here because of the natural clustering and computationally efficient properties respectively. “Bag of words” model is frequently used in natural language processing and information retrieval of text documents. In “bag of words” model, each document is represented as an unordered collection of words. When applied to image representation, it describes each image as a bag of visual words (node $x$ in Fig. 2). We are particularly interested in the application of latent topic models to such representation (Hofmann 1999; Blei et al. 2003; Teh et al. 2006). Similar to Sudderth et al. (2005a), Wang et al. (2006), we adapt a nonparametric generative model, Hierarchical Dirichlet process (HDP) (Teh et al. 2006), for our object category model. Compared to parametric latent topic models such as LDA (Blei et al. 2003) or pLSA (Hofmann 1999), HDP offers a way to sample an unbounded number of latent topics, or clusters, for each object category model. This property is especially desirable for OPTIMOL. Since the data for training keeps growing, we would like to retain the ability to “grow” the object class model when new clusters of images arise. Before we move on to introduce the HDP object category model in more detail in Sect. 3.1.3, we define the notations in Fig. 2 here.

- A *patch* $x$ is the basic unit of an image. Each patch is represented as a codeword of a visual vocabulary of codewords indexed by $\{1, \dots, T\}$.
- An *image* is represented as $N$ unordered patches denoted by $\mathbf{x} = (x_{j1}, x_{j2}, \dots, x_{jN})$, where $x_{ji}$ is the $i$th patch of the $j$th image.
- A *category* is a collection of $I$ images denoted by $\mathbf{D} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_I)$.

### *3.1.3 Hierarchical Dirichlet Process*

We represent an image as a document constituted by a bag of visual words. Each category consists of a variable number of latent topics corresponding to clusters of image patches with similar visual attributes. We model both object and background classes with HDP (Teh et al. 2006). Figure 2 shows the graphical model of HDP. In the HDP model, $\theta$ corresponds to the distributions of visual words given different latent topics shared among images. Let $x_{ji}$ be the $i$th patch in $j$th image. For each patch $x_{ji}$, there is a hidden variable $z_{ji}$ denoting the latent topic index. $\beta$ is the stick-breaking weights (Sethuraman 1994) and $\pi_j$ represents the mixing proportion of $z$ for the $j$th image. We now go through the graphical model (Fig. 2) and show how we generate each patch in an image. For each image class $c$,

- Sample $\beta \sim \mathrm{GEM}(\gamma)$. GEM is the stick-breaking process:

$$\beta'_k \sim Beta(1, \gamma) \quad \beta_k = \beta'_k \prod_{l=1}^{k-1}(1 - \beta'_l)$$
$$\beta = (\beta_1, \beta_2, \dots, \beta_\infty) \tag{1}$$

- Sample $\theta_k$ from the Dirichlet prior distribution $H$.
- Given the stick-breaking weights $\gamma$ and global cluster $\theta$, we generate each image in this class.
  - We first sample $\pi_j$, $\pi_j|\alpha, \beta \sim \mathrm{DP}(\alpha, \beta)$. DP denotes the Dirichlet Process introduced by Ferguson in 1973 (Ferguson 1973):

$$\pi'_{jk} \sim Beta\left(\alpha\beta_k, \alpha\left(1 - \prod_{l=1}^{k}\beta_l\right)\right)$$
$$\pi_{jk} = \pi'_{jk} \prod_{l=1}^{k-1}(1 - \pi'_{jl}) \tag{2}$$

    where $\pi_j = (\pi_{j1}, \pi_{j1}, \dots, \pi_{j\infty})$.
  - Given $\pi_j$, we are ready to generate each image patch $x_{ji}$
    - Sample the latent topic index $z_{ji}$ for patch $x_{ji}$ from a multinomial distribution $\pi_j$: $z_{ji}|\pi_j \sim \pi_j$
    - Sample $x_{ji}$ given $z_{ji}$ from a class dependent multinomial distribution $F$: $x_{ji}|z_{ji}, \theta_k \sim F(\theta^c_{z_{ji}})$

## 3.2 Learning

We have described our hierarchical model in details. We now turn to learning its parameters. In this subsection, we first describe the batch learning algorithm of our hierarchical model in Sect. 3.2.1. In Sect. 3.2.2, we introduce semi-supervised learning of this model. Finally, the efficient semi-supervised incremental learning is introduced in Sect. 3.2.3 for learning from large image dataset.

### *3.2.1 Markov Chain Monte Carlo Sampling*

In this section, we describe how we learn the parameters by Gibbs sampling (Geman and Geman 1984) of the latent variables. We choose the *Chinese restaurant franchise* (Teh et al. 2006) metaphor to describe this procedure. Imagine multiple Chinese restaurants serving the same set of dishes in the menu. At each table of each restaurant, a dish is shared by the customers sitting at that table. Metaphorically, we describe each image as one restaurant and the local cluster for the customer $x_{ji}$ as the table $t_{ji}$. Similarly, the global cluster for the $t$th table in the $j$th restaurant is represented as the dish $k_{jt}$:

$$t_{ji}|t_{j1},\ldots,t_{ji-1},\alpha,G_0 \sim \sum_{t=1}^{T_j} n_{jt}\delta_{t_{ji}=t} + \alpha G_0 \tag{3}$$

$$k_{jt}|k_{11},k_{12},\ldots,k_{21},\ldots,k_{jt-1}, \quad \gamma \sim \sum_{k=1}^{K} m_k\delta_{k_{jt}=k} + \gamma H \tag{4}$$

where $G_0 \sim DP(\gamma, H)$. $n_{jt}$ denotes the number of customers for table $t$. $T_j$ is the current number of tables. $m_k$ represents the number of tables ordered dish $k$. $K$ denotes the current total number of dishes. All these statistics are calculated without considering the current data point. A new table and new dish can also be generated from $G_0$ and $H$, respectively, if current data does not fit in any of the previous table or dish. For standard mixture models, the *Chinese restaurant franchise* can be easily connected to the stick breaking process by having $z_{ji} = k_{jt}$.

*Sampling the Table* According to (3) and (4), the probability of a new customer $x_{ji}$ being assigned to table $t$ is:

$$P(t_{ji}=t|x_{ji},t_{-ji},\mathbf{k}) \propto \begin{cases} \alpha p_{t_{new}} & \text{for } t = t_{new} \\ n_{jt} f(x_{ji}|\theta_{k_{ji}}) & \text{for used } t \end{cases} \tag{5}$$

We have

$$p_{t_{new}} = \sum_{k=1}^{K} \frac{m_k}{\sum_{k=1}^{K} m_k + \gamma} f(x_{ji}|\theta_{k_{ji}}) + \frac{\gamma}{\sum_{k=1}^{K} m_k + \gamma} f(x_{ji}|\theta_{k_{new}})$$

$f(x_{ji}|\theta_{k_{ji}})$ is the conditional density of patch $x_{ji}$ given all data items associated with $k$ except itself. The probability of assigning a newly generated table $t_{new}$ to a global cluster is proportional to:

$$\begin{cases} m_k f(x_{ji}|\theta_{k_{ji}}) & \text{for used } k \\ \gamma f(x_{ji}|\theta_{k_{new}}) & \text{for new } k \end{cases} \tag{6}$$

*Sampling the Global Latent Topic* For the existing tables, the dish can change according to all customers at that table. The global cluster $k_{jt}$ can be obtained from:

$$\begin{cases} m_k f(\mathbf{x}_{jt}|\theta_{k_{ji}}) & \text{for used } k \\ \gamma f(\mathbf{x}_{jt}|\theta_{k_{new}}) & \text{for new } k \end{cases} \tag{7}$$

Where $\mathbf{x}_{jt}$ represents all patches associated with image level mixture component $t$ in image $j$ except the current one. $f(\mathbf{x}_{jt}|\theta_{k_{jt}})$ is the conditional density of $\mathbf{x}_{jt}$ given all patches associated with topic $k$ except themselves. $n_{jt}$ and $m_k$ will be updated respectively regarding the table index and global latent topic assigned. Given $z_{ji} = k_{jt_{ji}}$, we in turn update $F(\theta^c_{z_{ji}})$ for the category $c$.

### 3.2.2 *Semi-supervised Learning*

Due to the large variation of web images, it requires large number of representative images to train a robust model. Manually selecting these images is time consuming and biased. In the framework of OPTIMOL, we employ a semi-supervised learning approach, specifically self training, to propagate the initial knowledge (Zhu 2006). As a wrapper algorithm, self training can be easily applied to existing models. It has been used in natural language processing to perform tasks such as parsing strings of words (McClosky et al. 2006). In computational biology, self training is employed for gene prediction (Besemer et al. 2001). Recently, it is also applied in computer vision by Rosenberg et al. (2005) to help object detection. All of the approaches show that, by employing a self training framework, one can achieve comparable result to state-of-the-art approach with less labeled training data. We will demonstrate later in Fig. 9 that with semi-supervised learning framework, OPTIMOL shows superior performance in comparison to the fully supervised learning framework using the same number of seed images. The basic idea of self training is:

- First, an initial model is trained with a limited amount of reliable labeled data.
- This model is applied to estimate the labels of the unlabeled data.
- The estimated labels is used to retrain the model.
- Repeat the training and classification procedure.

With this idea, self training allows the model to teach itself iteratively with new classification results. In each iteration of the self training, one can incorporate the new data to retrain the model either via the batch learning mode described in Sect. 3.2.1 or an incremental learning mode introduced later in Sect. 3.2.3. In the self training framework, data that are far away from the initial training set are unlikely to be selected to update the model. However, such data are very useful for generalization of the model. Thus, we design a "cache set" to solve this problem in Sect. 3.3.2.

### 3.2.3 *Incremental Learning of a Latent Topic Model*

Having introduced the object class model and the batched learning approach, we propose an incremental learning scheme for OPTIMOL. This scheme let OPTIMOL update the model at every iteration of the dataset collection process more efficiently. Our goal here is to perform incremental learning by using only new images selected at current iteration. We will illustrate in Fig. 9 (Middle) that this is much more efficient than performing a batch learning by using all images in the current dataset at every iteration. Meanwhile, it still retains the accuracy of the model as shown in Fig. 9. Let $\Theta$ denote the model parameters, and $I_j$ denote the $j$th image represented by a set of patches $x_{j1}, \ldots, x_{jn}$. For each patch $x_{ji}$, there is a hidden variable $z_{ji}$ denoting the latent topic index. The model parameters and hidden variable are updated iteratively using the current model and the input image $I_j$ in the following fashion:

$$z_j \sim p(z|\Theta^{j-1}, I_j) \quad \Theta^j \sim p(\Theta|z_j, \Theta^{j-1}, I_j) \tag{8}$$

where $\Theta^{j-1}$ represents the model parameters learned from the previous $j-1$ images. Neal and Hinton (1998) provide a theoretical ground for incrementally learning mixture models via sufficient statistics updates. We follow this idea by keeping only the sufficient statistics of the parameters associated with the existing images in an object dataset. Learning is then achieved by updating these sufficient statistics with those provided by the new images. One straightforward method is to use all the new images accepted by the current classification criterion. But this method will favor those images with similar appearances to the existing ones, hence resulting in over-specialized object models. To avoid such a problem, we take full advantage of the non-parametric HDP model by using a subset of the related images denoted as "cache set" to update our model. Here, "related images" refers to images classified as belonging to the object class by the current model. We detail the selection of the "cache set" in Sect. 3.3.2.

## 3.3 New Image Classification and Annotation

In the OPTIMOL framework, learning and classification are conducted iteratively. We have described the learning step in Sect. 3.2. In this subsection, we introduce the classification step in our framework. In Sect. 3.3.1, we describe how our model judges which images are related images against others. Sect. 3.3.2 describes the criterion to select the "cache set", a subset of the related images to be used to train our model. Finally, we introduce the annotation method in Sect. 3.3.3.

### 3.3.1 *Image Classification*

For every iteration of the dataset collection process, we have a binary classification problem: classify unknown images as a foreground object or a background image. In the current model, we have $p(z|c)$ parameterized by the distribution of global latent topics given each class in the Chinese restaurant franchise and $p(x|z,c)$ parameterized by $F(\theta_z^c)$ learned for each category $c$. A testing image $I$ is represented as a collection of local patches $x_i$, where $i = \{1, \ldots, M\}$ and $M$ is the number of patches. The likelihood $p(I|c)$ for each class is calculated by:

$$P(I|c) = \prod_i \sum_z P(x_i|z,c) P(z|c) \tag{9}$$

Classification decision is made by choosing the category model that yields the higher probability. From a dataset collection point of view, incorporating an incorrect image into the dataset (false positive) is much worse than missing a correct image (false negative). Hence, a risk function is introduced to penalize false positives more heavily:

$$\begin{aligned} R_i(A|I) &= \lambda_{Ac_f} P(c_f|I) + \lambda_{Ac_b} P(c_b|I) \\ R_i(R|I) &= \lambda_{Rc_f} P(c_f|I) + \lambda_{Rc_b} P(c_b|I) \end{aligned} \tag{10}$$

Here $A$ represents acceptance of an image into our dataset. $R$ denotes rejection. As long as the risk of accepting an image is lower than rejecting it, it is accepted. Image classification is finally decided by the likelihood ratio:

$$\frac{P(I|c_f)}{P(I|c_b)} > \frac{\lambda_{Ac_b} - \lambda_{Rc_b}}{\lambda_{Rc_f} - \lambda_{Ac_f}} \frac{P(c_b)}{P(c_f)} \tag{11}$$

where the $c_f$ is the foreground category while the $c_b$ is the background category. $\frac{\lambda_{Ac_b} - \lambda_{Rc_b}}{\lambda_{Rc_f} - \lambda_{Ac_f}}$ is automatically adjusted by applying the likelihood ratio measurement to a reference dataset[3] at every iteration. New images satisfying (11) are regarded as related images. They will be either appended to the permanent dataset or used to train the new model upon further criterion.

### 3.3.2 *The Cache Set*

In the self training setting, the model teaches itself by using the predicted related images. It is critical to distinguish random noisy images from difference caused by intra-class difference. How to extract the most useful information from the new classification result automatically? We use a "cache set" of images to incrementally update our model. The "cache set" is a less "permanent" set of good images compared to the actual image dataset. At each iteration, if all "good" images are used for model learning, it is highly likely that many of these images will look very similar to the previously collected images, hence reinforcing the model to be even more

[3] To achieve a fully automated system, we use the original seed images as the reference dataset. As the training dataset grows larger, the direct effect of the original training images diminishes in terms of the object model. They therefore become good approximation of a validation dataset.

specialized in selecting such images for the next iteration. Furthermore, it will also be computationally expensive to train with all "good" images. So the usage of the "cache set" is to retain a group of images that tend to be more diverse than the existing images in the dataset. For each new image passing the classification criterion (11), it is further evaluated by (12) to determine whether it should belong to the "cache set" or the permanent set.

$$H(I) = -\sum_{z} p(z|I) \ln p(z|I) \tag{12}$$

In Fig. 14, we demonstrate how to select the "cache set". According to Shannon's definition of entropy, (12) relates to the amount of uncertainty of an event associated with a given probability distribution. Images with high entropy are more uncertain and more likely to have new topics. Thus, these high likelihood and high entropy images are ideal for model learning. In the meantime, images with high likelihood but low entropy are regarded as confident foreground images and will be incorporated into the permanent dataset.

### 3.3.3 *Image Annotation*

The goal of OPTIMOL is not only to collect a good image dataset but also to provide further information about the location and size of the objects contained in the dataset images. Object annotation is carried out by first calculating the likelihood of each patch given the object class $c_f$:

$$p(x|c_f) = \sum_{z} p(x|z, c_f)\, p(z|c_f) \tag{13}$$

The region with the most concentrated high likelihood patches is then selected as the object region. A bounding box is drawn to enclose the selected patches according to (13). Sample results are shown in Fig. 20.

## 3.4 Discussion of the Model

We discuss here several important properties of OPTIMOL in this section.

*Dataset Diversity* Our goal is to collect a diverse image dataset which has ample intra class variation. Furthermore, the ideal model should be capable of collecting all possible object classes associated with different semantic meanings of a polysemous word. OPTIMOL is able to achieve both goals given its facility of accommodating new training data different from the previous ones. This is largely attributed to the property of object model (i.e. HDP) that is capable of generating unbounded number of topics to describe data with different aspects. Later, we show in Fig. 10 that our framework can collect a larger and more diverse image dataset compared to *Caltech 101*. Moreover, Fig. 11 demonstrates that OPTIMOL collects image in a semantic way by assigning visually different images to different clusters.

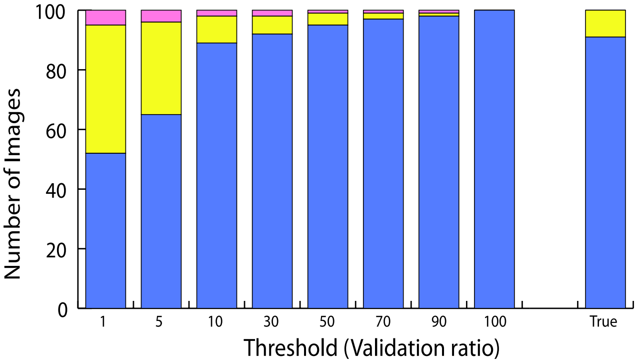

**Fig. 3** (Color online) Influence of the threshold (11) on the number of images to be appended to the dataset and held in the cache set on 100 "accordion" images. $x$-axis is the value of threshold represented in percentile. The validation ratio thresholds are 1, 5, 10, 30, 50, 70, 90, 100, which are equivalent to −1.94, 2.10, 14.24, 26.88, 44.26, 58.50, 100.22 and 112.46 in log likelihood ratio thresholds respectively for the current classifier. $y$-axis denotes the number of images. *Blue region* represents number of images classified as unrelated. *Yellow region* denotes the number of images that will be appended to the object dataset. *Pink region* represents number of images held in the "cache set". The "true" bar represents the proportion of true images in the 100 testing images. These bars are generated using the initial model learned from 15 seeds images. The higher the threshold is, the fewer number of images will be appended to the permanent dataset and held in the "cache set"

*Concept Drift* Self training helps OPTIMOL to accumulate knowledge without human interaction. However, it is prone to concept drift when the model is updated by unrelated images. The term "Concept Drift" refers to the phenomenon of a target variable changing over time. In the OPTIMOL framework, we are mostly concerned with the object model drifting from one category to another (e.g. from accordions to grand pianos). To avoid model drift, our system needs to decide whether an image should be discarded, appended to the permanent dataset or kept in the "cache set" to retrain the model. Using a constant number threshold for (11) to make this decision is not feasible since the model is updated in every iteration. The rank of the images is not a good choice either since there might not be any related images in current iteration. In the OPTIMOL framework, we use a threshold calculated dynamically by measuring the likelihood ratio of the updated model on a validation set. Those with likelihood ratio lower than the threshold are assumed to be "unrelated" and hence discarded. Among those "related" images, a proportion of images with high entropies are selected to be held in the "cache set" according to (12). Figure 3 shows the influence of the threshold on the number of images to be accepted, discarded or used for training. Basically, the number of images to be incorporated into the permanent dataset decreases along with the increase of the threshold. The same applies to the number of images to be held in the "cache set". If fewer images are kept in the "cache set" and are used to update the model, the model

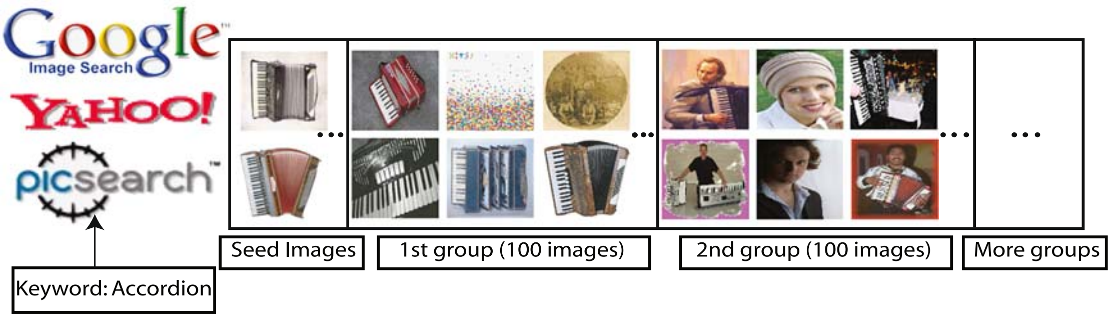

**Fig. 4** Downloading part in Step 1. A noisy "accordion" dataset is downloaded using "accordion" as query word in Google image, Yahoo! image and Picsearch. Downloaded images will be further divided into groups for the iterative process

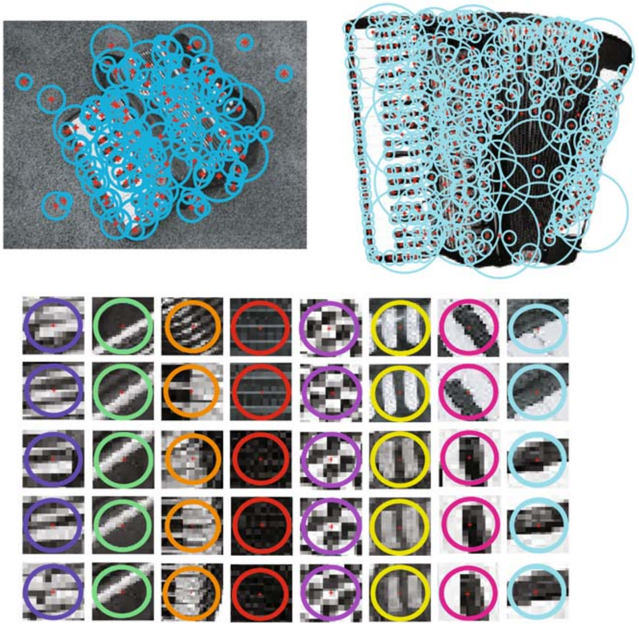

**Fig. 5** (Color online) Preprocessing in Step 1. *Top*: Regions of interest found by Kadir&Brady detector. The *circles* indicate the interest regions. The *red crosses* are the centers of these regions. *Bottom:* Sample codewords. Patches with similar SIFT descriptors are clustered into the same codeword, which are presented using the same color

tends to be similar to the initial model. In the extreme case, if the threshold equals 1, no image will be incorporated to the dataset. Neither will new images be used to retrain the model. The model will stay the same as the initial model. Hence the incorporated images will be highly similar to the initial dataset and the collected dataset will not be very diverse (Fig. 15, left). To the other extreme, if the threshold equals 0, unrelated images will be accepted. Some of these unrelated images with high entropies will be used to update the model. In this scenario, self training will reinforce the error in each iteration and tend to drift. We demonstrate this concept drift issue by showing the dataset collection performance of OPTIMOL with different likelihood ratio thresholds in Fig. 13.

## 4 Walkthrough for the Accordion Category

As an example, we describe how OPTIMOL collects images for the "accordion" category following Algorithm 1 and Fig. 1. We use Figs. 4–8 to show the real system.

- Step 1 (Downloading and preprocessing): As shown in Fig. 4, 1659 images are downloaded as our image pool by typing the query word "accordion" in image search engines such as Google image, Yahoo image and Picsearch. We use the first 15 images from the web resource as our seed images, assuming that most of them are related to the "accordion" concept. The remaining (non-seed) images are divided into 17 groups. The first 16 groups have 100 images each and the last, 17th group has 44 images. The OPTIMOL framework will process one group per iteration. Each image is represented as a set of unordered local patches. Kadir and Brady (2001) salient point detector offers compact representations of the image, which makes computation more efficient for our framework. We apply this detector to find the informative local regions that are salient over both location and scale. Considering the diversity of images on the web, a 128-dim rotationally invariant SIFT vector is used to represent each region (Lowe 1999). We build a 500-word codebook by applying K-means clustering to the 89058 SIFT vectors extracted from the 15 seeds images of each of the 23 object categories. Each patch in an image is then described by using the most similar codeword in the codebook via vector quantization. In Fig. 5, we show examples of detected regions of interest and some codeword samples.
- Step 2 (Initial batch learning): As shown in Fig. 6, a batch learning algorithm described in Sect. 3.2.1 is applied on the seed images to train an initial "accordion" model. Meanwhile, same number of background images are used to trains a background model. In model learning, the hyper-parameters $\gamma$ and $\alpha$ are constant numbers 1 and 0.01 respectively acting as smooth factors. We will show later in Fig. 12 how they influence the

model. According to (4), given a high $\gamma$ value, we expect to obtain more dishes (global topics in a category) in the model. Similarly, following (3), a higher $\alpha$ value populates more tables (local clusters in each image). After Step 2, we obtain a fairly good object category model, which can perform reasonable classification.

- Step 3 (Classification): Models obtained from the learning step are used to classify a group of images from the pool, typically 100 images. By measuring the likelihood ratio, this group is divided into unrelated images and related images as shown in Fig. 7. The unrelated images will be discarded. Related images are further measured by their entropy. High entropy ones will be held in the "cache set". Low entropy images will be appended to the permanent dataset. In classification, our threshold is selected as "30%". This is a conservative choice that allows only the top 30% validation images with highest likelihood ratio to be classified as foreground images. This threshold is equivalent to likelihood ratio threshold 26.88. As shown in Fig. 3, this criterion agrees (conservatively) with the observation that an average estimate of 15% of images returned by the search engine are related to the query word(s). 10% of the related images with high entropies will be kept in the "cache set" for the incremental learning of the model. These images also participate in next 2 iterations in classification. After three iterations, images still left in the "cache set" will be discarded.

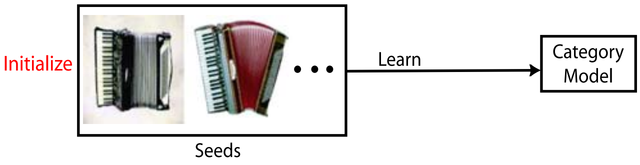

**Fig. 6** Initial batch learning. In the first iteration, a model is learned from the seed images. The learned model performs fairly well in classification as shown in Fig. 9

- Step 4 (Incremental Learning) As shown in Fig. 8, incremental learning is only applied to images held in the "cache set". In the meantime, the same number of new background images are used to update the background model. In this step, we keep the same set of learning parameters as those in Step 2.
- Repeat Step 3 and 4 till the user terminates the program or images in the downloaded image pool are exhausted.

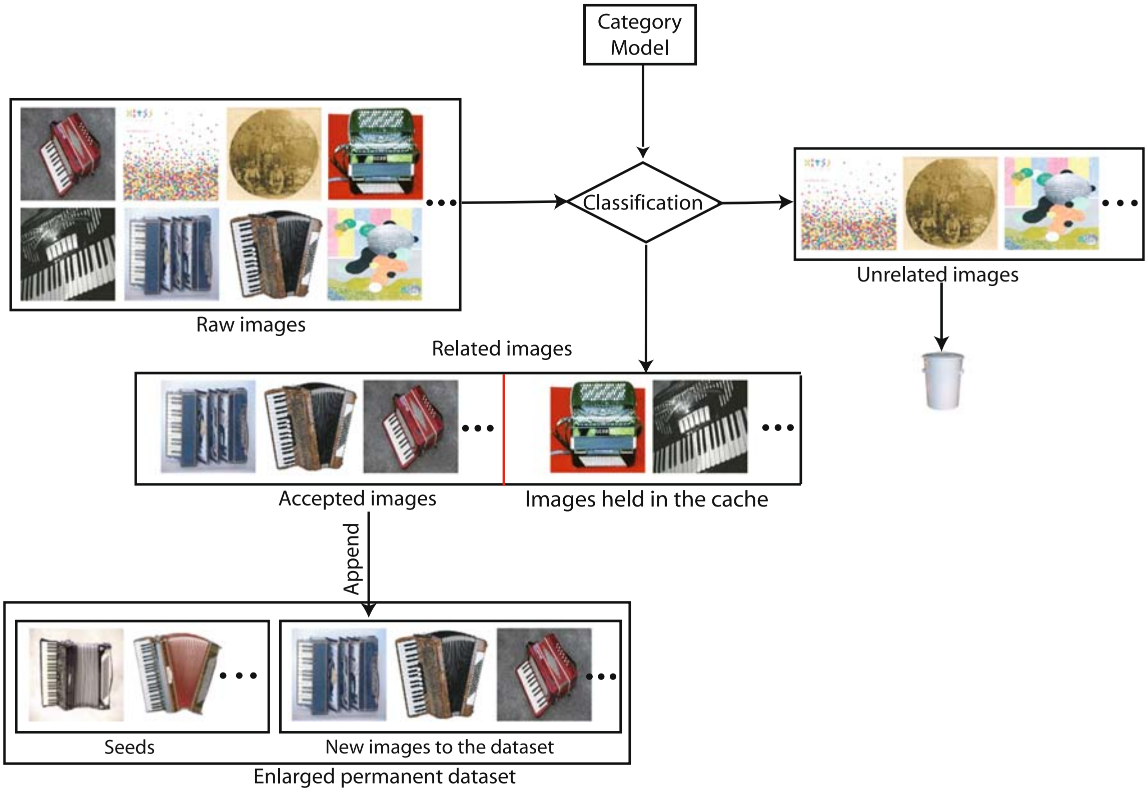

**Fig. 7** Classification. Classification is performed on a subset of raw images collected from the web using the "accordion" query. Images with low likelihood ratios measured by (11) are discarded. For the rest of the images, those with low entropies are incorporated into the permanent dataset, while the high entropy ones stay in the "cache set"

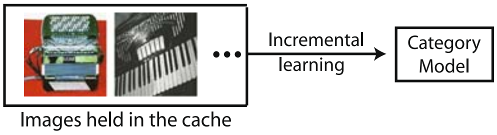

**Fig. 8** Incremental Learning. The model is updated using only the images held in the "cache set" from the previous iteration

## 5 Experiments & Results

We conduct three experiments to demonstrate the effectiveness of OPTIMOL. Experiment 1 consists of a set of analysis experiments.

- A performance comparison of the batch vs. incremental learning methods in terms of the number of collected images, processing time and the recognition accuracy.
- Diversity of our collected dataset. More specifically, comparison between the average images of our collected dataset and the *Caltech 101* dataset. In addition, we show average images of different clusters for the "accordion" and "euphonium" categories as examples to provide more insights into the model.
- Influence of the hyper-parameters $\gamma$ and $\alpha$ on the model. $\gamma$ and $\alpha$ control the number of global and local clusters of the images respectively.
- Dataset collection comparison of OPTIMOL by using different likelihood threshold values to demonstrate the issue of concept drift.
- Illustration of images in the permanent dataset, the "cache set" and the "junk set".
- Illustration of images collected by OPTIMOL using different entropy thresholds.
- Detection performance comparison of OPTIMOL by using different numbers of seed images.
- Polysemous class analysis (a case study of the polysemous words "mouse" and "bass").

Experiment 2 demonstrates the superior dataset collection performance of OPTIMOL over the existing datasets. In addition to dataset collect, it also provides satisfying annotations on the collected images. Experiment 3 shows that OPTIMOL is on par with the state-of-the-art object model learned from Internet images (Fergus et al. 2005b) for multiple object categories classification.

We first introduce the various datasets used in the experiments. Then we show experiment settings and results for the three experiments respectively.

### 5.1 Datasets Definitions

We define the following four different datasets used in our experiments:

1. Caltech 101-Web & Caltech 101-Human
   Two versions of the Caltech 101 dataset are used in our experiment. Caltech 101-Web is the original raw dataset downloaded from the web containing a large portion of visually unrelated images in each category. The number of images in each category varies from 113 (winsor-chair) to 1701 (watch). Caltech 101-Human is the clean dataset manually selected from Caltech 101-Web. The number of images in each category varies from 31 (inline-skate) to 800 (airplanes). By using this dataset, we show that OPTIMOL achieves superior retrieval performance to human labeled results.
2. Web-23
   We downloaded 21 object categories from online image search engines by using query words randomly selected from object category names in Caltech 101-Web. In addition, "face" and "penguin" categories are included in Web-23 for further comparison. The number of images in each category ranges from 577 (stop-sign) to 12414 (face). Most of the images in a category are unrelated images (e.g. 352 true "accordions" out of 1659 images).
3. Princeton-23 (Collins et al. 2008)
   This dataset includes the same categories as used in Web-23. However, it is a more diverse dataset which contains more images in every category. The images are downloaded using words generated by WordNet (Miller 1995) synset as the query input for image search engines. To obtain more images, query words are also translated into multiple languages, accessing the regional website of the image search engines. The number of images in each category varies from 4854 (inline-skate) to 38937 (sunflower).
4. Fergus ICCV'05 dataset
   A 7-Category dataset provided by (Fergus et al. 2005b). Object classes are: airplane, car, face, guitar, leopard, motorbike and watch.

### 5.2 Experiment 1: Analysis Experiment

*Comparison of Incremental Learning and Batch Learning* In this experiment, we compare the computation time and accuracy of the incremental learning algorithm, the batch learning algorithm as well as the base model learned from initial seed images (Fig. 9). To keep a fair comparison, the images used for batch learning and incremental learning are exactly the same in each iteration. For all three algorithms, background models are updated together with the foreground models by using the same number of images. All results shown here are collected from the "inline skate" dataset; other datasets yield similar behavior. Fig. 9 (left) shows that the incremental learning method is comparable to the batch method in the number of collected images. Both of them outperform the base model learned from the seed

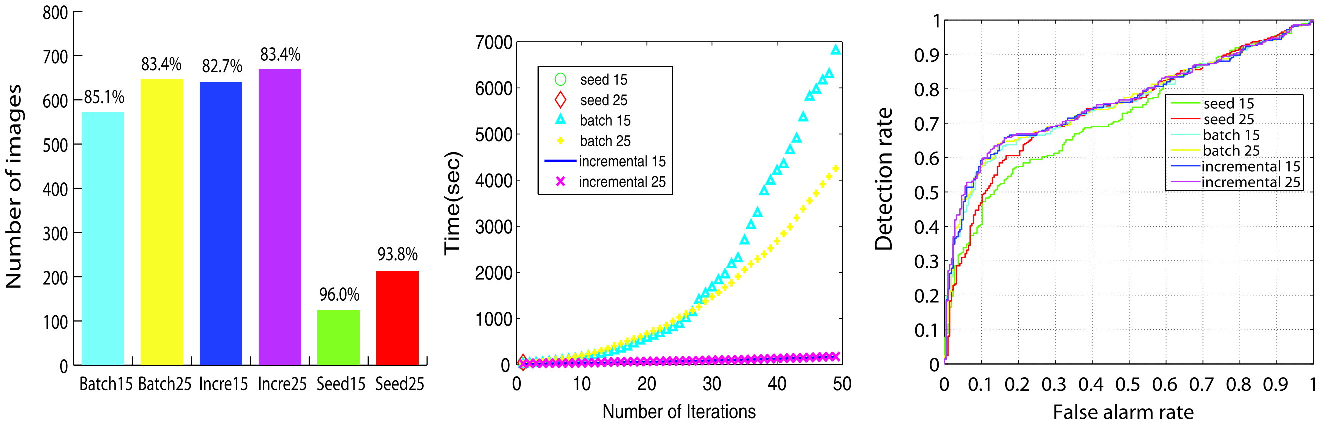

**Fig. 9** Batch vs. Incremental Learning (a case study of the "inline skate" category with 4835 images). *Left:* the number of images retrieved by the incremental learning algorithm, the batch learning algorithm and the base model. Detection rate is displayed on top of each bar. $x$-axis represents batch learning with 15 seed images, batch learning with 25 seed images, incremental learning with 15 seed images, incremental learning with 25 seed images, the base model learned from 15 seed images and 25 seed images respectively. *Middle:* Running time comparison of the batch learning method, the incremental learning method and the base model learned as a function of number of training iterations. The incrementally learned model is initialized by applying the batch learning algorithm on 15 or 25 training images, which takes the same amount of time as the corresponding batch method does. After initialization, incremental learning is more efficient compared to the batch method. *Right*: Recognition accuracy of the incrementally learned, batch learned models and the base model evaluated by using Receiver Operating Characteristic (ROC) Curves

images only. Fig. 9 (middle) illustrates that by incrementally learning from the new images at every iteration, OPTIMOL is more computationally efficient than a batch method. Finally, we show a classification performance comparison among OPTIMOL, the batch method and the model learned from seed images in Fig. 9 (right) by a set of ROC curves.

In our system, the image classifier evolves as the model gets updated in every iteration of the self training process. In the image dataset collection process, the newly updated classifier categorizes the current group of images into foreground images and background images. The testing images are therefore different in each iteration of the self training process. Evaluating different classifiers on different test image sets respectively provide little useful information of the classifier quality. A good classifier could perform poorly on a challenging dataset while a poor classifier might perform satisfactorily on a simple dataset. Thus, we only compare our model at the end of the incremental learning process with a model that is learned in a batch mode by testing both models on the same set of test images. We use an ROC curve to illustrate the classification result for each model, shown in Fig. 9. Classifier quality is measured by the area under its ROC curve. As demonstrated in Fig. 9 (right), while batch learning and incremental approaches are comparable to each other in classification, both of them show superior performance over the base models trained by seed images only. In addition, Fig. 9 (left) and Fig. 9 (right) show that the number of seed images has little influence on the performances of the iterative approaches. This can be easily explained by the property of self training which teaches the model automatically by using the predicted result. Once a decent initial model is learned, self training can use the correct detection to update the model. This is equivalent to feeding the model manually with more images.

*Diversity Analysis* In Fig. 10, we show the average image of each category collected by OPTIMOL comparing with those of *Caltech101*. We also illustrate images collected by OPTIMOL from the Caltech 101-web and Web-23 datasets online.[4] We observe that images collected by OPTIMOL exhibit a much larger degree of diversity than those in *Caltech101*.

Furthermore, we use "accordion" and "euphonium" categories as examples to demonstrate the learned internal structure of our dataset in Fig. 11. Figure 11 demonstrates how our model clusters the images. The average image at the top of each tree is very gray indicating that our collected dataset is highly diverse. The middle layer shows the average images of different clusters in this dataset. Attached to these average images are the example images within each cluster. Each of the clusters exhibits unique pattern whereas the root of each tree demonstrates a combination of these patterns. Here only the three clusters with most images are shown. Theoretically, the system can have unbounded number of clusters given the property of HDP.

*Hyper-Parameter Analysis* The concentration parameter $\gamma$ controls the number of global clusters shared within each

[4] http://vision.stanford.edu/projects/OPTIMOL/main/main.html#Dataset.

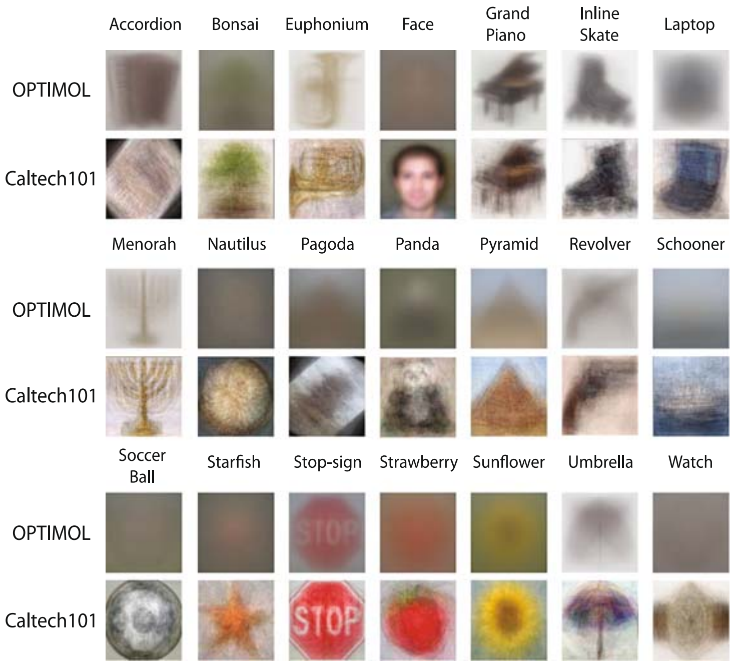

**Fig. 10** Average image of each category, in comparison to the average images of *Caltech101*. The grayer the average image is, the more diverse the dataset is

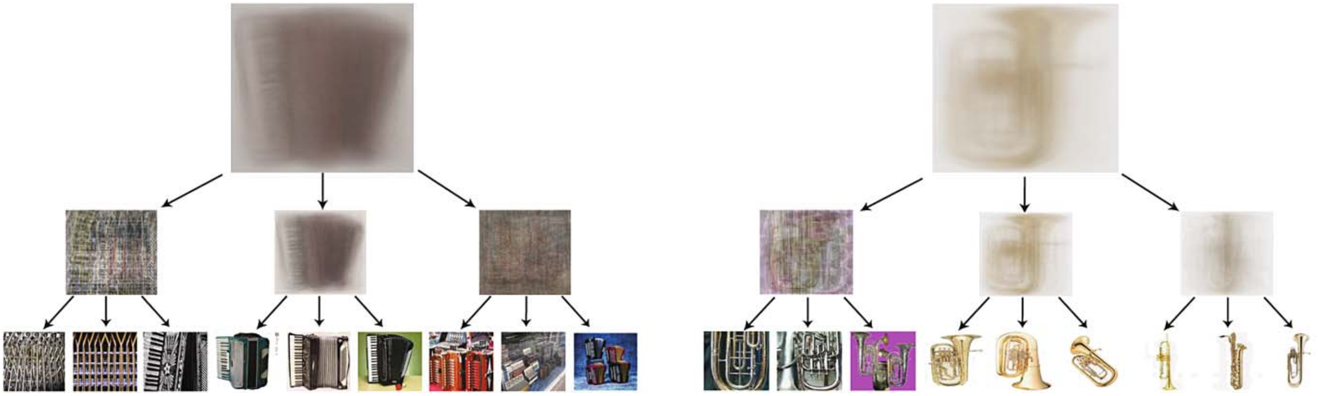

**Fig. 11** Diversity of our collected dataset. *Left:* Illustration of the diverse clusters in the "accordion" dataset. Root image is the average image of all the images in "accordion" dataset collected by OPTIMOL. Middle layers are average images of the top 3 "accordion" clusters generated from the learned model. Leaf nodes of the tree structure are 3 example images attached to each of the cluster average image. *Right:* Illustration of the diverse clusters in the "euphonium" dataset

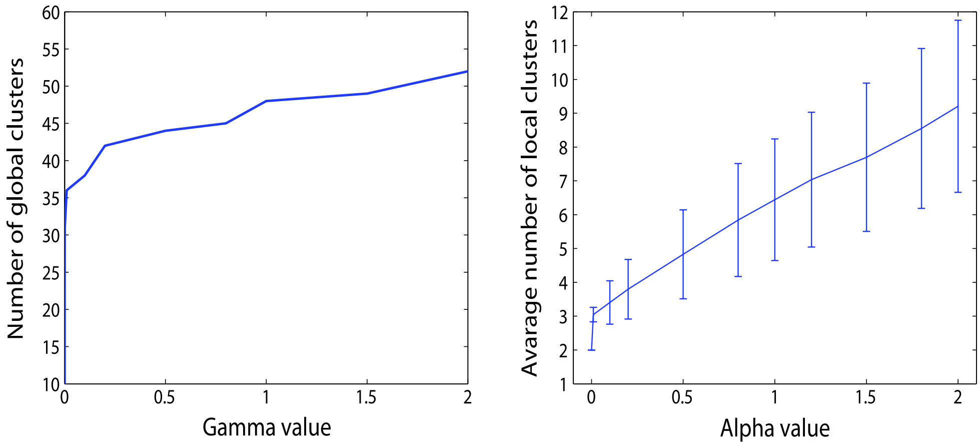

**Fig. 12** *Left*: Number of global clusters shared within each category as a function of the value of $\gamma$. Please refer to Fig. 2, Sects. 3.1.3 and 3.2.1 for detailed description of $\gamma$. *Right:* Average number of clusters in each image as a function of the value of $\alpha$. The standard deviation of the average numbers are plotted as *vertical bars* centered at the data points. Please refer to Fig. 2 and Sect. 3.2.1 for detailed description of $\alpha$

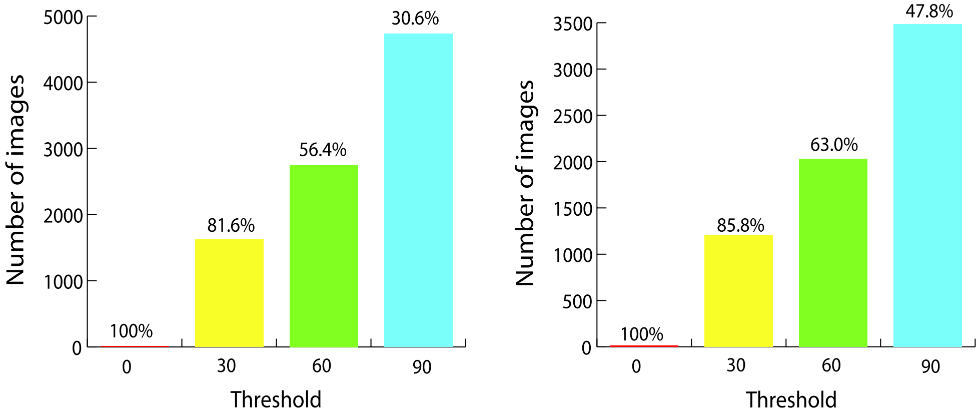

**Fig. 13** *Left*: Data collection results of OPTIMOL with different likelihood ratio threshold values for the "accordion" dataset. $x$-axis denotes the likelihood ratio threshold values in percentile. $y$-axis represents the number of collected images. The number on the top of each bar represents the detection rate for OPTIMOL with that entropy threshold value. *Right*: Data collection results of OPTIMOL with different likelihood ratio threshold values for the "euphonium" dataset

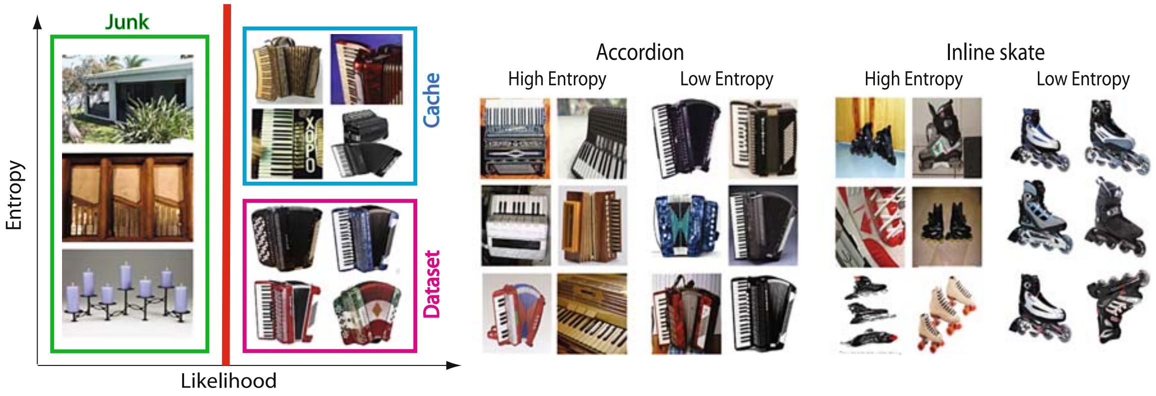

**Fig. 14** *Left*: Illustration of images in the permanent dataset, the "cache set" and the "junk set". $x$-axis represents the likelihood while $y$-axis represents the entropy. If the likelihood ratio of an image is higher than some threshold, it is selected as a related image. This image will be further measured by its entropy. If the image has low entropy, it will be appended to the permanent dataset. If it has high entropy, it will stay in the "cache set" to be further used to train the model. *Right*: Examples of high and low entropy images in "accordion" and "inline-skate" classes

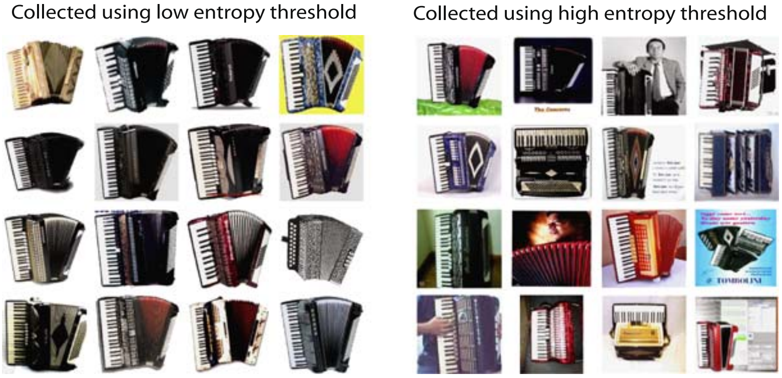
Collected using low entropy threshold
Collected using high entropy threshold

**Fig. 15** Sampled images from dataset collected by using different entropy threshold values. *Left*: Example images from dataset collected with entropy threshold set at top 100% (all images). *Right*: Example images from dataset collected with entropy threshold set at top 30%

class. $\alpha$ influences the number of local clusters in each image. We demonstrate the influence of these hyper parameters on the number of global and local clusters in the "accordion" class in Fig. 12. In Fig. 12 (left), we show that as the value of $\gamma$ increases, the number of global clusters estimated by the model increases too. The average number of local clusters in each image increases when $\alpha$ increases (Fig. 12 (right)).

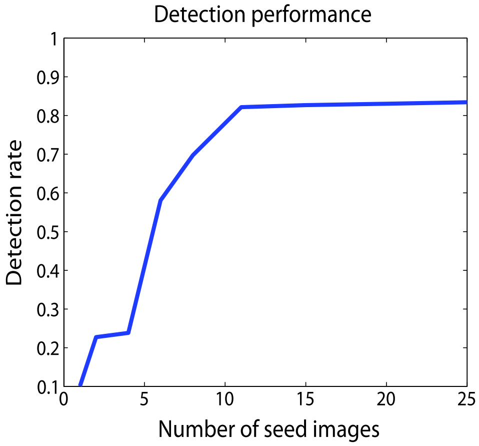

**Fig. 16** Detection performance. $x$-axis is the number of seed images. $y$-axis represents the detection rate

*Concept Drift Analysis* We have discussed in Sect. 3.4 that "concept drift" is the phenomenon of object model drifting from one category to another. This will result in degraded recognition accuracy. To demonstrate the issue of "concept drift", we compare the dataset collection performance of OPTIMOL by using different likelihood ratio threshold values. We present results of the "accordion" and "euphonium" datasets where likelihood ratio thresholds are set at 0, 30, 60, 90 percentile respectively in Fig. 13. Our experiment shows that a tight likelihood threshold allows fewer images to be classified as the foreground images with less false positives but more misses. A low likelihood threshold can help OPTIMOL to collect more images. But it introduces relatively more false positives hence leads to concept drift.

*Illustration of Images in the Permanent Dataset, the "Cache Set" and the "Junk Set"* We have discussed in Sect. 3.3.2

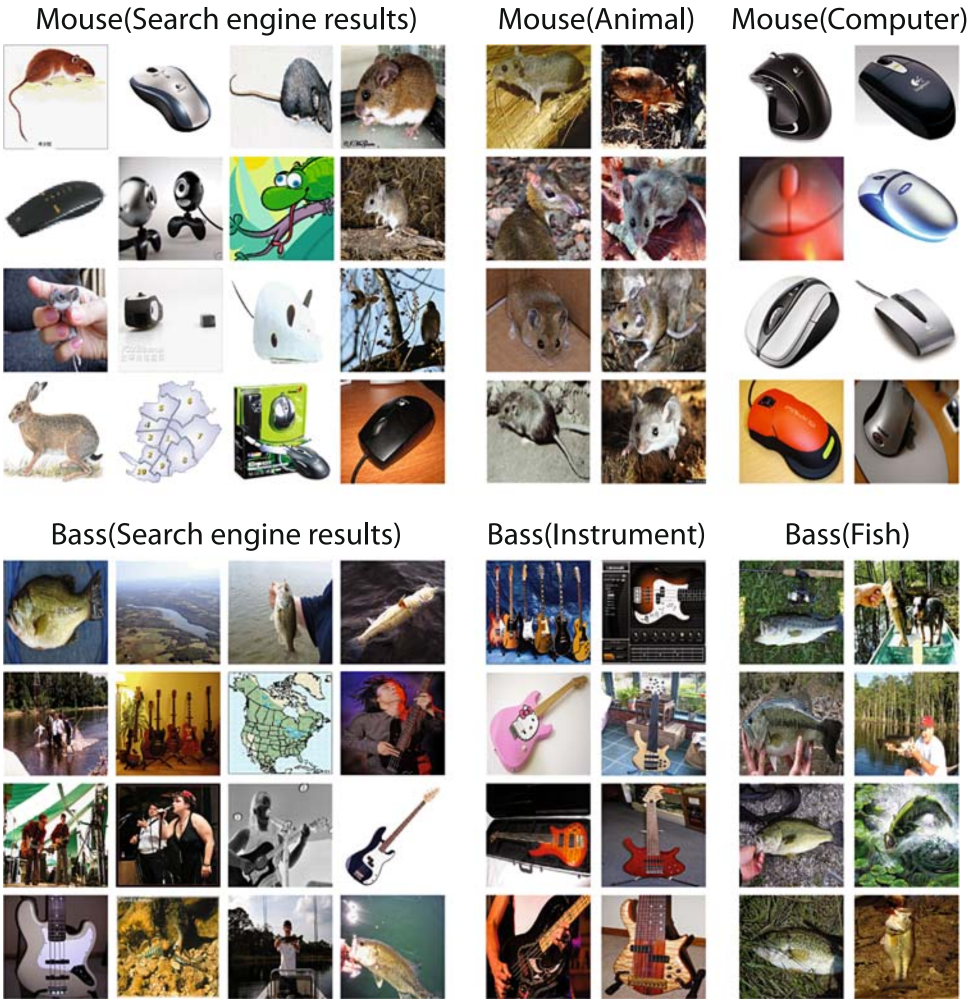

**Fig. 17** Polysemy discovery using OPTIMOL. Two polysemous query words are used as examples: "mouse" and "bass". *Left*: Example images of "mouse" and "bass" from image search engines. Notice that in the "mouse" group, the images of animal mouse and computer mouse are intermixed with each other, as well as with other noisy images. The same is true for the "bass" group. For this experiment, 50 seed images are used for each class. *Right*: For each query word, example images of the two main topic clusters discovered by OPTIMOL are demonstrated. We observe that for the "mouse" query, one cluster mainly contains images of the animal mouse, whereas the other cluster contains images of the computer mouse

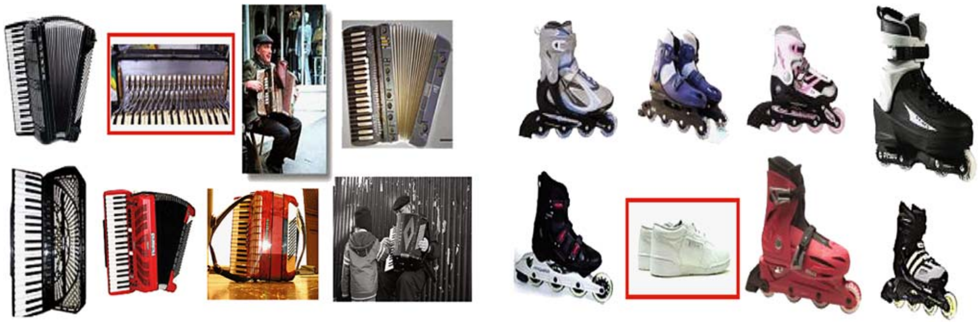

**Fig. 18** (color online) *Left*: Randomly selected images from the image collection result for "accordion" category. False positive images are highlighted by using *red boxes*. *Right*: Randomly selected images from the image collection result for "inline-skate" category

| | a | c | f | g | l | m | w |
|---|---|---|---|---|---|---|---|
| **airplane** | 76.0 | 14.0 | 0.3 | 5.3 | 0.3 | 0.3 | 4.8 |
| **car** | 1.0 | 94.5 | 0.3 | 4.5 | 0.3 | 0.3 | 0.3 |
| **face** | 0.5 | 1.4 | 82.9 | 3.7 | 0.5 | 0.5 | 11.5 |
| **guitar** | 2.2 | 4.9 | 5.6 | 60.4 | 13.3 | 0.2 | 13.3 |
| **leopard** | 1.0 | 2.0 | 1.0 | 5.0 | 89.0 | 1.0 | 2.0 |
| **motorbike** | 0.3 | 5.5 | 0.3 | 5.5 | 1.0 | 67.3 | 20.5 |
| **watch** | 1.7 | 5.5 | 17.7 | 11.0 | 5.5 | 5.0 | 53.6 |

**Fig. 19** Confusion table for Experiment 3. We use the same training and testing datasets as in Fergus et al. (2005b). The average performance of OPTIMOL is 74.82%, whereas Fergus et al. (2005b) reports 72.0%

that learning with all the images accepted in classification lead to over-specialized dataset. To avoid this problem, we introduce the "cache set" to incrementally update the model. In this experiment, we compare the appearances of the images incorporated into the permanent dataset, the "cache set" as well as the discarded ones (in "junk set"). We show example images from the "accordion" class in Fig. 14 (left). We observe that those images to be appended to the dataset are very similar to the training images. Images kept in the "cache set" are more diverse ones among the training images whereas images being discarded are unrelated to the image class. In Fig. 14 (right), we show more example images with highest and lowest entropy values from "accordion" and "inline-skate" classes.

*Illustration of Images Collected Using Different Entropy Thresholds* In addition, we show example images from the "accordion" dataset collected by using two different entropy threshold values. Specifically, the two entropy threshold values selected are: top 30% of the related images with high entropy and all related images. Our experiment shows that a low entropy threshold allows a large proportion of the related images to be used to learn the model. Most of them have similar appearance compared to the seed images. Learning from these images makes the model susceptible to over-specialized. In other words, the updated model tends to collect even more similar images in the next iteration. Figure 15 (left) shows that images collected by OPTIMOL with a low threshold are highly similar to each other. On the other hand, a high threshold provides more diverse images for the model learning, which leads to a more robust model capable of collecting more diverse images. We show these diverse images in Fig. 15 (right).

*Detection Performance Comparison* In this experiment, we compare detection performance of OPTIMOL with different numbers of seed images. Using "accordion" dataset as an example, we show in Fig. 16 detection performance as a function of number of seed images. We use the detection rate as the criterion to measure the performance of detection. A higher detection rate indicates better performance. Our experiment shows that when the number of seed images is small, the detection rate increases significantly along with the number of seed images. When adequate initial training images are provided to train a good classifier, OPTIMOL acts robustly in selecting good examples to train itself automatically. From then on, adding seed images makes little difference in the self-training process.

*Polysemous Class Analysis* We have discussed in the introduction that one challenge in collecting images from text queries is the issue of polysemy. A "mouse" could mean a "computer mouse", or an "animal mouse". In this experiment, we demonstrate that OPTIMOL is capable of discovering different clusters of images that reflect the polysemous nature of the query word(s). Figure 17 illustrates the result. As Fig. 17 shows, an image search result of "mouse" gives us images of both the "computer mouse" and the "animal mouse", in addition to other noisy images not necessarily related to either. Using a small number of seed images, OPTIMOL learns a model that captures the polysemous nature of the query. In Fig. 17, we show examples of images belonging to two main topic clusters estimated by OPTIMOL. It is clear that one topic contains images of the animal mouse, whereas the other contains images of the computer mouse. OPTIMOL achieves this discovery of multiple semantic clusters (i.e. polysemy) due to its ability to automatically assign meaningful topics to different images.

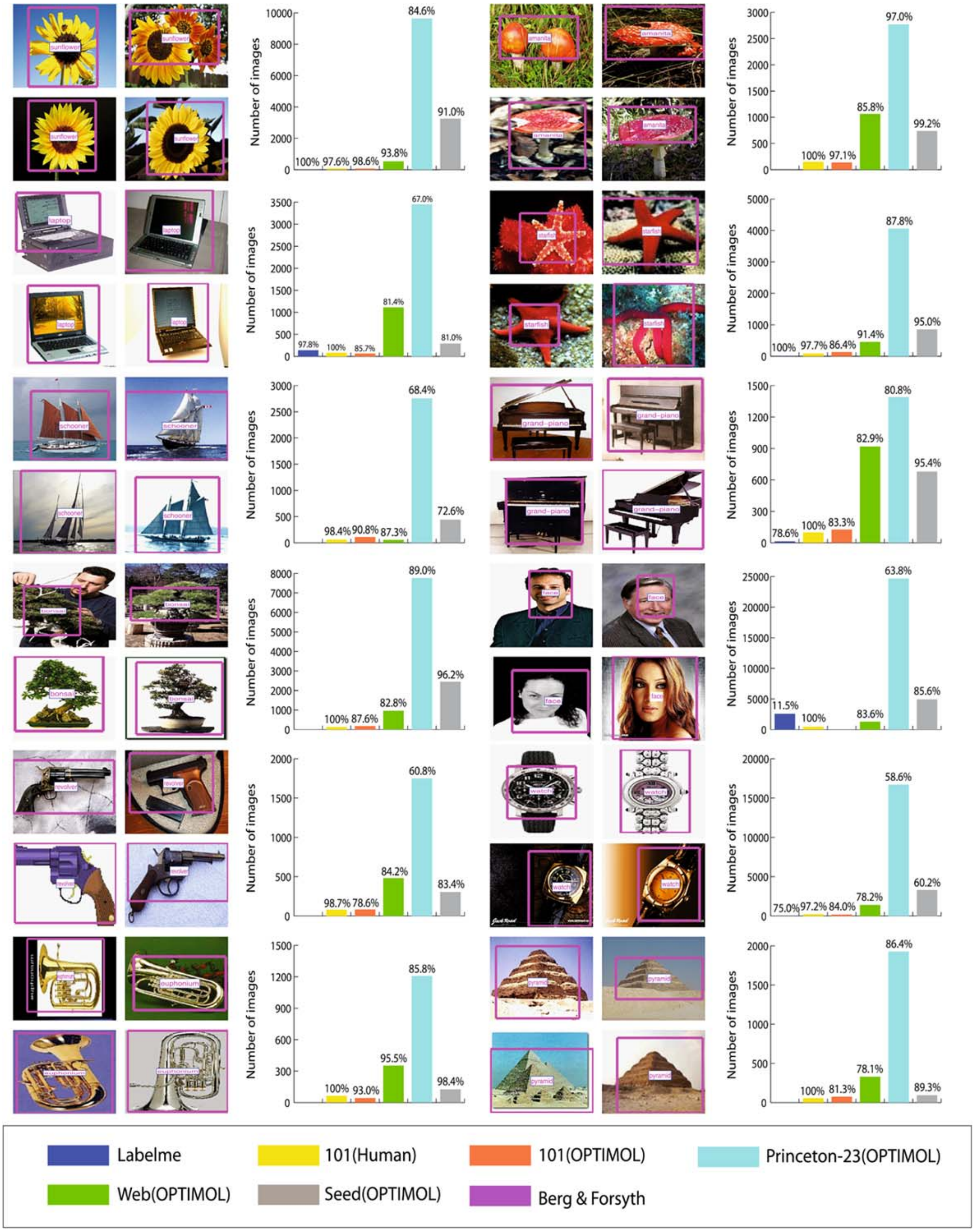

**Fig. 20** (Color online) Image collection and annotation results by OPTIMOL. The notations of the bars are provided at the bottom. Each row in the figure contains two categories, where each category includes 4 sample annotation results and a bar plot. Let us use "Sunflower" as an example. The *left sub-panel* gives 4 sample annotation results (bounding box indicates the estimated locations and sizes of the "Sunflower"). The *right sub-panel* shows the comparison of the number of images in "Sunflower" category given different datasets. The *blue bar* indicates the number of "Sunflower" images in LabelMe dataset, the *yellow bar* the number of images in Caltech 101-Human. The OPTIMOL results are displayed using the *red*, *green*, and *cyan bars*, representing the numbers of images retrieved for the "Sunflower" category in Caltech 101-Web, Web-23 and Princeton-23 dataset respectively. The *gray bar* in each figure represents the number of images retrieved from the Princeton-23 dataset by the base model trained with only seed images. The number on top of each bar represents the detection rate for that dataset. Since the pictures in the "face" category of Caltech 101-Human were taken by camera instead of downloading from the web, the raw Caltech images of the "face" category are not available. Hence, there is no result for "face" category by 101 (OPTIMOL). All of our results have been put online at http://vision.stanford.edu/projects/OPTIMOL.htm

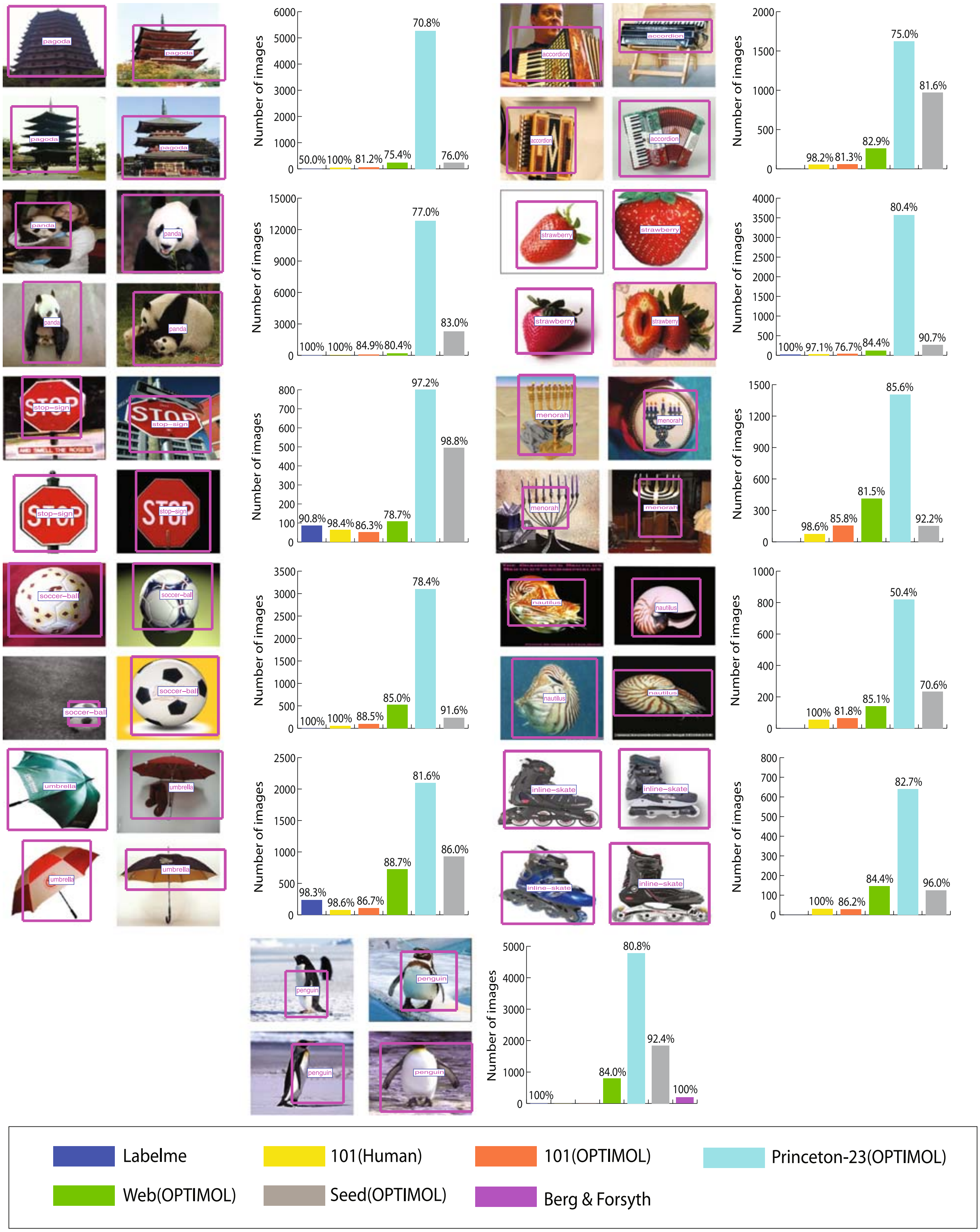

**Fig. 21** Image collection and annotation results by OPTIMOL. Notation is the same as Fig. 20

### 5.3 Experiment 2: Image Collection

21 object categories are selected randomly from Caltech 101-Web for this experiment. The experiment is split into three parts: 1. Retrieval from Caltech 101-Web. The number of collected images in each category is compared with the manually collected images in Caltech 101-Human. 2. Retrieval from Web-23 using the same 21 categories as in part 1. 3. Retrieval from Princeton-23 using the same 21 categories as in part 1. Results of these three experiments are displayed in Figs. 20 and 21. We first observe that OPTIMOL is capable of automatically collecting very similar number of images from Caltech 101-Web as the humans have done by hand in Caltech 101-Human. Furthermore, by using images from Web-23, OPTIMOL collects on average 6 times as many images as Caltech 101-Human (some even 10 times higher). Princeton-23 provides a further jump on the number of collected images to approximately 20 times as that of Caltech 101-Human. In Fig. 20, we also compare our results with LabelMe (Russell et al. 2005) for each of the 22 categories. A "penguin" category is also included so that we can compare our results with the state-of-art dataset collecting approach (Berg and Forsyth 2006). In all cases, OPTIMOL collected more related images than the Caltech 101-Human, the LabelMe dataset and the approach in Berg and Forsyth (2006). In addition, we conduct an additional experiment to demonstrate that OPTIMOL performs better than the base model by comparing their performance of dataset collection. The result is shown in Fig. 20, where the number of images collected by the base model is represented by the gray bar. The likelihood ratio threshold is set at the same value for the full OPTIMOL model and the base model. The comparison indicates that the full OPTIMOL model collects significantly more images than the base model. This is attributed to the effectiveness of the iterative classification and model learning in OPTIMOL. Note that all of these results are achieved without any human intervention[5], thus suggesting the viability of OPTIMOL as an alternative to costly human dataset collection. In Fig. 18, we demonstrate sample images from the OPTIMOL-collected datasets of "accordion" and "inline-skate" categories in the Princeton-23 data. We highlight the false positives among the images. These mistakes are most likely due to the similar appearance of the false positive images to those of the foreground images.

### 5.4 Experiment 3: Classification

To demonstrate that OPTIMOL not only collects large datasets of images, but also learns good models for object classification, we conduct experiment on Fergus ICCV'05 dataset. In this experiment, we use the same experiment settings as in Fergus et al. (2005b) to test the multi-class classification ability of OPTIMOL. 7 object category models are learnt from the same training sets used by Fergus et al. (2005b). We use the same validation set in Fergus et al. (2005b) to train a 7-way SVM classifier to perform object classification. The input of the SVM classifier is a vector of 7 entries, each denoting the image likelihood given each of the 7 class models. The results are shown in Fig. 19, where we achieve an average performance of 74.8%. This result is comparable to the 72.0% achieved by Fergus et al. (2005b). Our results show that OPTIMOL is capable of learning reliable object models.

## 6 Conclusion and Future Work

We have proposed a new approach (OPTIMOL) for image dataset collection and model learning. The self training framework makes our model more robust and generalized whereas the incremental learning algorithm boosts the speed. Our experiments show that as a fully automated system, OPTIMOL achieves accurate diverse dataset collection result nearly as good as those of humans. In addition, it provides a useful annotation of the objects in the images. Further experiments show that the models learnt by OPTIMOL are competitive with the current state-of-the-art model learned from Internet images for object classification. Human labor is one of the most costly and valuable resources in research. We provide OPTIMOL as a promising alternative to collect larger diverse image datasets with high accuracy. For future studies, we will further improve the performance of OPTIMOL by refining the model learning step and introducing more descriptive object models.

## References

Agarwal, S., Awan, A., Roth, D. (2004). Learning to detect objects in images via a sparse, part-based representation. *PAMI*, *26*(11), 1475–1490.

Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D. M., & Jordan, M. I. (2003). Matching words and pictures. *The Journal of Machine Learning Research*, *3*.

Barnard, K., & Forsyth, D. (2001). Learning the semantics of words and pictures. In *Eighth IEEE international conference on computer vision* (pp. 408–415).

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 711–720.

[5] We use 15 images from Caltech 101-Human as seed for the image collection experiment of Caltech 101-Raw since we do not have the order of the downloaded images for Caltech 101-Raw. The detection rates of Caltech 101-Raw and Web-23 in Fig. 20 are comparable indicating the equivalent effects of automatic and manual selection of seed set on image dataset collecting task.

Berg, T. L., & Forsyth, D. A. (2006). Animals on the web. In *Proc. computer vision and pattern recognition*.

Besemer, J., Lomsadze, A., & Borodovsky, M. (2001). GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions. *Nucleic Acids Research*, *29*(12), 2607.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer, New York.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Bosch, A., Zisserman, A., & Munoz, X. (2006). Scene classification via pLSA. *Proc. ECCV*, *4*, 517–530.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on computational learning theory* (pp. 144–152).

Carson, C., Thomas, M., Belongie, S., Hellerstein, J.M., & Malik, J. (1999). Blobworld: A system for region-based image indexing and retrieval. In *Third international conference on visual information systems* (pp. 509–516).

Chen, Y., Wang, J. Z., & Krovetz, R. (2003). Content-based image retrieval by clustering. In *Proceedings of the 5th ACM SIGMM international workshop on multimedia information retrieval* (pp. 193–200).

Collins, B., Deng, J., Li, K., & Fei-Fei, L. (2008). Toward scalable dataset construction: An active learning approach. In *Proc. ECCV.*

Csurka, G., Bray, C., Dance, C., & Fan, L. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV* (pp. 1–22).

Deng, Y. Manjunath, B. S., Kenney, C., Moore, M.S., & Shin, H. (2001). An efficient color representation for image retrieval. *IEEE Transactions on Image Processing*, *10*(1), 140–147.

Fei-Fei, L., & Perona, P. (2005). A Bayesian hierarchy model for learning natural scene categories. *Computer Vision and Pattern Recognition*.

Fei-Fei, L., Fergus, R., & Perona, P. (2003). A Bayesian approach to unsupervised one-Shot learning of object categories. In *Proceedings of the 9th international conference on computer vision* (pp. 1134–1141). Nice, France.

Fei-Fei, L., Fergus, R., & Perona, P. (2004). Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *Workshop on generative-model based vision*.

Fei-Fei, L., Fergus, R., & Perona, P. (2006). One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Fei-Fei, L., Fergus, R., & Torralba, A. (2007). Recognizing and learning object categories. Short course CVPR. http://people.csail.mit.edu/torralba/shortCourseRLOC/index.html.

Felzenszwalb, P., & Huttenlocher, D. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, *1*, 55–79.

Feng, H. M., & Chua, T. S. (2003). A bootstrapping approach to annotating large image collection. In *Proceedings of the 5th ACM SIGMM international workshop on multimedia information retrieval* (pp. 55–62).

Fergus, R., Perona, P., & Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In *Proc. computer vision and pattern recognition* (pp. 264–271).

Fergus, R., Perona, P., & Zisserman, A. (2004). A visual category filter for Google images. In *Proc. 8th European conf. on computer vision*.

Fergus, R., Perona, P., & Zisserman, A. (2005a). A sparse object category model for efficient learning and exhaustive recognition. In *Proc. computer vision and pattern recognition*.

Fergus, R., Fei-Fei, L., Perona, P., & Zisserman, A. (2005b). Learning object categories from Google image search. In *Computer vision, 2005. ICCV 2005. Tenth IEEE international conference*.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, *1*(2), 209–230.

Fink, M., & Ullman, S. (2007). From Aardvark to Zorro: A benchmark of mammal images.

Freund, Y., & Schapire, R. E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory: Second European conference, EuroCOLT'95, proceedings* (p. 23). Barcelona, Spain, 13–15 March 1995. Berlin: Springer.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Machine learning—international workshop then conference* (pp. 148–156). San Mateo: Morgan Kaufmann.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 721–741.

Griffin, G., Holub, A., Perona, P. (2007). Caltech-256 object category dataset.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval* (pp. 50–57).

Jain, A. K., & Vailaya, A. (1996). Image retrieval using color and shape. *Pattern Recognition*, *29*(8), 1233–1244.

Jeon, J., Lavrenko, V., & Manmatha, R. (2003). Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 119–126). New York: ACM.

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

Krempp, S., Geman, D., & Amit, Y. (2002). *Sequential learning with reusable parts for object detection* (Technical report). Johns Hopkins University.

LeCun, Y., Huang, F., & Bottou, L. (2004). Learning methods for generic object recognition with invariance to pose and lighting. In *Proc. CVPR*.

Leibe, B., & Schiele, B. (2004). Combined object categorization and segmentation with an implicit shape model. In *Proc. workshop on statistical learning in computer vision*, Prague, Czech Republic.

Li, J., Wang, J. Z., & Wiederhold, G. (2000). IRM: integrated region matching for image retrieval. In *Proceedings of the eighth ACM international conference on multimedia* (pp. 147–156).

Lowe, D. (1999). Object recognition from local scale-invariant features. In *Proc. international conference on computer vision*.

McCallum, A., Freitag, D., & Pereira, F. (2000). Maximum entropy Markov models for information extraction and segmentation. In *Proc. 17th international conf. on machine learning* (pp. 591–598).

McClosky, D., Charniak, E., & Johnson, M. (2006). Effective self-training for parsing. In *Proceedings of the main conference on human language technology conference of the North American Chapter of the Association of Computational Linguistics* (pp. 152–159). Morristown: Association for Computational Linguistics.

Miller, G. A. (1995). WordNet: A lexical database for English. *Communications of the ACM*, *38*(11), 39.

Neal, R., & Hinton, G. (1998). A view of the EM algorithm that justifies incremental, sparse and other variants. In M.I. Jordan (Ed.), *Learning in graphical models* (pp. 355–368). Norwell: Kluwer Academic.

PASCAL (2006). The PASCAL object recognition database collection. http://www.pascal-network.org/challenges/VOC/databases.html.

Pawan Kumar, M., Torr, P.H.S., & Zisserman, A. (2005). Obj cut. In *Proceedings of the 2005 IEEE computer society conference on computer vision and pattern recognition* (Vol. 1, pp. 18–25). Washington, DC, USA, 2005. Los Alamitos: IEEE Computer Society.

Rosenberg, C., Hebert, M., & Schneiderman, H. (2005). Semi-supervised selftraining of object detection models. In *Seventh IEEE workshop on applications of computer vision*.

Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2005) Labelme: a database and web-based tool for image annotation.

Schroff, F., Criminisi, A., & Zisserman, A. (2007). Harvesting image databases from the web. In *Computer vision, 2007. ICCV 2007. IEEE 11th international conference*.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, *4*(2), 639–650.

Sivic, J., Russell, B. C., Efros, A., Zisserman, A., & Freeman, W. T. (2005). Discovering object categories in image collections. In *Proc. international conference on computer vision*.

Sivic, J., & Zisserman, A. (2003). Video Google: A text retrieval approach to object matching in videos. In *Ninth IEEE international conference on computer vision, 2003. Proceedings* (pp. 1470–1477).

Sudderth, E., Torralba, A., Freeman, W., Willsky, A. (2005a). Describing visual scenes using transformed Dirichlet processes. *Advances in Neural Information Processing Systems*, *18*, 1297–1304.

Sudderth, E., Torralba, A., Freeman, W. T., & Willsky, A. (2005b). Learning hierarchical models of scenes, objects, and parts. In *Proc. international conference on computer vision*.

Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2006). Hierarchical Dirichlet processes. *Journal of the American Statistical Association*.

Wang, G., Zhang, Y., & Fei-Fei, L. (2006). Using dependent regions for object categorization in a generative framework. *Computer Vision and Pattern Recognition*.

Weber, M., Welling, M., & Perona, P. (2000). Unsupervised learning of models for recognition. In *Proc. European conference on computer vision* (Vol. 2, pp. 101–108).

Yanai, K., & Barnard, K. (2005). Probabilistic web image gathering. In *ACM SIGMM workshop on multimedia information retrieval* (pp. 57–64).

Yao, Z.-Y., Yang, X., & Zhu, S.C. (2007). Introduction to a large scale general purpose groundtruth dataset: methodology, annotation tool, and benchmarks. In *6th int. conf. on EMMCVPR*.

Zhou, X. S. & Huang, T. S. (2002). Unifying keywords and visual contents in image retrieval. *Multimedia, IEEE*, *9*(2), 23–33.

Zhu, X. (2006). *Semi-supervised learning literature survey*. Computer Science, University of Wisconsin—Madison.