Towards Recognizing Any Scene

Anonymous CVPR submission

Paper ID ****

Abstract

...

1. Introduction

Scene perception is the visual perception of an environment as viewed by an observer at any given time. It includes not only the perception of individual objects, but also such things as their relative locations, and expectations about what other kinds of objects might be encountered [CITE Biederman]. Understanding how humans perceive scenes has been one of the most important topics for human vision and computer vision study. Much progress has been made in this field during the past decades [CITE SOME PROMINANT COMPUTER VISION & HUMAN VISION WORK]. In particular, with the popularity of powerful classification tools, most researchers choose to model the scene recognition problem as a classification problem where each image is classfied to belong to a scene category. As a result, much efforts are spent in minimizing the classification error.

Though this paradigm seems to be natural and has become a standard for many years, there are fundamental issues to be addressed. To show the disadvantage of modeling scene recognition problem as a classification problem, let's start from a simple example. Imagine a small room that has a bed and couch on one end, and cupboard, stove and other cooking related facilities on the other end (Fig. 1). Looking at only one end of the room, it is appropriate to predict that the room is likely to be a bedroom or a kitchen. However, looking at the whole, it is inappropriate to classify this room either to be a bedroom or a kitchen. From this example, we see that the scene perception of an environment can be different along with the change of the viewpoint. Essentially, the bed and couch are coupled together to form a unit for rest and the cupboard, stove and other facilities are coupled together to form a unit for cooking. A smart scene recognition algorithm should be able to identify those coupled objects as a coherent unit.

Figure 1. composite scene example

In this work, we review the concept of "scene" and explore to discover scenes in a data driven approach. Viewing from linguistics point of view, we hypothesize that the concept of scene is introduced so that people can use it to compactly communicate the gist of what they see, in particular, the environment (such as outdoor and indoor) and the functionality of such environment (such as kitchen and bathroom). Inspired by the recent success of using objects for scene recognition [CITE OBJECT BANK], in this work we use groups of objects to characterize the environment and its functionality. Therefore, "scene" is formulated as a set of "object groups" and each "object group" is a group of objects that typically co-occur and interact as a whole.

Formally, we propose to learn object groups using unsupervised feature learning framework subject to a list of regularizations. Therefore, we can build an image representation in the "object groups" space. To verify that our learned "object groups" aggregate objects that are semantically coherent and faithfully reflect the distance in scene space, we did a clustering experiment in the "object group" space. Results show that simple clustering algorithms such as k-means would effectively learn image clusters that are highly consistent with human defined scene categories, significantly outperforming image clusters learned from the original object space. To interprete the phenomenon, we hypothesize that the proposed "object group" is constituted by

a group of semantically (functionally or environmentally) coherent objects, therefore, distance in this space better reflects the scene-level similarity of images. This hypothesis is consolidated by human study. Every subject is asked to scan the objects in a group and check if they are semantically coherent (together perform some function or consistent in an environment).

Intuitively, given the learned object groups, we can cluster annotated images and discover underlying scenes without human intervention in a data-driven manner. Here, we further extend the scene relationships of images beyond simple L2 distance. Because scenes can share object groups (e.g., studio and kitchen share the same object group of "cooking" functionality, therefore, we define relations of scenes according to the object groups they contain so that we can visulaize collections of images in a more interesting way.

To annotate scenes in unseen images, we train object groups detectors to localize "object groups" using low-level and high-level features. Given the detector responses, we can interconnect unseen images to our collected image database according to the scene relationship. On the other hand, because the appearance of objects and their spatial relationship in a particular "object group" are more constrained than in the general case, object group detectors give better detection performance than individual object detectors.

To summarize, the main contribution of the paper is: ...

2. Related Work

This section needs to be carefully written. Discuss related work in object & scene recogntion, in particular, make sure to include the following lines of work:

1. traditional scene classification work
2. latest scene work by Oliva & Torralba (900 Scenes)
3. total scene understanding work
4. multi-label classification work [Jiebo Luo]
5. the latest "Visual Phrase" work by Sadeghi & Farhadi
6. contextual work

In Sadeghi & Farhadi's paper, there is a paragraph talking about machine translation, which is interesting. They also talked about object interaction works, which is interesting, too. We might need to talk about it, too.

Difference from Sadeghi & Farhadi's work: First, in [**?**], visual phrases are manually defined, focusing on activities of a subject (e.g. a person or a dog running) in practice. Differently, our "object group" is automatically learned from data. Second, in each of their visual phrases, the number of objects is fixed (one or two), so they can use Felzonszwalb's deformable model with fixed number of parts for detection. Differently, our "object group" is learned from the MDL principle and each "object group" has flexible number of objects. That makes our model more powerful for scene understanding. For example, in a classroom, there should be rows of desks and chairs. Such difference drives us looking for different detection model.

3. Object group learning

As introduced before, the idea behind the concept of "object group" is a set of objects that tend to coupled as a unit in scene perception. The visual cue of these objects in an object group is that they are likely to co-occur with a specific relative locations. Formally, in this paper we model each object group as a template and we learn these template as linear filters in an unsupervised feature learning framework. Given the learned object group template, we can obtain the responses of these templates on each image. We then do max-pooling on the responses so that we get an object group representation for each image. We will give details in the rest of this section.

3.1. Notations in Image and object group representation

Let $\mathcal{I} = \{I_1, I_2, \ldots, I_p\}$ denote the set of images, where $I_i \in \mathcal{I}$ is a single specific image. Let $\mathcal{G} = \{G_1, G_2, \ldots, G_g\}$ denote the set of g object groups. Then our goal is then to learn $\mathcal{G}$ from $\mathcal{I}$.

Let $\mathcal{O} = \{O_1, O_2, \ldots, O_h\}$ denote a set of predefined words, which can be object names or visual words obtained from clustering image patch descriptors.

Suppose $I_i \in \mathcal{I}$ is an image sampled from from $\mathcal{I}$. We show how we build the representation for $I_i \in \mathcal{R}^{M \times N \times h}$. We first equally divide I_i both horizontally and vertically into $M \times N$ grids. For each grid, we compute a histogram of words from $\mathcal{O}$ (See Fig 2). When $\mathcal{O}$ is the universe of object names, we require ground truth object annotations so that every pixel is labeled to belong to one of the objects in $\mathcal{O}$ and each bin z of the histogram is the percentage of pixels labeled O_z in that grid. For visual words, the representation

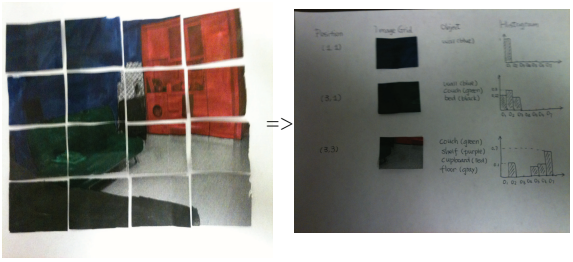

Figure 2. Represent an image as grids of histograms

can be built similarly except that we build histogram by the percentage of a certain visual words in a grid. For statement clarity, we will assume $\mathcal{O}$ to be the universe of object names in the rest of this paper if there is no further notification.

Suppose $G_j \in \mathcal{G}$ is an object group template, then we represent $G_j \in \mathcal{R}^{m\times n\times h}$, where $m \leq M$ and $n \leq N$. It is divided both horizontally and vertically into $m \times n$ grids, and in each grid there exists a histogram of words from $\mathcal{O}$. $G_j(x, y, :)$ is a vector with $h = |\mathcal{O}|$ elements, each of which describes the portion of the corresponding word in $\mathcal{O}$. Therefore, an object group is a template that corresponds to the grids of histograms representation of an image.

Here is a simple example to illustrate such idea .Suppose we have $h = 5$ predefined individual objects listed as follows, $\mathcal{O} = \{O_1, O_2, O_3, O_4, O_5\} = \{$"tree", "desk", "flower", "house", "bed"$\}$. Then, $G_j(x, y, :) = [0.4, 0.3, 0.2, 0.1, 0]^T$ means for an image grid cell with position (x, y) in G_j, we expect 40% of the area depicts trees, 30% of the area depicts desks, 20% of the area depicts flowers, 10% of the area depicts houses and no bed at all in this image grid cell. Now it's clear for us to see that G_j integrates both the position information and area information.

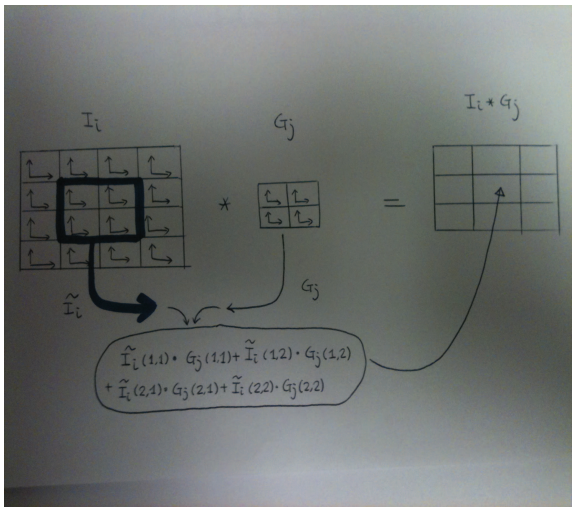

Figure 3. Generate responses of object groups

Given an object group template G_j, we can slide it upon the image grids and compute a response at each location. For computational efficiency, we compute the responses by correlating the object group template with the image grid (See Fig 3)

3.2. Problem formulation

In our assumption, objects from the same object group tend to co-occur with a fixed relative location. Therefore, our object group based representation needs to be invariant if the whole object group translate together.

Here, we achieve the translation invariance by taking the max pooling operation. More specifically, let $s_i \in \mathcal{R}^g$ denote the object group based representation we want to compute. Then the j-th element corresponding to the j-th object group G_j is calculated as follows:

$$s_i(j) = \max\{I_i * G_j\}, \quad j = 1, \ldots, g. \tag{1}$$

3.2.1 Sparse Filtering Model

In this section, we first give our overall unsupervised object group learning model and then give detailed explanation of the design of our model. The overall model can be represented as the following optimization problem over object group template:

$$\text{minimize}_G \quad \sum_{i=1}^{p} \frac{\|s_i\|_1}{\|s_i\|_2} - \frac{\|\bar{s}\|_1}{\|\bar{s}\|_2} \tag{2}$$

$$\text{subject to} \quad s_i(j) = \max\{I_i * G_j\},\ j = 1, \ldots, g; \tag{3}$$

$$\bar{s} = \frac{1}{p}\sum_{i=1}^{p} s_i; \tag{4}$$

$$\sum_{k=1}^{h} G_j(:, :, k) = 1,\ j = 1, \ldots, g; \tag{5}$$

$$G_j(:, :, k) \geq 0,\ j = 1, \ldots, g,\ k = 1, \ldots, h. \tag{6}$$

Next, we will explain the design of each constraint and the objective.

3.2.2 G_j: Object Ingredient Analysis

Since each element in G_j represents the proportion of a word appeared in an image area, it's then natural to have constraints written in (5) and (6) for each G_j. Namely, we require the bin values in each grid of the object group template to be non-negative and normalized to be a probability.

3.2.3 $s_i(j)$: Object Group Response

As described at the beginning of this section, Constraint (3) gives a definition of $s_i(j)$ so that the object group based representation of an image is invariant with respect to the translation of the whole object group. Therefore, when literally reads from Constraint (3), $s_i(j)$ is the maximum of the correlation of image I_i and object group G_j. If $s_i(j)$ is large, it means that image I_i contains object group G_j with a high probability; conversely, if $s_i(j)$ is small, it is very unlikely that G_j is contained in I_i.

One more piece of information can be extracted from the "max" function, i.e., the most likely position for an object group G_j to appear in image I_i, assuming G_j is contained in I_i. To find such position, we simply extract the index associated with the maximum from the "max" function. More accurately, we can take argmax$\{I_i * G_j\}$ to find where the object group is.

3.2.4 Bias the Responses with Sparsity Pattern

$s_i(j)$ as described above, richly encode the information required to represent image I_i in the language of $\mathcal{G}$. It plays a key role in stating the relation between images and object groups. To make the discussion in this part convenient, we are going to realign all the $s_i(j)'s$ into a big matrix, called S:

$$S \in \mathcal{R}^{p\times g}, \text{ where } [S]_{ij} = s_i(j), 1 \le i \le p, 1 \le j \le g. \tag{7}$$

So each row of S corresponds to an image, we may call it an object group based representation for the image. This matrix is essentially a feature matrix and our objective encourages S to bias towards our desired strucutre.

There have been a lot of researches in the field of regularizing feature distribution patterns. Properties like **Population Sparsity**, **Lifetime Sparsity** and **High Dispersal** have been explored in the neuroscience literature. New methods designed based on these properties have emerged and been verified in machine learning [CITE papers on Sparse Filtering].

Our model heavily inherits the properties mentioned above to achieve good performance in our experiments. Put it in our language using the S matrix. We desire it to have the following properties.

- **Population Sparsity** Each image (each row of S) should be represented by only a few active (non-zeros) object groups. This is simply inferred from the fact that each image can only have a limited number of objects, hence even more limited number of object groups in it.
- **Lifetime Sparsity.** Each column of S , i.e., each object group should only be active in a few images. This requirement clearly asks for the discrimination ability of $\mathcal{G}$ so that images can be separated in the object group space.
- **High Dispersal.** We demand that no one column of S should have significantly more activity than the other columns. Concretely speaking, we consider the mean activations of each object group (feature) obtained by averaging the values in the S matrix across the rows (images), which is described by Constraint (4). The components of $\bar{s}$ should be roughly the same, implying that all object groups have similar contributions.

Following the design of [CITE Sparse Filtering], we did not explicitly enforce the Lifetime Sparsity, but it is shown that the Lifetime Sparsity can be implied once Population Sparsity and High Dispersal are pursued.

Then it is straight forward to encode Population Sparsity and High Dispersal in the objective (2), wherein the first term of the objective describes Population Sparsity and the second term describes High Dispersal.

3.3. Model Pretraining using pLSA

As the objective of this model is non-convex, we can only search local optima. In practice, a good initialization point turns to be important for learning a good model that both gives good performance and aligns well with human intuition. We use a pLSA model to learn the initial feasible solution. Concretely, we ...

4. Recognition

In this part, we will show how to localize object groups in images and use the detected object groups for "any scene" recognition.

To evaluate our algortihm, we collect a data set of 2475 images of 17 scene classes with full object annotation. The images are picked from LabelMe[CITE Torralba]. Additionally, we create a very challenging dataset consisting of 300 images of "composite scene", namely, each contains more than one functional object groups.

4.1. Automatic Scene Grouping

In this section, we show how we automatically group images into scene clusters using our concept of object groups.

Method	Average Precision
Object Histogram	54%
Object group + ℓ_2, kmeans pretrain	67%
Object group + ℓ_1, kmeans pretrain	69%
Object group + ℓ_2, pLSA pretrain	73%
Object group + ℓ_1, pLSA pretrain	75%

Table 1. Scene Grouping by Ground Truth Object Annotation

Method	Average Precision
Object Histogram	
Object group + ℓ_1	
Object group + ℓ_2	

Table 2. Scene Grouping by Visual Words

(1,1): table:0.874, vase:0.023, bowl:0.021, floor:0.017, plate:0.012, flowers:0.011, wall:0.008, sink:0.006,
(1,2): table:0.888, bowl:0.016, floor:0.015, plate:0.012, vase:0.012, wall:0.010, flowers:0.008, chair:0.008,
(2,1): table:0.994, floor:0.006,
(2,2): table:0.999, floor:0.001,

Figure 4. Examples for object groups. Each column is an object group. The first row is the histogram of objects, the latter rows are example pictures of the object group and the last row is the mean image of the object group

We will first show an ideal case where all the object annotations are provided. This assumption corresponds to the case when $\mathcal{O}$ is the universe of object names. We will then show our experiment result of learning object group using visual words.

To evaluate the automatic scene grouping performance, for each cluster, we will provide it a scene label base upon the image closest to the cluster center. This method can be viewed as an "active" classification method where only a very limited number of images needs to be labeled. Tabel 4.1 shows the performance.

Next, we show how we use visual words for the clustering. [Haizi's paragraph]

4.2. Object Group Localization

To evaluate the performance of the object group localization, we randomly pick 70% images from the dataset for training and use the rest for test. The localization performance is evaluated by PR curve, see Figure **??**. We compare the detection performance with (1) simply running individual object detectors using Felzonszwalb's object detectors and then adding a surrounding bounding box, (2) Visual Phrase detector. Some detection examples are shown in Figure 4.

Figure 5. Detection examples for object groups

4.3. Scene retrieval

As our experiments in Sec 4 shows, the distance in the object groups space better reflects the semantic similarity of images. In this experiment, we use our learned object group detectors to localize the object groups so that we build a feature vector for an unseen image in the object group space. We then do image retrieval in the space.

Our experiments uses the output of Sec **??** to build the feature vectors and retrieve the test images from training images using ℓ_1 distance and ℓ_2 distance. For each query image, we take the top 10 examples and calculate the Average Precision as a measurement. We show the results and some examples in Figure **??**

1. **Dataset** Undetermined. Have not found Lazebnik's dataset. We can use LabelMe images (1200 images of 10 classes in the previous section) as query, all SUN dataset and MIT Indoor images as test.

2. **Evaluation metric** Average Precision, scene label as the ground truth

3. **Control Experiments**

   (a) Object group + ℓ_1

   (b) Object group + ℓ_2

   (c) Object group + good distance

   (d) SPM+histogram intersection

   (e) GIST+L2

Method	Average Precision
Object group + ℓ_1	
Object group + ℓ_2	
Object group + good distance	
SPM + histogram intersection	
GIST+L2	

Figure 6. Retrieval figure

4.4. Any scene decomposition

4.4.1 Scene labeling for composite scene images

In this task, we predict scene labels for composite scene images and localize where the scene part is. For example, as in Figure 1, it is expected that the algorithm will predict bedroom, kitchen and laundry as labels and localize where the scene part is.

1. **Dataset** Composite scene dataset and the 17 scene dataset
2. **Evaluation** We evaluate the scene label prediction and localization performance respectively, in terms of accuracy.
3. **Control Experiments** For control experiments, we compare with ordinary distance measure such as ℓ_1 and ℓ_2 distance and predict multiple labels from k-nearest neighbor[1], measured by accuracy. We also compare with different image representations such as object bank, SPM and GIST.
   (a) Object group + ℓ_1
   (b) Object group + ℓ_2
   (c) Object group + good distance
   (d) SPM+histogram intersection
   (e) GIST+L2

We show results and figures in Figure 4.4.1.

Figure 7. Any scene decomposition

5. Conclusion

The concept of "object group" is important!

[1] Usually, k-nearest neighbor classifier predict a single label by do max-voting. Here, we predict the labels for an image as the union of all labels in the k-nearest neighbors.