Semantic Scene Concept Learning by an Autonomous Agent

Weiyu Zhu

Illinois Wesleyan University
PO Box 2900, Bloomington, IL 61702
wzhu@iwu.edu

Abstract

Scene understanding addresses the issue of "what a scene contains". Existing research on scene understanding is typically focused on classifying a scene into classes that are of the same category type. These approaches, although they solve some scene-understanding tasks successfully, in general fail to address the semantics in scene understanding. For example, how does an agent learn the concept label "red" and "ball" without being told that it is a color or a shape label in advance? To cope with this problem, we have proposed a novel research called semantic scene concept learning. Our proposed approach models the task of scene understanding as a "multi-labeling" classification problem. Each scene instance perceived by the agent may receive multiple labels coming from different concept categories, where the goal of learning is to let the agent discover the semantic meanings, i.e., the set of relevant visual features, of the scene labels received. Our preliminary experiments have shown the effectiveness of our proposed approach in solving this special intra- and inter-category mixing learning task.

1. Introduction

Scene understanding addresses the issue of "what a scene contains". Existing research on scene understanding is based typically on either scene modeling (Belongie, Malik and Puzicha 2002; Selinger and Nelson 1999) or supervised learning (Murase and Nayar 1995; Mel 1997). In both cases, a detector is built to differentiate one scene label from another, where all labels of interest come from the same category type. Although the existing approaches solve some scene understanding tasks successfully, they in general fail to address another important issue in visual perception: the semantics. For example, how does an agent learn mutually non-exclusive labels such as *red*, *ball*, and *cat* without being told of their category types in advance? The capability of learning the semantics of a label is crucial for intelligent human-computer interaction and robotic natural language acquisition.

To learn semantics of scene labels, supervised learning usually fails the task because one learning system can only classify the labels of one category type. For example, semantically it is valid that a scene receives both labels of *red* and *ball* if it contains a ball object in red color. However, supervised learning fails to address this scenario because the label *red* and *ball* belong to different category types (i.e., the color category and shape category).

To tackle this problem, we have proposed a novel research called "multi-labeling" scene concept learning. The fundamental idea is to learn semantics of scene labels via creating the associations between labels and the relevant visual features contained in images. Scene labels that an agent receives may come from multiple concept categories that are unknown to the agent beforehand. For example, given a scene containing a coke can, the valid labels include *red*, *can*, or *coke*. However, the robot is not told that *red* is a label related to colors while *can* refers to a shape.

Figure 1 illustrates the difference between the task that we are dealing with and that of a supervised learning problem. For both cases, the agent receives labels resided in the leaf nodes (the *l*-nodes). The challenge of the "multi-labeling" learning case lies in the unknown hidden layers, i.e., the category type of the label received (the C-nodes). In addition, the multi-labeling learning has to deal with the scenario that a given scene instance receives multiple labels of different category types instead of only one as in the supervised learning case.

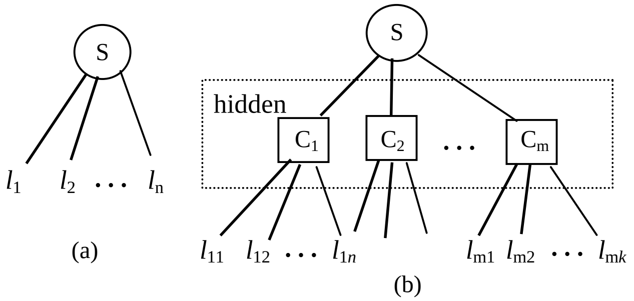

S
l_1 l_2 . . . l_n
(a)
S
hidden
C_1 C_2 . . . C_m
l_{11} l_{12} . . . l_{1n} l_{m1} l_{m2} . . . l_{mk}
(b)

Figure 1. (a) Supervised learning; (b) multi-labeling. S-node: scene instance; C-node: category type; l-node: label

A similar research on scene concept understanding is termed symbol grounding (Duygulu et al. 2002; Mori, Takahashi and Oka 1999; Gorniak and Roy 2004). However, to our best knowledge, none of the existing symbol grounding research addresses the case that a concept label comes from multiple concept categories. For instance, the study in (Duygulu et al. 2002) is to match keywords, which could be more than one, with the relevant components in a picture, where all of the labels are of the same category type, i.e., the category of the objects of interest in a scene.

In this study, we have proposed a generic model, which is based on joint probability density function of visual features, for multi-labeling scene concept learning. As a preliminary step, we have developed a small-sized system, which is based on the proposed learning model, for semantic scene concept learning. The proposed method uses a two-level Bayesian inference network to determine the category type of a scene label. Our preliminary experiments have shown the effectiveness of this method in catching the semantics of labels of unknown category types.

The proposed learning methods are formulated in Section 2. Section 3 and 4 presents the details of our approach, including feature extraction and concept category inferences. Experiments are given in Section 5, followed by the conclusion in Section 6. To avoid confusions, some terminologies used in this paper are summarized below.

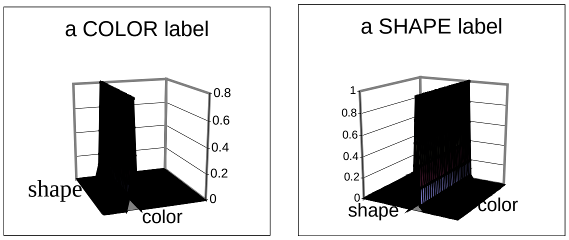
- **Scene labels**: Semantic descriptions of a scene, such as *red*, *square*, and *Pepsi*. All scene label terms are in italics font in this paper.
- **Concept (label) category**: A class of labels characterizing the same type of visual attribute. The categories studied in this paper include color, shape, and the objects of interest. Concept category terms are capitalized in this paper, e.g., color category is denoted as COLOR.
- **Features**: Visual information extracted from an image. A feature is a vector of data. For example, a color feature is a triplet of {hue, saturation, value} and a shape feature could be a vector of seven invariant moments.

2. Task Formulation

2.1 A General Purpose Learning Framework

The goal of semantic scene concept learning is to discover the associations between relevant visual features and scene concept labels that characterize certain semantic meanings of a scene. Given a set of visual features, the task of concept learning can be formulated as discovering the Joint Probability Density Functions (JPDFs) of the visual features of a concept label. For example (Figure 2), given that a scene of interest is characterized with color and shape features, the joint visual feature space is defined as the direct sum of the color and shape feature spaces (for simplicity, the color and shape feature spaces are represented as the two axes in the figure). The JPDF of a typical COLOR label is given in (a), and (b) displays the JPDF of a typical SHAPE label. Similarly, one may obtain the JPDF of an arbitrary object of interest given that the semantics of an OBJECT label can be represented adequately with the combinations of color and shape features only.

Once the associations between labels and feature JPDFs are built, scene labels are retrieved by matching the scene features detected in a picture to the JPDFs of the labels learned. By thresholding the degrees of matches, a set of scene labels are retrieved and used to describe the given scene. For instance, when the robot sees a Pepsi soda can, it will retrieve the related labels *blue*, *can* and *Pepsi*, etc.



(a) (b)

Figure 2. JPDFs of typical COLOR and SHAPE labels

2.2 Proposed Approach

The feature JPDF-based representation scheme may serve as a general model for semantic scene concept learning. However, without incorporating heuristic knowledge, the computation of the JPDF of a scene label has to be based on statistical counting only, which could be time consuming in practice. To cope with this difficulty, our strategy is to utilize the domain knowledge of scene labels to facilitate the learning of feature JPDFs. Specifically, we parameterize, i.e., set the format of, the JPDF of each concept category type according to our knowledge. By fixing the format of feature JPDFs, the learning is formulated as solving two problems: 1) determine whether a scene label belongs to a category by evaluating how well the observed features agree with the format of the JPDF of that category type; 2) compute the parameter set accordingly if the category type of a label can be (or almost be) determined.

At this stage, our study has been focused on learning the labels that can be used to describe a scene containing an object. Preliminarily, we are interested in color and shape features of an object scene. Scene labels come from three categories: COLOR, SHAPE, and the group of the objects of interest (denoted as OBJECT). The formats of feature JPDFs of these category types are defined heuristically as:

$$\begin{aligned}
&\mathrm{JPDF}_{\mathrm{COLOR}} = \Phi(\mu_c, \Psi_c) \\
&\mathrm{JPDF}_{\mathrm{SHAPE}} = \Phi(\mu_s, \Psi_s) \\
&\mathrm{JPDF}_{\mathrm{OBJECT}} = \sum_k \left(\Phi(\mu_s^k, \Psi_s^k) \prod_l \Phi(\mu_c^l, \Psi_c^l) \right)
\end{aligned} \tag{1}$$

where $\Phi(\mu, \Psi)$ is a normal distribution centered at the vector μ with Ψ being the covariance matrix. The subscripts c and s stand for a color or shape feature vector, respectively. The index k indicates the possible views of an object and l indexes the colors that are associated with a certain view of the object. The definition in (1) assumes that the JPDF of a COLOR label must follow a certain normal distribution centered at a color feature vector and be independent to the shape features. Similarly, the JPDF of a SHAPE label is characterized with a shape feature vector (plus the covariance matrix) and is independent to the color features. The JPDF of an OBJECT label is a Gaussian mixture of all possible views of that object, each of which consists of a shape feature and a combination of several color features since the object may contain multiple colors.

According to the above definition, the nature of scene concept learning is to compute the likelihood of a label belonging to a certain category type and meanwhile determine the parameter set accordingly. The first problem is solved using a two-level Bayesian inference network that will be described in the following sections. The second one is solved using the Maximum Likelihood Estimation (MLE) algorithm (Duda, Hart, and Stork 2001) according to the observed scene examples of the concept label.

3. Feature Extraction and Representation

3.1 Preprocessing

Since our focus is semantic scene concept learning based on extracted visual features, the processing of feature extraction was simplified by setting the background of a scene uniform and simple. For each scene instance, an intensity based image segmentation algorithm was used to extract the region of interest from the background (Faugerous 1983).

3.2 Color Feature Extraction

Each extracted region was decomposed into several "significant" color components represented in HSV {hue, saturation, value} standard. A color component is said significant if the number of pixels of that color account for over 30% of the extracted region. Since we are interested more in the hue information in color comparison, the three components in an HSV triplet were weighted by factors of 1.0, 0.2, and 0.05 in similarity comparison.

3.3 Shape Representation

Three factors were considered in choosing a shape descriptor in this study. First, it should be invariant to shifts, scaling, and rotations. Second, the descriptor is ideally in a fixed length for the purpose of comparison. Third, some inner properties, such as holes, need to be addressed. With these considerations in mind, we have chosen to use invariant moments (Hall 1979; Hu 1962) plus a so-called centralized edge distribution histogram for shape representation. A thorough study of shape descriptions can be found in (Mehtre, Kankanhalli and Lee 1997; Scassellati, Alexopoulos and Flickner 1994).

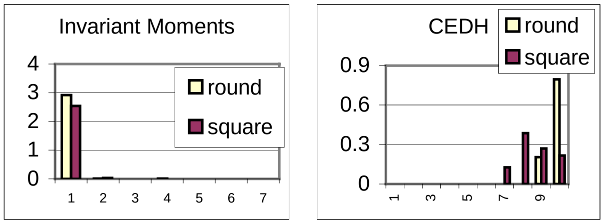
Invariant Moments. The invariant moments method was first proposed by Hu (Hu 1962). The formula used in this study was borrowed from Hall (Hall 1979). An invariant moment set consists of seven shape coefficients calculated from the extracted region of interest. The obtained descriptor is invariant to shifts, scaling, and rotations.

Centralized Edge Distribution Histogram. The invariant moment descriptor is effective in differentiating between irregular shapes while it is fairly insensitive to regular symmetric shapes such as *circles*, *squares*, and *pentagons*. To overcome this shortage, we have proposed to use another shape descriptor called Centralized Edge Distribution Histogram (CEDH), which is defined as follows.

1. Compute the distance d_k from each edge point k to the geometric center of the shape.
2. Create histogram of $d_k / d_{\max}$, where $d_{\max} = \max_k d_k$.
3. Quantize the histogram into 10 units uniformly, corresponding to the 10 normalized distances from an edge point to the center of the region.

The CEDH descriptor is invariant to shape translations, scaling, and rotations. Empirically this measurement is a good complement to the invariant moment descriptor since it is effective in differentiating between symmetric shapes. An example is given in Figure 3, in which the invariant moments and CEDH of two symmetric shapes: *round* and *square*, are compared. The invariant moment descriptor is insensitive to these shapes while CEDH differentiates them quite well.

By combining the two features, a descriptor consisting of 17 real values (7 invariant moments + 10 CEDH) is built. Although the resultant descriptor does not characterize object shapes uniquely, its performance on shape differentiation is, however, empirically satisfactory.



(a) (b)

Figure 3. Comparison of two shape descriptors

4. Concept Category Inferences

The key of scene concept learning is to determine the category type of a label according to the examples observed in learning. Once the category type is determined, the related set of parameters (defined in equation 1) is calculated according to the MLE method and used to represent this label. The category type of a label is estimated using a two level Bayesian inference network, which consist of (and referred to as) *local inference* and *global inference*.

4.1 Local Inference

The aim of local inference is to calculate the probability of the category type of a label of interest according to the visual features obtained from the scene examples received in learning. The term "local" indicates that the category types of other labels are not used for the inference.

The basic idea of local inference is to evaluate the evidence of the observed scene examples of a label agreeing with the format of the feature JPDFs of a given category type defined in equation 1. The network for local inference is given in Figure 4. The root node (Category Type) has three values corresponding to the three category types. The evidence that a certain category type, say category *i*, is supported by the observed scene examples is computed as follows. First, the set of parameters of category *i* defined in equation 1 is calculated according to the examples observed using the MLE method (it is worth noticing that this operation has nothing to do with the one introduced in section 2.2 and the one in the last step in the summary in section 4.3). The resultant set of parameters is used to calculate the evidence that the observed examples having a label of category *i*. Denote an observed example of the label of interest as $\mathbf{x}_k$, where $1 \le k \le N$ with *N* being the number of the observed examples of this label. The evidence that supports the category type *i* is calculated as

$$P(\mathbf{X} \mid CT_i) = \prod_{k=1}^{N} J_i(\mathbf{x}_k) \qquad (2)$$

where $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$; J_i is the feature JPDF with respect to category type *i* discussed above; the term CT_i can take one of the three values of a category type: OBJECT, COLOR, or SHAPE. By applying the Bayesian inference rules (Pearl 1988), the posterior probability P(CT_i | **X**) is calculated as

$$P(CT_i \mid \mathbf{X}) \propto P(\mathbf{X} \mid CT_i)\pi(CT_i) = \prod_{k=1}^{N} J_i(\mathbf{x}_k)\pi(CT_i) \qquad (3)$$

where the prior $\pi(CT_i)$ is set uniformly as 1/3.

Given a label of interest, say label *j*, the output of local inference is a set of posterior probability $P(CT_i \mid \mathbf{X}_j)$ that indicates the category type likelihood of this label given the scene examples observed in learning.

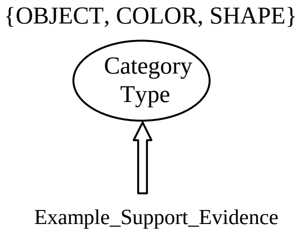


Figure 4. Inference network based on local features only

4.2 Global Inference

The idea of global inference is to adjust the category type probability of a label using the category type information of other labels. The motivation of doing this adjustment is as follows. Given a label, denoted as *a*, that could be a COLOR or OBJECT label according to the output of the local inference network, i.e.,

$$P(\text{OBJECT} \mid \mathbf{X}_a) \approx P(\text{COLOR} \mid \mathbf{X}_a) >> P(\text{SHAPE} \mid \mathbf{X}_a).$$

Meanwhile, there exists a COLOR label, denoted as *b*, that has already been learned (i.e., P(COLOR | $\mathbf{X}_b$) is high) and has the same or similar color mean (μ) as that of label *a*. In this case, it is safe to say that the label *a* is unlikely to be a COLOR label because we assume that each label must represent uniquely a certain semantics of a scene. That is, it is impossible to have one physical color receive two different color labels in learning.

The global inference network is given in Figure 5. The input of the network is the sets of category type probabilities of all the labels calculated from the local inference network, i.e., the set of $P(CT_i \mid \mathbf{X}_j)$, where *i* indexes the category types and $\mathbf{X}_j$ is the set of observed examples of label *j*. The output of the network is a set of adjusted category type probabilities for each label.

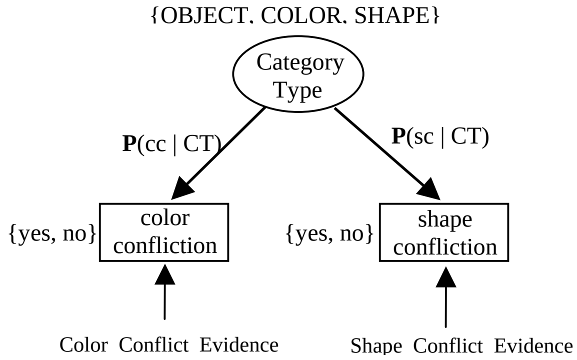


Figure 5. Inference network based on global information

The root node (Category Type) of the inference network is defined the same as that in the local inference network. The color or shape confliction node takes one of two values: *yes* or *no*. The evidence of having a confliction (the case of *yes*) is given by ev_m, which is calculated as

$$ev_m = \max_{k \neq j}\left\{\frac{1}{2}\exp\left(-\left\|\mathbf{f}_m^k - \mathbf{f}_m^j\right\|\right)\cdot P\left(CT_m \mid \mathbf{X}_k\right)\right\} \qquad (4)$$

where *m* indexes the modules of color (1) or shape (2); index *j* refers to this label and *k* is for any other labels; $\mathbf{f}_m$ represents the mean feature (color or shape) vector of a label. The term $P(CT_m \mid \mathbf{X}_k)$ is the probability of label *k* being a color or shape label calculated from the local inference network. According to the definition in (4), the factor ev_m changes from 0 to 0.5, which corresponds to the cases from non-conflicting to conflicting, respectively. Consequently, the evidence of a confliction node taking the value of *yes* (1) or *no* (2) is given by $\left\{e_m^1, e_m^2\right\}$={$ev_m$, 1-$ev_m$}, respectively. The adjusted category type probability is therefore calculated according to (Pearl 1988):

$$P(CT_i \mid \mathbf{e}) \propto \prod_{m=1}^{2} P(\mathbf{e}_m \middle| CT_i)\pi(CT_i) = \prod_{m=1}^{2}\left\{\sum_{l=1}^{2}\left[e_m^l P(c_m^l \middle| CT_i)\right]\pi(CT_i)\right\}$$

Where the prior $\pi(CT_i)$ is the probability $P(CT_i \mid \mathbf{X}_j)$ obtained from the local inference network (equation 3). The conditional probability matrix $\mathbf{P}(cc \mid CT)$ and $\mathbf{P}(sc \mid CT)$ is defined heuristically as

$$\mathbf{P}(cc \mid CT) = \begin{pmatrix} 0.5 & 0 & 0.5 \\ 0.5 & 1 & 0.5 \end{pmatrix}^T, \quad \mathbf{P}(sc \mid CT) = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0.5 & 0.5 & 1 \end{pmatrix}^T$$

By applying the local and global inference engines a set of consistent category probabilities for each label is obtained, which is used for scene information retrieval later. The effectiveness of introducing the global inference engine is discussed further in the experimental section.

4.3 Summary of the Learning Method

We summarize the proposed approach for semantic scene concept learning by an autonomous agent as below:

1. Collect examples of scene labels via the interactions between the agent and the environment. For example, the teacher shows the agent an object once a time and meanwhile assigns a relevant concept label accordingly.
2. Calculate the concept category probabilities for each label using the local inference network (section 4.1).
3. Do global inference based on the category probabilities of all of the labels calculated from local inference (section 4.2).
4. Calculate the parameter sets formulated in equation 1 accordingly using the MLE method. Use the resultant feature JPDF for future scene information retrieval.

5. Experimental Results

The proposed semantic scene concept learning system has been tested on both a simulated and real learning robot.

5.1 Simulations

Our simulations were conducted over a set of artificial shapes, colors and objects of interest. Figure 6 displays the set of artificial shapes used for learning. Nine color labels were defined consisting of *red*, *yellow*, *orange*, *light blue*, *dark blue*, *green*, *pink*, *purple*, and *black*. Color samples were collected from the palette in MS windows according to humans' perceptual judgments. 40 virtual objects were defined, each of which consisted of 1 to 4 combinations of the predefined colors and shapes (each combination corresponds to a possible view of the virtual object). Overall a total of 63 (14 shapes, 9 colors, 40 objects) scene labels were defined.

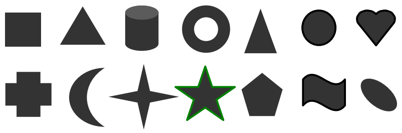


Figure 6. Artificial shapes used in simulations

The learning process was simulated as follows. A virtual object, along with a virtual view, was selected randomly one at a time. According to the view hypothetically observed, the system issued randomly one of the 63 concept labels that correctly described the current scene. For example, if a scene contains a red square, the candidate labels are *red*, *square*, and the name of the corresponding virtual object. To make the learning more close to real, the "observed" color and shape features were perturbed at each "observation". For colors, the perturbation was to add Gaussian noise to the RGB values of a registered color. For shapes, a perturbing affine transform defined as

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \tau + \varepsilon_1 & \varepsilon_2 \\ \varepsilon_3 & \tau + \varepsilon_4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \qquad (5)$$

was used, where (α, β) is a pair of random shifting factors; τ is a random scalar between 0.5 and 2, and ε_k are small random perturbing values.

The learning performance was evaluated in terms of scene label recalls. Specifically, the agent was presented with an arbitrary view of an arbitrary object and was asked to retrieve all of the learned scene labels that best describe the current scene. Figure 7 displays the recalls with a varying size of training examples with and without using the global inference (GI) engine. The statistics were collected over 100 independent runs, each of which consisted of 50 random testing views. The resultant curves show clearly the contribution of the global inference engine, with which the label recalls were significantly boosted.

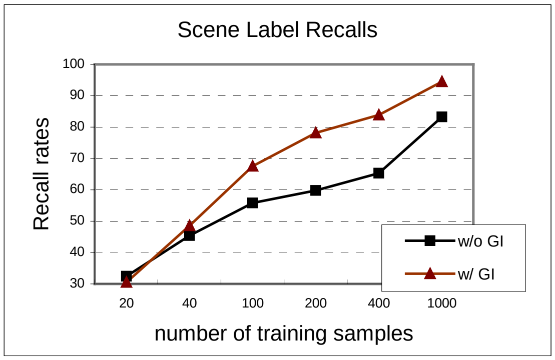


Figure 7. Recalls of scene label retrievals

5.2 Implementation on a Real Robot

We have used the proposed method to teach a real robot to learn labels of colors (COLOR), shapes (SHAPE), and the names of the objects of interest (OBJECT). Figure 8 displays the set of objects used in the experiments. Each object consists of one or more colors. The shape of an object may change from different perspectives. During the training, the teacher picked an object to show the robot and meanwhile assigned a relevant label (using the keyboard) according to the judgment of the teacher.

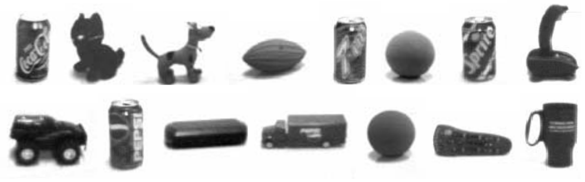
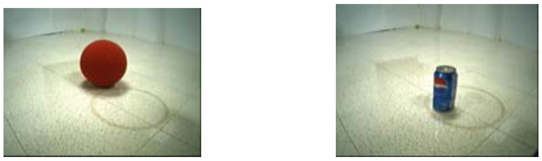


Figure 8. The set of objects used in learning

The testing phase was similar to that in the simulation, in which a total of 22 labels (6 colors, 3 shapes, and 13 objects) were covered. Table 1 compares the recalls after 40 training examples (data were collected in additional 40 testing examples). Encouragingly, the proposed method received a quite satisfactory recall after a small period of training. An example of training and testing is given in Figure 9, in which (a) displays the scenario that the agent received a label *red* when a red ball was presented in training. In testing (figure (b)), the agent successfully retrieved all of the labels, i.e., *blue*, *can*, and *Pepsi*, that were related to the scene containing a Pepsi can.

Learning mode	W/ GI	W/O GI
Recalls	86.5%	65.2%

Table 1. Recalls in real robot learning in 40 testing examples.



Label received: *red*

(a)

Retrieval: *blue*, *can*, *Pepsi*

(b)

Figure 9. Example of training (a) and testing (b)

6. Conclusion and Future Work

The objective of this research is to explore a new world in vision study, i.e., learning the semantics of scene labels by an agent. The learning task is formulated in a way of associative memory where the aim is to discover the associations between scene labels and relevant visual features. The capability of semantic-level scene concept learning is crucial for intelligent human-computer interaction.

A generic model for semantic scene concept learning is proposed in this study, based on which a small-sized concept learning system was developed using a two-level Bayesian inference network. While the assumption and setup of our preliminary study were to some extent artificial and simple, the experimental result has displayed the effectiveness of the proposed approach in catching the semantics of scene labels by an agent.

Based on our preliminary work, the future research will be carried out in the following two directions:

1. Develop an integrated learning approach for general-purpose scene concept understanding. So far, the agent is able to learn scene labels that come from several fixed categories whose domain knowledge, i.e., the format of feature JPDFs, is known. Although the proposed learning method is expansible, it is more desirable to have an agent be able to learn adaptively and autonomously.
2. Explore richer visual features. Our current study is focused on static attributive features such as colors and shapes. This restriction imposes limits on many scene-understanding tasks. As another focus in future study, we will investigate richer scene features, including both spatial and temporal visual patterns, to address more comprehensive semantics of a natural scene.

References

Belongie, S., Malik, J., and Puzicha, J., 2002. Shape Matching and Object Recognition Using Shape Context. *PAMI*.

Duda, R., Hart, P., and Stork, D., 2001. Pattern Classification. 2ed, John Wiley & Sons, 85-89.

Duygulu, P., Barnard, K., Freitas, N., and Forsyth, D., 2002. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. 17th ECCV 97-112.

Faugerous, O, 1983. Fundamentals in Computer Vision. Cambridge University Press.

Gorniak, P., and Roy, D., 2004. Grounded Semantic Composition for Visual Scenes, *Journal of Artificial Intelligence Research*, 21: 429-470.

Hall, E., 1979. Computer Image Processing and Recognition. Academic Press.

Hu, M., 1962. Visual pattern recognition by moment invarients. *IRE Transactions on Information Theory* 8:179-187.

Mehtre, B., Kankanhalli, M., and Lee, W., 1997. Shape Measures for Content Based Image Retrieval: A Comparison. *Information Processing & Management* 33(3): 319-337.

Mel, B., 1997. SEEMORE: Combining Color, Shape, and Texture Histogramming in a Neurally-Inspired Approach to Visual Object Recognition. *Neural Computing* 9(4): 777-804.

Mori, Y., Takahashi, H., and Oka, R., 1999. Image-to-word transformation based on dividing and vector quantizing images with words. *1st Int'l Workshop on Multimedia Intelligent Storage and Retrieval Management*.

Murase, H., and Nayar, S., 1995. Visual Learning and Recognition of 3-D Objects from Appearance. *International Journal of Computer Vision* 14(1): 5-24.

Pearl, J., 1988. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Press.

Scassellati, B., Alexopoulos, S., and Flickner, M., 1994, Retrieving images by 2D shape: a comparison of computation methods with human perceptual judgments. *Proc. of Spie - the Int'l society for Optical Engineering,* (2185): 2-14.

Selinger, A., and Nelson, R., 1999. A Perceptual Grouping Hierarchy for Appearance-based 3D Object Recognition. *Computer Vision and Image Understanding.* 76(1): 83-92.