Semi-Supervised Learning and Feature Evaluation for RGB-D Object Recognition

Yanhua Cheng, Xin Zhao, Kaiqi Huang*, Tieniu Tan

Center for Research on Intelligent Perception and Computing
National Laboratory of Pattern Recognition
Institute of Automation Chinese Academy of Sciences, Beijing 100190, China

Abstract

With new depth sensing technology such as Kinect providing high quality synchronized RGB and depth images (RGB-D data), combining the two distinct views for object recognition has attracted great interest in computer vision and robotics community. Recent methods mostly employ supervised learning methods for this new RGB-D modality based on the two feature sets. However, supervised learning methods always depend on large amount of manually labeled data for training models. To address the problem, this paper proposes a semi-supervised learning method to reduce the dependence on large annotated training sets. The method can effectively learn from relatively plentiful unlabeled data, if powerful feature representations for both the RGB and depth view can be extracted. Thus, a novel and effective feature termed CNN-SPM-RNN is proposed in this paper, and four representative features (KDES [1], CKM [2], HMP [3] and CNN-RNN [4]) are evaluated and compared with ours under the unified semi-supervised learning framework. Finally, We verify our method on three popular and publicly available RGB-D object databases. The experimental results demonstrate that, with only 20% labeled training set, the proposed method can achieve competitive performance compared with the state of the arts on most of the databases.

Keywords: RGB-D, object recognition, feature representation, feature evaluation, semi-supervised learning.

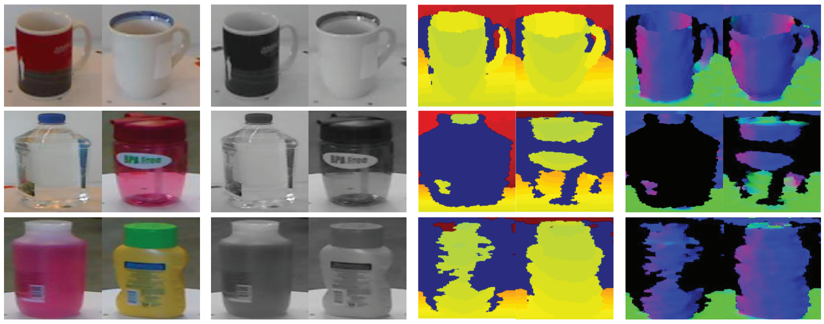
*Corresponding author.
Email addresses: yh.cheng@nlpr.ia.ac.cn (Yanhua Cheng), xzhao@nlpr.ia.ac.cn (Xin Zhao), kqhuang@nlpr.ia.ac.cn (Kaiqi Huang), tnt@nlpr.ia.ac.cn (Tieniu Tan)

1. Introduction

Recently, RGB-D data has attracted great interest in computer vision and robotics community with the advent of new depth sensors, such as Kinect. The Kinect-style depth cameras are capable of providing high quality synchronized images or videos of both color and depth, which represent an opportunity to dramatically improve the performance of many vision problems, e.g., object recognition [5, 6], detection [7, 8, 9], tracking [10, 11, 12], SLAM [13, 14] and human activity analysis [15, 16]. This is mainly because that depth information has many extra advantages: being invariant to lighting and color variations, allowing better separation from the background and providing pure geometry and shape cues. Furthermore, many RGB-D datasets [5, 6, 13, 14, 15, 17, 18] have been published for public use to promote the development of such research areas.

This paper mainly focuses on object recognition, which is a fundamental problem in computer vision and pattern recognition. Although many methods [1, 2, 3, 4, 19] have been presented to promote RGB-D object recognition, they chiefly aim at extracting effective features from the novel RGB-D data and using a supervised learning model to achieve good classification performance. However, supervised learning models always require for large amount of manually labeled data. The collection of enough labeled training set is an expensive and difficult task. Thus, it is important to get rid of this problem by utilizing relatively plentiful and convenient unlabeled RGB-D data.

With the ability to handle the unlabeled data, a semi-supervised learning framework is proposed in this paper by considering the two distinct views of RGB-D data effectively. Although there are many successful semi-supervised learning algorithms in the literature, e.g., self-training [20, 21, 22], co-training [23, 24] and graph-based methods [25, 26, 27], we are especially interested in the co-training method because of its unique advantages over the RGB-D data: co-training was theoretically proved to be very successful in combining the labeled and unlabeled data under two strong assumptions (including "conditional independence given the label")) in [23], then the work [24] proved that much weaker assumptions were sufficient to guarantee co-training, when given appropriately strong PAC-learning algorithms on each view. Intuitively, the two weaker assumptions can be described as follows: (1) Each example contains two distinct views, and each view provides sufficient information to determine the label of the example; (2) The two views should not be too highly correlated. It means that, there



(a) RGB (b) grayscale (c) depth (d) surface normals

Figure 1: Four modalities including RGB, grayscale, depth, and surface normals are alternative to capture cues for RGB-D object recognition. RGB images and depth maps are directly imaged by Kinect-style cameras, while grayscale images and surface normals are computed from the RGB images and depth maps respectively. In the figure, each row consists of two instances from the same category (Examples are from the Washington RGB-D object dataset [6]).

should exist some examples which can be confidently recognized by one view but not by the other view (or vice versa) to make the co-training algorithm work effectively. RGB-D data meets the two assumptions very well. Firstly, RGB-D data contains two distinct views, RGB and depth. Both of them can provide useful cues for object recognition: RGB images can describe rich color, texture and appearance information for the object, while depth maps can sketch pure geometry and shape cues. Secondly, the image capturing modes of RGB (e.g., RGB cameras) and depth (e.g., infrared cameras) are very different, guaranteeing the independence of the two views.

Given two distinct views (RGB and depth) for each example, the key to the success of co-training is to obtain effective feature representation for each view. Thus a powerful feature CNN-SPM-RNN will be proposed in this paper based on the feature CNN-RNN [4]. CNN-RNN combines a single convolutional neural network (CNN) and multiple recursive neural networks (RNN [28]) to learn high-level features for each RGB-D object. Since learning the optimal structure of each RNN tree from the raw data is highly time consuming as described in [28], CNN-RNN utilizes fixed-tree RNN structure to hierarchically aggregate the CNN responses very efficiently. However, the fixed-tree RNN requires for the fixed-size of the inputs by simply cropping or warping all the images, which may degrade the recognition performance after such artificial processing. Inspired by the pioneer work [29], which applied a spatial pyramid pooling layer (SPM [30]) to the supervised deep learning model [31] to adapt the model for arbitrary sizes of inputs, we extend its core idea to the unsupervised CNN-RNN feature learning model and design a new feature learning structure, termed CNN-SPM-RNN. Towards SPM layer, the main differences between CNN-SPM-RNN and the work [29] are twofold: (1) A pooling layer with different pyramid partitions in [29] is sufficient to guarantee the success of the **supervised** deep learning model, benefiting from the back-propagation of errors and the fine-tuning of filters. However, we empirically find that a single pooling layer can even make the **unsupervised** CNN-RNN model worse, probably because the low-level convolutional responses cannot capture local object structures very effectively after pooling. Thus a feature coding layer is added to encode the convolutional responses and high-level feature responses are obtained to represent local information powerfully. Then the pooling layer is performed to result in fixed-scale feature maps for the fixed-tree RNNs. (2) Compared to the 2D spatial pyramid pooling for the RGB modality in [29], 3D spatial pyramid pooling is utilized for the depth modality to effectively capture shape cues of objects. The details are given in Section 2.2. Further more, we find that two additional modalities, grayscale images and surface normals, can largely benefit view representation, as shown in Fig. 1. We introduce a unified feature evaluation framework by combining RGB and grayscale to capture visual appearance (i.e., the RGB view), while depth and surface normals to capture shape cues (i.e., the depth view) for all the representative features, including KDES [1], CKM [2], HMP [3], CNN-RNN [4] and CNN-SPM-RNN.

An early version of our work was presented in [32] to explore co-training for RGB-D object recognition. In this paper, we extend [32] in the following aspects: developing a more powerful feature termed CNN-SPM-RNN based on [4, 32], introducing a unified framework to fairly evaluate all the representative features, and presenting a wide array of experiments to demonstrate the effectiveness of the proposed semi-supervised method with powerful RGB-D features.

The major contributions of this paper are summarized as follows.

- Propose a complete and systematic semi-supervised learning framework for RGB-D object recognition using co-training. We theoretically analyse that the framework can take full advantage of the characteristics of the new RGB-D data, and significantly benefit object recognition by learning from large amount of unlabeled RGB-D data.

- Present a novel feature CNN-SPM-RNN to effectively represent RGB-D data. To the best of our knowledge, this is the first work to successfully apply SPM

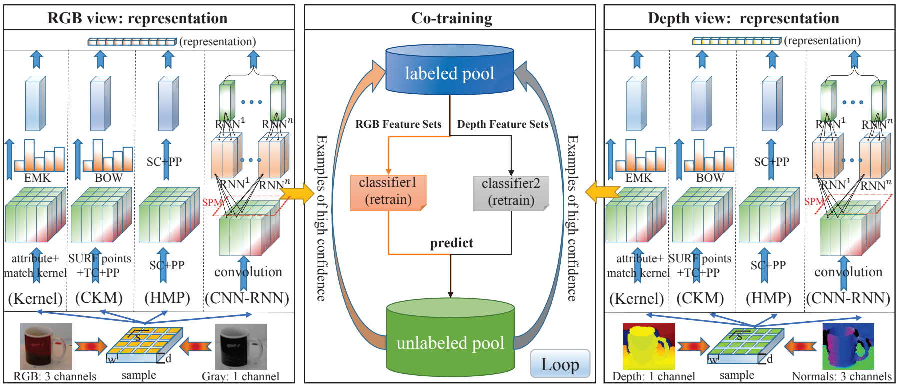


Figure 2: Our semi-supervised learning framework for RGB-D object recognition. Firstly, we extract features to represent the RGB and depth view of each object respectively. Then we employ co-training to iteratively learn from the unlabeled data using the two distinct feature sets. In the figure, TC means triangular coding, PP means pyramid pooling, SC means sparse coding (see Section 2.3 for details).

to the unsupervised deep learning model to address the problem of cropping or warping. The core idea is inspired from the pioneer work [29], which utilized SPM layer in the supervised deep learning model and yielded impressive results.

- Analyse and Evaluate most representative RGB-D features in an unbiased way by utilizing four data modalities, including RGB, grayscale, depth and surface normals, which can provide a meaningful guideline how to best represent the new RGB-D data.

The rest of this paper is organized as follows: Section 2 proposes our semi-supervised learning framework for RGB-D object recognition, including the semi-supervised learning method based on co-training, the feature CNN-SPM-RNN, and a unified framework for feature evaluation. Section 3 empirically evaluates and ranks all the representative features in an unbiased way, and shows the comparison of our semi-supervised method with the state of the arts on several public RGB-D object databases. Finally, Section 4 concludes the paper and discusses the future work.

2. Our Semi-supervised Framework

As shown in Fig. 2, our semi-supervised learning framework is proposed for RGB-D object recognition. There are three modules in the framework: (1) feature representation for the RGB view; (2) feature representation for the depth view; and (3) exploiting co-training to utilize a small set of labeled data and large amount of unlabeled data.

The core idea of the framework is to improve the two classifiers trained on the two distinct feature sets iteratively by co-training. Thus how to extract effective feature representation for each view is the fundamental step.

In the following subsections, we first prove that co-training can succeed in learning from unlabeled RGB-D examples, and propose the co-training algorithm for RGB-D object recognition. Then we introduce our feature CNN-SPM-RNN, followed by a unified framework to evaluate recent state-of-the-art features for RGB-D objects.

2.1. Semi-supervised Learning

We employ co-training as our semi-supervised learning method to learn from the unlabeled RGB-D data, as shown in Fig. 2. Firstly, Two assumptions to guarantee the success of learning with co-training are introduced. Then a specific co-training algorithm for RGB-D object recognition is proposed.

2.1.1. Theoretical Assumptions

Some notations are given as follows: Let D denote the distribution over the feature space $F = F_1 \times F_2$, where F_1 and F_2 correspond to two different views of an example. Assume F^+ and F^- are the positive and negative regions of F respectively (for simplicity we consider binary classification here), and Let c be the target function. Then for $i \in \{1, 2\}$, we define $F_i^+ = \{f_i \in F_i : \ c_i(f_i) = 1\}$ and $F_i^- = F_i - F_i^+$. In order to bootstrap co-training, a initial labeled set for the two views $S_1^0 \subseteq F_1^+$ and $S_2^0 \subseteq F_2^+$ are provided. During the iterative learning procedure of co-training, a hypothesis h_i is devised as a subset of F_i, where $f_i \in h_i$ means that h_i is confident that f_i is positive, and $f_i \notin h_i$ means that h_i has no opinion.

The research [24] proved it was sufficient for co-training to succeed, when given the two assumptions on the underlying data distribution:

- The learning algorithm for each view is able to learn from positive data only.
- The marginal distribution D^+ is ϵ-expanding ($\epsilon > 0$).

The first assumption means that, $\forall D_i^+$ over F_i^+, given access to examples from D_i^+, each learning algorithm is able to produce a hypothesis h_i such that $Pr(error_{D_i^+}(h_i) \leq \epsilon) \geq 1 - \delta$, where $\epsilon, \delta > 0$. This can be thought of as predicting the examples either "positive with confidence" or "has no opinion". According to [24], this assumption is easy to fulfill in practice if the positive class is cohesive and the negative class is not;

Algorithm 1 Co-Training Algorithm for RGB-D Object Recognition

Input:

$\mathbf{F_{RGB}}$: RGB feature set; $\mathbf{F_{depth}}$: depth feature set;
$\theta_{\mathbf{RGB}}$: a confidence threshold for RGB feature set;
$\theta_{\mathbf{depth}}$: a confidence threshold for depth feature set;
$\mathbf{L}$: labeled pool; $\mathbf{U}$: unlabeled pool;
$\mathbf{I}$: the maximum number of iteration rounds

Output:

$\mathbf{C_{RGB}}$:RGB classifier; $\mathbf{C_{depth}}$: depth classifier

```
i ← 0;
repeat
    C_RGB ← train(F_RGB, L);
    C_depth ← train(F_depth, L);
    C_RGB → predict(F_RGB, U), for each predicted class c_j, choose |n_j| most
        confident examples and add them to L, ∀n_j, Score(n_j) ≥ θ_RGB;
    C_depth → predict(F_depth, U), for each predicted class c_j, choose |n_j| most
        confident examples and add them to L, ∀n_j, Score(n_j) ≥ θ_depth;
    i++;
until i > I or U is empty
return C_RGB and C_depth;
```

The second assumption can be interpreted as the following definition:

Definition D^+ is ϵ-expanding if for any $S_1 \subseteq F_1^+,\ S_2 \subseteq F_2^+$, we have

$$\Pr(S_1 \oplus S_2) \geq \epsilon \min\left[\Pr(S_1 \wedge S_2), \Pr(\bar{S}_1 \wedge \bar{S}_2)\right].$$

where $Pr(S_1 \wedge S_2)$ denotes the probability mass on examples that are confidently predicted as positive region by both views, and $Pr(S_1 \oplus S_2)$ denotes the probability mass on examples for which we are confident about just one view. Note that ϵ-expanding is necessary to guarantee co-training will succeed, because if S_1 and S_2 are confident sets and do not expand, then we might never see the expected situation that examples for one hypothesis could help the other.

2.1.2. Co-training Algorithm

An intuitive interpretation of co-training is as follows: Firstly, two initial classifiers over the respective views are trained on a small labeled sample. Then each classifier is used to label the confident examples for the other classifier, for which these examples can be seen as random training instances. In this case, each classifier can benefit from the additional examples by the other one and improve its classification accuracy in every rounding training.

We propose our co-training algorithm for RGB-D object recognition (multi-class

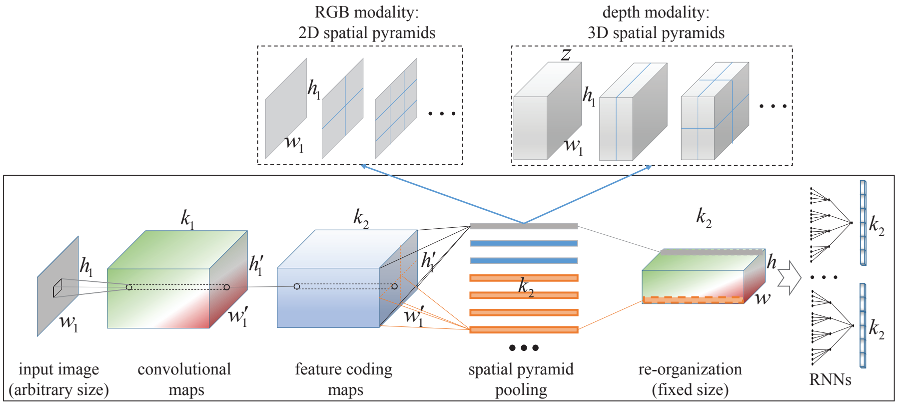


Figure 3: An overview of the feature learning structure of CNN-SPM-RNN. The SP-M layer in this paper consists of feature coding, spatial pyramid pooling, and re-organization, which can input convolution feature maps with arbitrary sizes (e.g., $w_1' \times h_1' \times k_1$, where $w_1' \times h_1'$ is the size of each feature map, and k_1 is the number of feature maps), and then output fixed-scale feature maps (i.e., w, h and k_2 are fixed to the same for all inputs.

classification) in Algorithm 1. Firstly, we extract feature representations F_{RGB} and F_{depth} for the RGB and depth view respectively. Then we train two linear SVM classifiers C_{RGB} and C_{depth} based on a small set of initial labeled examples L using the two feature sets. C_{RGB} and C_{depth} are applied to predict the examples from the unlabeled training sets U separately. For each classifier, $|n_j|$ most confidently predicted instances of each class whose scores are higher than a threshold will be transferred from U to L in every iteration. Generally, we assign $|n_j|$ a small value and keep it the same for all the classes. The algorithm runs until the iteration number reaches the given maximum threshold or all the unlabeled examples in U are labeled. The outputs of the algorithm are the updated classifiers C_{RGB} and C_{depth}.

At the inference time, C_{RGB} and C_{depth} are combined to predict the category of the given example based on their classification scores:

$$c = \arg_{c_i \in \chi} Max(\alpha S^{c_i}_{C_{RGB}} + (1-\alpha) S^{c_i}_{C_{depth}}) \tag{1}$$

where χ is the label set of all the categories, $S^{c_i}_{C_{RGB}}$ and $S^{c_i}_{C_{depth}}$ are predicted scores of category c_i for an given example, and α is the coefficient to control the contribution of each view.

2.2. CNN-SPM-RNN

CNN-SPM-RNN is built on the unsupervised feature learning structure of CNN-RNN [4]. CNN-RNN mainly consists of three steps: resizing all the images to the same scale, extracting low level feature for each image by a single convolutional layer, and finally applying multiple fixed-tree RNNs to learn high order feature representation based on the low level feature responses. Although CNN-RNN can learn powerful features from the raw data, such artificial processing of the first step, i.e., resizing all the images to the same scale by simply cropping or warping the images, may degrade the performance of the learned features. In order to adopt CNN-RNN model for images of arbitrary sizes, we replace the first step of CNN-RNN by a SPM layer, which is composed of three steps: feature coding, spatial pyramid pooling and re-organization, as showed in Fig. 3. To fairly compare CNN-SPM-RNN with CNN-RNN, the parameters of the single-layer CNN and multiple RNNs are kept the same as the work [4], i.e., $k_1 = 128$ filters with 9×9 size are learned for the single-layer CNN, the input fixed-scale feature maps for each RNN are $27 \times 27 \times 128$-dimensional ($w = 27, h = 27, k_2 = 128$). Now we describe the details of each step of the proposed SPM layer.

Feature Coding. The goal is to learn high-level local features to represent objects more powerfully, compared with the low-level convolutional descriptors. First, a codebook $\{c_1, c_2, ..., c_{k_2}\}$ ($k_2 = 128$, $c_i \in \mathbb{R}^{k_1=128}$) is learned by k-means clustering over the sampled convolutional descriptors. Second, each convolutional descriptor $x \in \mathbb{R}^{k_1=128}$ is encoded by the codebook with triangular voting [33]:

$$\begin{aligned} & f(x) = (f_{c_1}(x), f_{c_2}(x), ..., f_{c_{k_2}}(x)), \\ & s.t. \;\; f_{c_i}(x) = max(0, \mu - \|x - c_i\|_2^2), \\ & \quad \mu = \tfrac{1}{k_2} \textstyle\sum_{i=1}^{k_2} \|x - c_i\|_2^2. \end{aligned} \tag{2}$$

where $f(x) \in R^{k_2=128}$.

Spatial Pyramid Pooling. 2D and 3D spatial pyramid pooling are employed for the RGB and depth modality, respectively. For the 2D spatial pyramid pooling, the partitions are constrained in the two-dimensional image space. While for the 3D spatial pyramid pooling, the partitions are performed in the three-dimension depth space. See Fig. 3 for an intuitive understanding. The work [34] also showed that 3D spatial pyramids were necessary to represent the depth modality. In this paper, we set the number of the pyramid bins as $27 \times 27 = 729$, in order to obtain the same size of the fixed-scale feature maps as [4] for a fair comparison. For each bin, max pooling is used to aggregate the neighboring features to a 128-dimensional feature vector.

Re-Organization. After spatial pyramid pooling, the convolutional responses with arbitrary sizes are transformed to a fixed number of feature vectors. We re-organize all the feature vectors to a 3D feature map ($\in \mathbb{R}^{27\times27\times128}$) with a fixed order. Finally, the 3D feature map is input to multiple RNNs to learn the global feature representation as [4].

We employ CNN-SPM-RNN to extract features for each modality of RGB (2D spatial pyramids), grayscale (2D spatial pyramids), depth (3D spatial pyramids) and surface normals (2D spatial pyramids), respectively. For each object, the RGB feature and grayscale feature are concatenated to represent the appearance information, while depth feature and surface normal feature are combined to capture shape cues.

2.3. Feature Analysis and Evaluation

Various features have already been developed for RGB-D object recognition. In this section, we introduce four state-of-the-art features: kernel descriptors (KDES) [1], convolutional k-means descriptors (CKM) [2], hierarchical matching pursuit (HMP) [3], and convolutional-recursive neural networks (CNN-RNN) [4], which are more discriminative and robust than the popular orientation histogram features, such as SIFT [35] and spin images [36]. In order to compare these features with the CNN-SPM-RNN, we analyse the characteristics of them first, and then propose a unified framework to extract all these features effectively for an unbiased evaluation.

2.3.1. Feature Analysis

The four representative features: KDES [1], CKM [2], HMP [3] and CNN-RNN [4], employ very different methods to extract features from the raw data, compared to the handcrafted features utilized in the baseline work [6]. Furthermore, they take advantage of different data modalities among RGB images, grayscale images, depth maps and surface normals to capture cues for object recognition, as shown in Table 1. The analysis of the above features is shown as follows:

The baseline work [6] extracts **a set of handcrafted features** to represent the two distinct views of RGB-D objects. To capture the visual appearance of the RGB view, they extract SIFT descriptors over grayscale images, texton histograms [37] and color histograms over the RGB images. The shape of the depth view is represented by spin images computed from depth maps and surface normals. Regardless of their effectiveness, these well tuned handcrafted features are hard to design and only can capture a

Features	RGB View		Depth View	
	RGB	Grayscale	Depth	Surface Normals
Handcrafted features [6]	√	√	√	√
KDES [1]	√	√	√	√
CKM [2]	√	–	√	–
HMP [3]	√	√	√	√
CNN-RNN [4]	√	–	√	–
CNN-SPM-RNN	√	√	√	√

Table 1: Different data modalities are exploited to capture cues for object recognition for different methods. Generally, RGB and grayscale images can capture visual appearance of the RGB view, while depth maps and surface normals can capture shape information of the depth view.

small set of recognition cues from raw data. For example, SIFT is able to capture some sort of edge information while ignores color information; Spin images are extended to 3D objects analogous to SIFT over 2D images, but also has limited capability to capture useful shape or geometry information.

KDES [1] provides a generalized way to extend orientation histogram features like SIFT to a broad class of similar feature patterns. The previous work [38] has already shown that the well-designed SIFT features are equivalent to a certain type of match kernel over image patches. Thus, it is very convenient to design a set of kernel descriptors on top of various attributes, including 3D shape, physical size, edges, gradients, etc.

CKM [2] adapts single-layer feature learning networks based on k-means clustering for 2D images [33] to RGB-D data. To keep the feature learning process as effective as [33], CKM takes the depth channel as the fourth channel of the RGB channels and directly learns features from the four channels. By using the state-of-the-art image pre-processing and feature encoding of [33], CKM can obtain useful translational invariance of low-level features from raw data such as edges, and can be robust to small deformations of objects. However, without information of grayscale images and surface normals, the performance of CKM is restricted a lot for object recognition.

HMP [3] constructs a two-layer architecture to generate features over complete RGB-D images based on sparse coding. It can discover low-level structures such as edges at the first layer, and high-level structures such as shapes and object parts at the second layer. HMP learns features from each data modality, then combines the RGB

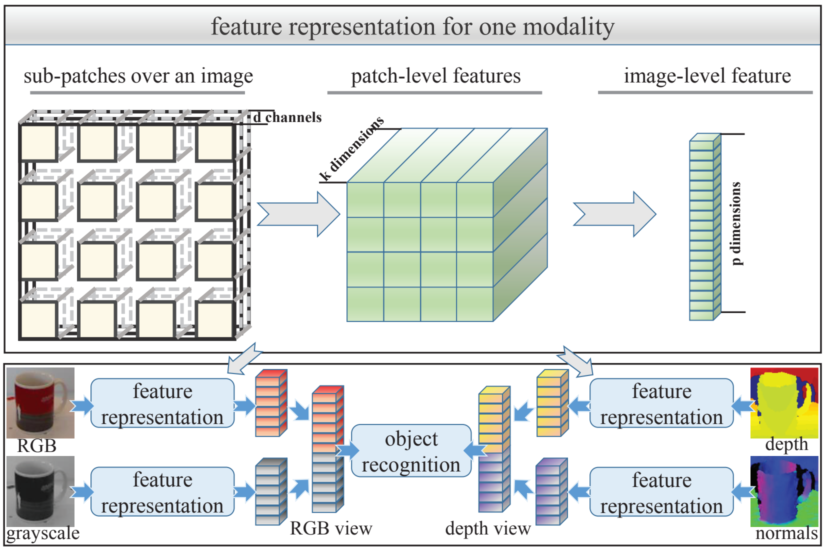


Figure 4: A unified framework to represent the RGB and depth view of RGB-D objects. For each type of features, we extract them from four data modalities, respectively. Then combine the RGB and grayscale features to capture visual appearance of the RGB view, and the depth and normal features to capture shape of the depth view.

and grayscale features to represent the visual appearance of the RGB view, and captures the shape cues by concatenating the depth and surface normal features.

CNN-RNN [4] is a deep feature learning model based on a combination of convolutional and recursive neural networks. The single CNN layer can learn low-level translationally invariant features which are assembled by multiple RNNs [28] to construct high order representation. Similar to CKM, CNN-RNN only makes use of RGB images and depth maps for object recognition.

Both the baseline work and KDES utilize manually designed features, while CK-M, HMP, CNN-RNN and CNN-SPM-RNN belong to unsupervised feature learning methods.

2.3.2. Unbiased Feature Evaluation

To obtain an unbiased evaluation for all the above features, we propose a unified framework to represent RGB-D objects by adapting them to the four data modalities, as shown in Fig. 4. For each type of features, the RGB and grayscale images are used to capture visual appearance of the RGB view, and the depth and surface normal images are exploited to capture shape cues of the depth view. We can obtain more powerful view representation by utilizing additional information of grayscale and surface normals, compared to those only based on RGB and depth in their original papers.

Specifically, we are capable of extracting CKM descriptors from each data modality respectively following the same process of [2] over RGB-D images. Then, we learn the image-level features for each data modality using a bag-of-words model with spacial pyramid pooling [30]. Finally, following the framework, CKM features from the four data modalities will be combined to represent the RGB view and depth view respectively. To distinguish our CKM features from the original paper [2], we call them **enhanced CKM**; Similarly, we employ CNN-RNN to learn features not only from the RGB and depth images but also from the grayscale and surface normal images. Then the RGB and grayscale features are combined to describe the RGB view, while the depth and surface normal features are concatenated to represent the depth view. We call our CNN-RNN features **enhanced CNN-RNN**. Since the baseline work, KDES, HMP and CNN-SPM-RNN have already taken advantage of the four data modalities for object recognition, we keep them the same with the original papers.

3. Experiments

Our experiments are carried out on three publicly available RGB-D object datasets. On the first challenging Washington RGB-D Object Database [6], we evaluate all the representative features in an unbiased way and compare the performance of the proposed semi-supervised method with the state of the arts. On the other two datasets, we further verify the effectiveness of the semi-supervised learning method. All the experimental codes including the introduced features and the semi-supervised learning method are released at the website http://www.openpr.org.cn/.

We follow the unified framework in Section 2.3.2 to extract all the introduced features for the RGB view (RGB + grayscale) and the depth view (depth + normals). The experimental settings of each feature extraction method are as follows:

KDES: To construct image-level features for each kind of kernel descriptors, we follow the process of [1] to obtain high performance, which considers $[1 \times 1, 2 \times 2, 3 \times 3]$ pyramid subregions, and use EMK [39] with 500 basis vectors learned by k-means on 400,000 kernel descriptors sampled from training images. The resulting dimensionality per kernel descriptor based image representation is $(1 + 4 + 9) \times 500 = 7000$, then reduced to 1000 using principal component analysis.

Enhanced CKM: To guarantee the feature learning method effective and successful as [2], a bag-of-words model is exploited to construct image-level features for the

CKM descriptors extracted from each data modality. We learn 1000 codewords using k-means on 400,000 descriptors and use average pooling with spacial pyramids $[1 \times 1, 2 \times 2, 3 \times 3]$ to compute the feature responses. The final dimensionality of per image-level feature representation is 14,000.

HMP: HMP can directly learn the image-level features from each data modality. Note that we only aggregate the responses of the patch features at the second layer instead of a jointly pooling [3], because the dimensionality of the jointly pooling is up to 36,050 (grayscale, depth) or 58,100 (RGB, normals) per object, which requires too much memory and computing time. Our processing can reduce the dimensionality of all the four modalities' representations to 7,000 and can obtain approximate performance.

Enhanced CNN-RNN: Similar to [4], for each data modality (resized to the fixed-scale, e.g., 148×148), we learn 128 filters by k-means clustering over the patches, and use 64 randomly initialized RNNs to compose the convolutional responses to the final image representation with $128 \times 64 = 8192$ dimensions.

CNN-SPM-RNN: CNN-SPM-RNN is proposed to extract powerful features from the raw data (without cropping or warping). For a fair comparison, the parameters of the single-layer CNN and multiple RNNs are kept the same with the CNN-RNN model [4]. Similarly, to obtain the same size of the fixed-scale feature map, the size of the codebook is set to $k_2 = 128$, and the number of the pyramid bins is set to $27 \times 27 = 729$. Since there are a lot of selectable ways to partition the image to collect the equal number of bins, we simply choose one without fine-tuning. The configurations are given below.

2D spatial pyramids: $\{3 \times 3, 8 \times 8, 16 \times 16, 20 \times 20\}$

3D spatial pyramids:

$$\begin{aligned}
&\{1 \times 1 \times 1, 1 \times 1 \times 2, 1 \times 1 \times 4, 1 \times 1 \times 8, 1 \times 1 \times 16, 1 \times 1 \times 32, 1 \times 1 \times 36,\\
&1 \times 3 \times 1, 1 \times 3 \times 2, 1 \times 3 \times 4, 1 \times 3 \times 8, 1 \times 3 \times 16, 1 \times 3 \times 32,\\
&3 \times 1 \times 1, 3 \times 1 \times 2, 3 \times 1 \times 4, 3 \times 1 \times 8, 3 \times 1 \times 16, 3 \times 1 \times 32,\\
&2 \times 2 \times 1, 2 \times 2 \times 2, 2 \times 2 \times 4, 2 \times 2 \times 8, 2 \times 2 \times 16, 2 \times 2 \times 32\}.
\end{aligned}$$

Finally, the re-organization 3D feature map of each modality of each object is input into the fixed-tree RNNs, and composed to the global feature representation with $128 \times 64 = 8192$ dimensions.

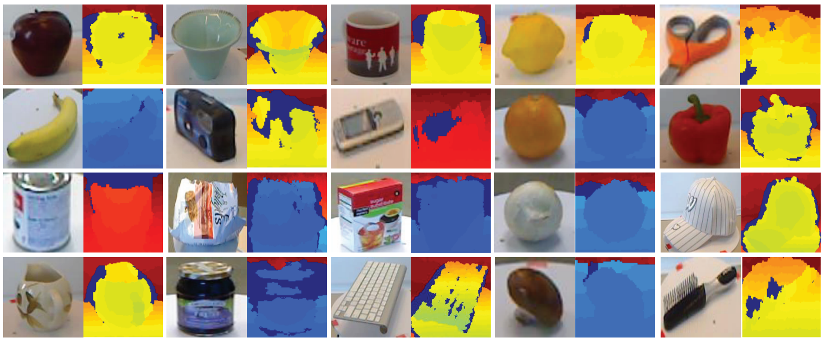


Figure 5: Object examples from the Washington RGB-D object database. Each object (RGB + depth) shown here belongs to a different category.

3.1. Washington RGB-D Object Database

The first experimental database is a large-scale hierarchical multi-view RGB-D object dataset [6]. The database consists of a total of 207,920 RGB-D images containing 300 physically distinct everyday object instances (see Fig. 5). All the instances are grouped into 51 categories. Each object instance is imaged from three viewing heights (30°, 45° and 60° above the horizon) while it rotates on a turntable, resulting in roughly 600 images per instance. We subsample every 5th frame for each instance and reduce the size of the total database to 41,877 RGB-D images.

This paper focuses on category recognition. The dataset settings for supervised learning and our semi-supervised learning are given as below: (1) for supervised learning, we follow the setting in [6], which provides 10 random splits to generate training and test sets from the database. For each split, one object instance is selected randomly from each category for testing and all remaining object instances are for training, resulting in around 35,000 training examples and 6,877 test examples. Note that all training sets are labeled for supervised learning; (2) for our semi-supervised learning, the main setting is the same as the supervised learning, but the training set is randomly divided into two parts: labeled and unlabeled examples, e.g., if we label 20% of the training set, we get around 7,000 labeled and 28,000 unlabeled for the training set. Note that the learning process of co-training is based on the labeled and unlabeled examples of the training set. All the experiments are repeated 10 times on the test set and the average accuracies are given.

Methods	Labeled Size	Depth	RGB	Combine
Linear SVM [6]	35k	53.1 ± 1.7	74.3 ± 3.3	81.9 ± 2.8
Kernel SVM [6]	35k	64.7 ± 2.2	74.5 ± 3.1	83.8 ± 3.5
Random Forest [6]	35k	66.8 ± 2.5	74.7 ± 3.6	79.6 ± 4.0
IDL [19]	35k	70.2 ± 2.0	78.6 ± 3.1	85.4 ± 3.2
3D SPMK(L=2) [34]	35k	67.8	–	–
KDES [1]	35k	78.8 ± 2.7	77.7 ± 1.9	86.2 ± 2.1
CKM [2]	35k	–	–	86.4 ± 2.3
original HMP [40]	35k	70.3 ± 2.2	74.7 ± 2.5	82.1 ± 3.3
HMP [3]	35k	81.2 ± 2.3	82.4 ± 3.1	87.5 ± 2.9
CNN-RNN [4]	35k	78.9 ± 3.8	80.8 ± 4.2	86.8 ± 3.3
CNN-RNN+CT [32]	7k	77.7 ± 1.4	81.8 ± 1.9	87.2 ± 1.1
enhanced CNN-RNN+CT	7k	82.0 ± 2.1	84.1 ± 1.3	89.9 ± 1.3
CNN-SPM-RNN+CT	7k	**83.6** ± 2.3	**85.2** ± 1.2	**90.7** ± 1.1

Table 2: Comparison of recent results on the Washington RGB-D object database. CT means the proposed semi-supervised learning method with co-training.

3.1.1. Comparison to the state-of-the-art

We compare the results of our semi-supervised learning method to the recent methods, as shown in Table 2. For semi-supervised learning, we only label 20% of the training set, and exploit the enhanced CNN-RNN and CNN-SPM-RNN to extract RGB-D features, respectively. We utilize linear SVMs to train the two classifiers, and set $\alpha = 0.65, I = 500$ in Equation 1 when combine the two classifiers to predict the test set. The results demonstrate that our method can achieve the state-of-the-art performance against other methods. Furthermore, the CNN-SPM-RNN features are more discriminative than the enhanced CNN-RNN features, showing the effectiveness of maintaining the natural data sizes and aspect ratios by the SPM layer.

Among all these methods, the previous work [32] also employs semi-supervised learning method for RGB-D object recognition. However, The work [32] only extracts features from RGB images and depth images based on the CNN-RNN model [4], without considering grayscale images and surface normals. The experimental results also demonstrate the recognition performance can be improved with the additional information provided by grayscale and surface normals.

3.1.2. Unbiased Feature Evaluation in Supervised Setting

Since the two distinct feature sets F_{RGB} and F_{depth} are crucial for our semi-supervised learning, it's very necessary for us to fairly evaluate different features and choose the best one. Firstly, we give unbiased feature evaluation in supervised setting:

Methods	Depth	RGB	Combine
KDES [1]	78.8 ± 2.7	77.7 ± 1.9	86.2 ± 2.1
enhanced CKM	82.8 ± 2.4	81.8 ± 2.7	87.5 ± 2.4
HMP [3]	81.2 ± 2.3	82.4 ± 3.1	87.5 ± 2.9
enhanced CNN-RNN	82.4 ± 2.3	84.8 ± 1.4	89.9 ± 1.4
CNN-SPM-RNN	$\mathbf{83.4} \pm 2.4$	$\mathbf{85.4} \pm 1.3$	$\mathbf{90.7} \pm 1.4$

Table 3: Unbiased feature evaluation in supervised setting. Here all the training examples of size 35k are labeled, and linear SVM is used as our classifier.

All the five feature extraction methods are exploited to represent the RGB-D objects respectively as introduced in Section 2.3, and linear SVMs are used as our classifiers. We set $\alpha = 0.65$ when combine the two classifiers C_{RGB} and C_{depth} to recognize objects. The results are shown in Table 3.

It is worth to note that the ranking results in our evaluation are quite different from the published results. As shown in Table 2, the published results are ranked as follows: KDES < CKM < CNN-RNN < HMP. However, the capabilities of CKM and CNN-RNN are restricted a lot since grayscale images and surface normals are ignored in [2, 4]. The reasonable ranking results in our unbiased feature evaluation in Table 3 are: KDES < enhanced CKM, HMP < enhanced CNN-RNN. It can be analysed that kernel descriptors are a set of generalized histogram features like SIFT, although different kernel descriptors can be designed to capture different cues of objects, this kind of handcrafted features still have limited capability to describe an object; Both enhanced CKM and HMP are unsupervised feature learning models, which can learn more powerful features from the raw data than a set of manually designed kernel descriptors. The unsupervised learning structure can learn translational invariance of low-level features (the first layer in enhanced CKM and HMP) as well as some sort of high-level structures (the second layer of HMP) from the raw data. The enhanced CNN-RNN performs better, which employs a deep feature learning structure to discover discriminative and robust features. Our CNN-SPM-RNN achieves the state-of-the-art performance. Experimental results imply that CNN-SPM-RNN features are of the highest probability to guarantee the success of co-training in learning from the unlabeled RGB-D data.

3.1.3. Unbiased Feature Evaluation in Semi-supervised Setting

The same experiments are executed in the semi-supervised setting. Similarly, we extract the five features as the unified framework in Section 2.3, then exploit co-training

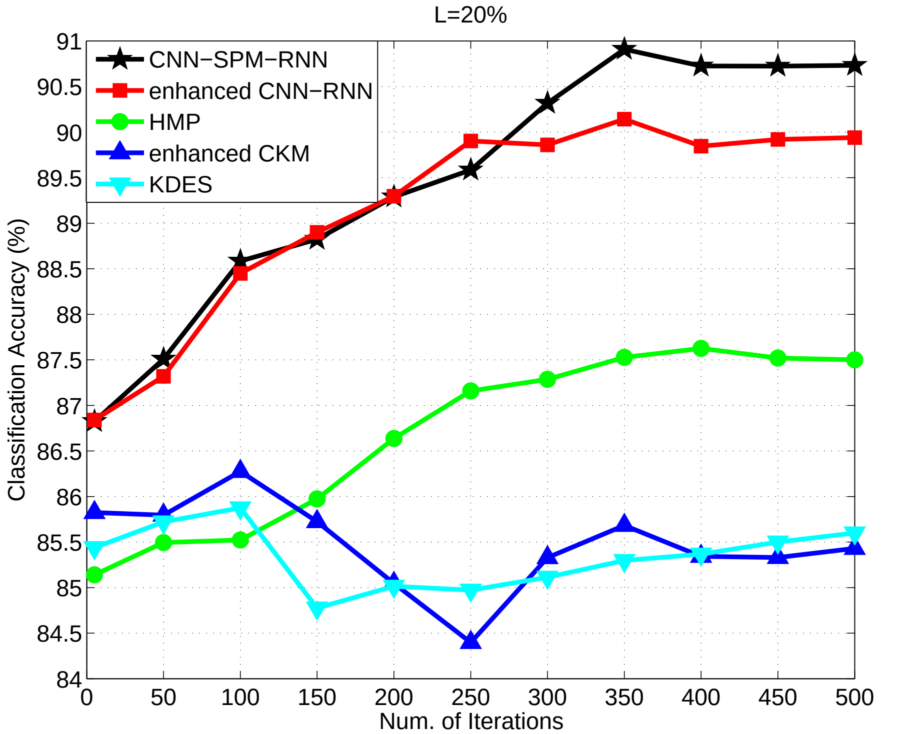


Figure 6: Unbiased feature evaluation in semi-supervised learning. We only label 20% (around 7k) of the training set, and use linear SVM as our classifiers. In the figure, we only report the combined results on the testing set.

to iteratively improve the two linear SVM classifiers through learning from the unlabeled data. In the learning process, we set $|n_j| = 2$ for each predicted class. We combine the two classifiers to predict the testing set and give the recognition accuracy every 50 iterations, as shown in Fig 6.

Among the five kinds of features, only CNN-SPM-RNN, enhanced CNN-RNN and HMP can make co-training succeed in learning from the unlabeled RGB-D data. The reason is that the two classifiers C_{RGB} and C_{depth} based on CNN-SPM-RNN, enhanced CNN-RNN or HMP features can be reliable to learn from the unlabeled data in each iteration, i.e., most examples transferred from the unlabeled pool U to the labeled pool L are correctly labeled. However, C_{RGB} and C_{depth} based on kernel or enhanced CKM features could add many examples from the unlabeled pool U with incorrect labels, which can in turn degrade the performance of the two classifiers. The rationale behind this result is that the features extracted by shallow learning models (enhanced CKM: one layer, KDES: manually designed) are not as robust as the deeper learning models (HMP: two layers, enhanced CNN-RNN and CNN-SPM-RNN: multiple lay-

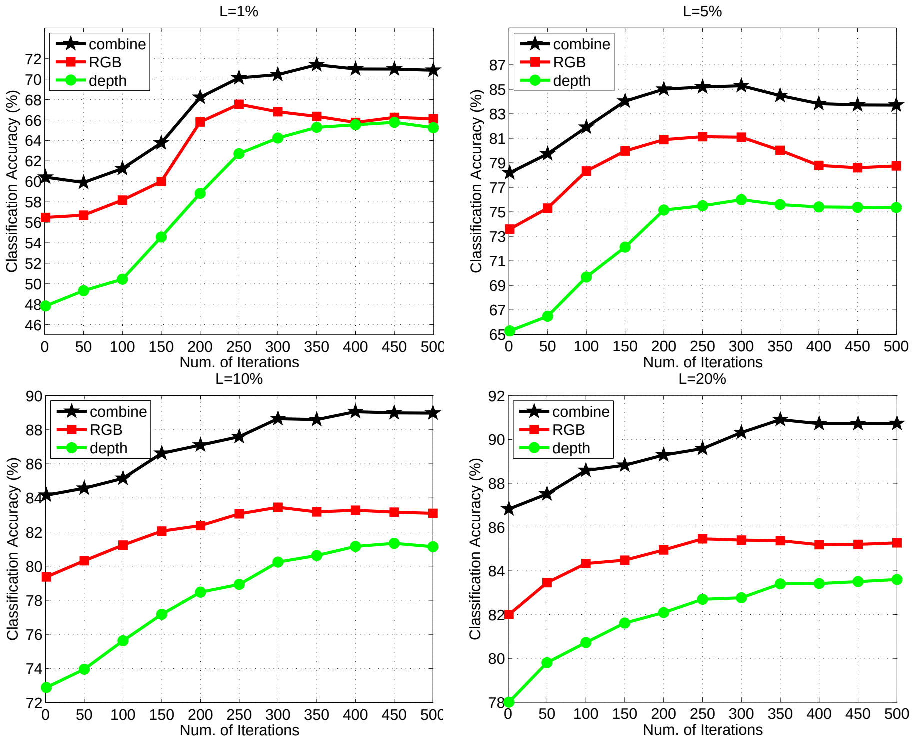


Figure 7: Recognition accuracy with iteration number I and the labeled size L ($|n_j| = 2, \alpha = 0.65$).

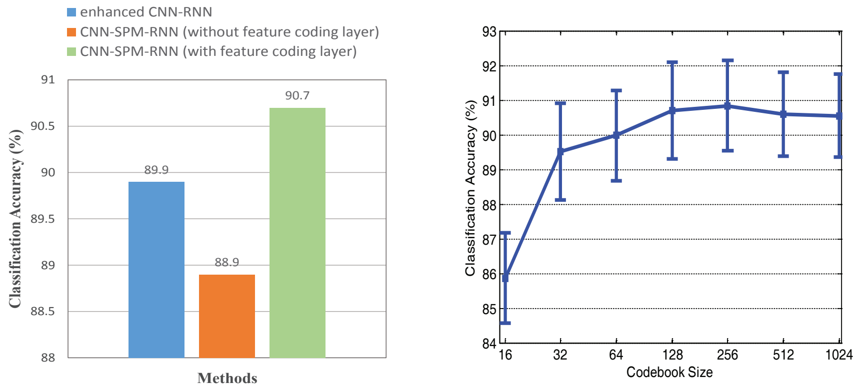
ers.) The biggest difference between the semi-supervised learning and the supervised learning, is that enhanced CKM performs much worse than HMP. It is probable that the features learned by the two-layer learning structure of HMP can obtain more robustness and generalization ability than one-layer enhanced CKM features. As well, CNN-SPM-RNN performs the best in our semi-supervised learning.

3.1.4. Parameter Analysis

In this section, first, we analyse the effectiveness of the CNN-SPM-RNN feature learning model. Second, using the CNN-SPM-RNN model with default setting to extract features, we analyse the effectiveness of the co-training model.

CNN-SPM-RNN. CNN-SPM-RNN model is proposed to address the problem of cropping or warping in CNN-RNN model, and learn more powerful features from the raw data. Besides spatial pyramid pooling, the success of CNN-SPM-RNN is highly dependent on the feature coding layer as well as its codebook size.

(1) CNN-SPM-RNN with and without feature coding layer. Fig. 8 (a) shows that a feature coding layer can significantly improve the performance of the learned fea-



(a) (b)

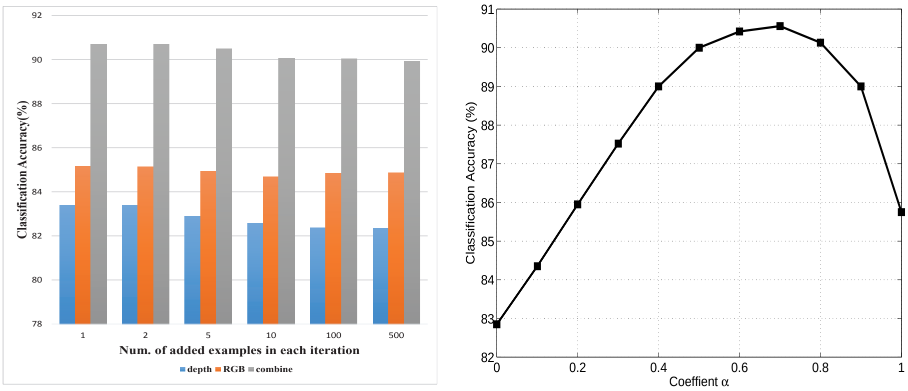
Figure 8: (a) Performance of CNN-SPM-RNN with feature coding layer ($k_2 = 128$) and without feature coding layer; (b) Performance of CNN-SPM-RNN with the codebook size of the feature coding layer. For both (a) and (b), the classification accuracy is based on the supervised setting.

tures. Through feature coding, we can learn high-level feature responses based on the low-level convolutional responses, which can further improve the effectiveness of the learned features. This is also one of the main differences between our work (for unsupervised deep learning model) and He et al's work [29] (for supervised deep learning model), when applying spatial pyramid pooling to adopt the feature learning model for arbitrary image sizes.

(2) Influence of the codebook size of the feature coding layer. Fig. 8 (b) demonstrates that the performance of the CNN-SPM-RNN feature can rapidly grow with larger codebook size at the beginning. When the codebook size $k_2 > 128$, the recognition accuracy keeps stable with a very high value. The main reason is that more codewords can describe more patterns of features for object recognition, while some mild overfitting can exist for very large codebook size. Considering the efficiency and accuracy, $k_2 = 128$ is used for CNN-SPM-RNN.

Co-training. Using CNN-SPM-RNN model to extract features for each modality of each object, the co-training method is closely related to the iteration number I, the labeled training size L, the number of added examples $|n_j|$, and the coefficient α. We analyze each by fixing other parameters in the following.

(1) Influence of the number of iterations. As shown in Fig. 7, all the accuracy curves of co-training from $L = 1\%$ to $L = 20\%$ suggest the similar trends as the number of iterations increases. After several hundreds of iterations (around 400), the



(a) (b)

Figure 9: (a) Recognition accuracy with the added confident examples $|n_j|$ for each class in each iteration ($I = 500,\ L = 20\%, \alpha = 0.65$); (b) Recognition accuracy with the coefficient α to combine the RGB view and depth view classifier ($I = 500, L = 20\%, |n_j| = 2$).

recognition accuracy can rise and converge to a relatively high value, since most of the unlabeled examples have been transferred from the unlabeled pool to the labeled pool. This characteristic of co-training is very useful and practical as we can determine the final iteration number from a wide range and require for high precision at the same time.

(2) Influence of the labeled size of the training set. Fig. 7 also shows the labeled size L can greatly determine the growth rate of each co-training curve and the final recognition accuracy. When L is given a smaller size, it implies a much bigger growth potential value along with the learning process of co-training, this is mainly because the two initialized much weaker classifiers C_{RGB} and C_{depth} can be improved a lot by using additional examples from the unlabeled pool. We see that a bigger size L can keep a much higher final recognition accuracy ($L = 1\%$: 70.9%; $L = 5\%$: 83.7%; $L = 10\%$: 89.0%; $L = 20\%$: 90.7%;). It is very reasonable since the two classifiers C_{RGB} and C_{depth} are more reliable to add examples with correct labels from the beginning to the end, when given more labeled training examples.

(3) Influence of the number of added confident examples in each iteration. To keep the balance of the size for each category in the training pool, we try to add the same number of confident examples for each category in each iteration (i.e., $|n_j|$)). Fig. 9a reveals that the recognition results are very robust to $|n_j|$ when it rises from 1 to 500. Notice that the actually added examples are simultaneously constrained by the score

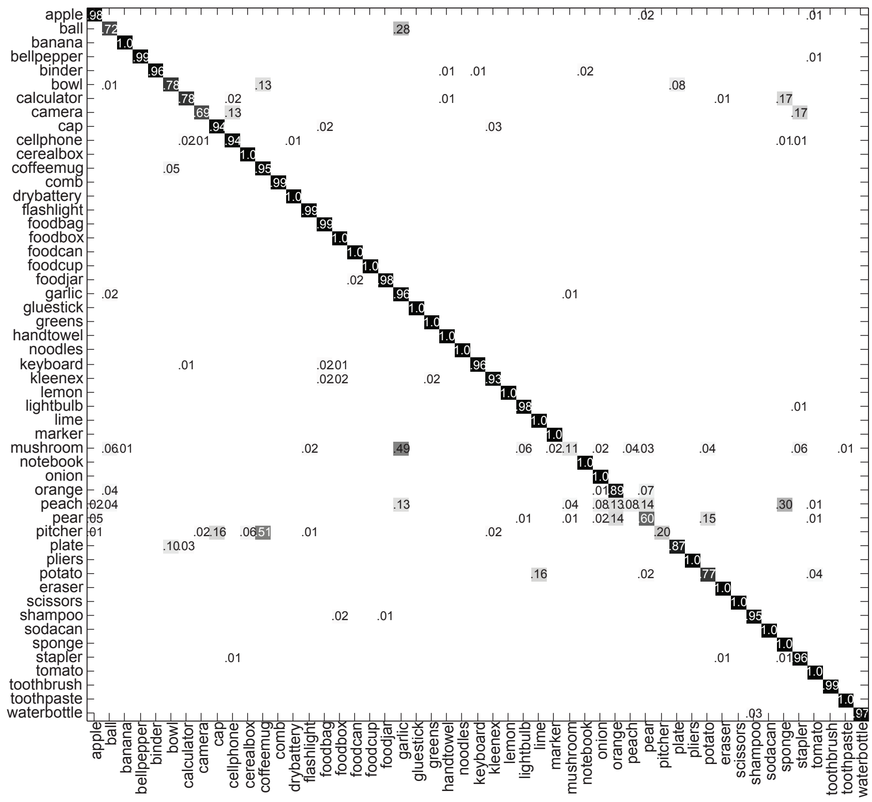


Figure 10: Confusion matrix of our semi-supervised learning method on the Washington RGB-D dataset($L = 20\%$, $I = 500$, CNN-SPM-RNN features). The y-axis indicates the ground true labels, and the x-axis indicates the predicted labels. Some misclassifications are: mushroom as garlic, pitcher as coffee mug

thresholds of θ_{RGB} and θ_{depth}, since we do not trust those examples with too low scores. For the Washington RGB-D dataset, we fix $\theta_{RGB} = \theta_{depth} = 0.1$.

(4) Influence of the coefficient to combine the two view classifiers. As shown in Fig. 9b, when change the coefficient α from 0 (only using the depth classifier) to 1 (only using the RGB classifier), the result can gradually rise to the top (when $\alpha \in [0.5, 0.8]$) and then degrade. It is reasonable that object recognition can benefit a lot by regarding both RGB view and depth view effectively.

We conclude that it is convenient to determine the parameter for our co-training algorithm, since a wide range of parameters ($I \geq 400,\ L \geq 10\%, 0.5 \leq \alpha \leq 0.8, |n_j| \geq 1$) can keep co-training successful. This characteristic is very important and useful in practical usage. For the Washington RGB-D dataset, we select $I = 500,\ L = 20\%, \alpha = 0.65, |n_j| = 2$ for a balance of performance and efficiency.

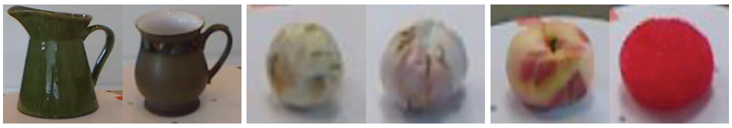


Figure 11: Examples of confused categories on the Washington RGB-D dataset. Pitcher classified as coffee mug, mushroom as garlic, and peach as sponge due to similar color or shape.

3.1.5. Error analysis

The confusion matrix of our semi-supervised learning method over the 51 categories is shown in Fig. 10. Most categories can be correctly classified, meaning that our method can achieve high precision of object recognition with only a small set of labeled data.

We show some examples of the often misclassified categories in Fig. 11. Mushrooms have almost the same appearance with garlic, pitchers look similar to coffee mugs from some angle, and peaches have similar shape with sponges.

3.2. 2D3D Object Database

We also verify the effectiveness of the semi-supervised learning method on the second public RGB-D object database, called 2D3D [5]. It consists of 156 object instances organized into 14 categories (see examples in Fig. 12). Each instance is recorded every $10°$ around the vertical axis on a turntable, yielding 36 views per instance.

We also focus on category recognition. For supervised learning, we follow the setting in [5]. We first sample 18 views for each instance and reduce the size of the total database to 2,808 RGB-D images. Then we randomly split the database into training and test sets, resulting in 82 instances with a total of 1476 views in the training set, while 74 objects with 1332 views in the test set. For the semi-supervised learning, we keep the main setting the same with the supervised. The only difference is that, we randomly divide the training set into two parts: 20% labeled and 80% unlabeled. Note that we also utilize additional instance labels in the process of co-training, in order to balance the examples of each instance in the training pool U. It is very important for co-training to bootstrap from such a small size of labeled examples.

Table 4 shows the results for both the supervised setting and the semi-supervised setting. When all the training examples are labeled, CNN-SPM-RNN can achieve the best result with 92.5% accuracy. It is worth to note that KDES ranks the second and is superior to enhanced CKM, HMP and enhanced CNN-RNN. The main reason is prob-

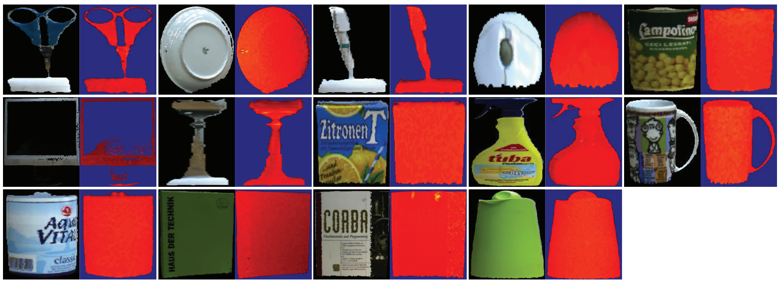


Figure 12: Object examples from the 2D3D object database. Each object (RGB + depth) shown here belongs to a different category.

Methods	Labeled Size	Depth	RGB	Combine
Browatzki et al. [5]	1476	74.6	66.6	82.8
KDES [1]	1476	88.7	**89.1**	92.8
enhanced CKM	1476	87.1	82.8	88.7
HMP [3]	1476	87.6	86.3	91.0
enhanced CNN-RNN	1476	88.7	88.2	92.5
enhanced CNN-RNN+CT	328	86.0	85.3	88.2
CNN-SPM-RNN	1476	**89.4**	88.5	**92.9**
CNN-SPM-RNN+CT	328	86.0	85.4	88.4

Table 4: Comparison of results on the 2D3D object database. CT means the proposed semi-supervised learning method with co-training.

able that KDES takes advantage of many manually designed attributes such as object size, shape, edges, etc, which can help a lot to depict the objects than many learning based features on the relatively small scale 2D3D database. When only given 20% labeled training set, the performance of the enhanced CNN-RNN and CNN-SPM-RNN with co-training are 88.2% and 88.4%, respectively. It is reasonable that the performance of our semi-supervised learning is lower than the supervised learning. Because co-training starts with such a few labeled examples for a multi-class recognition problem, one or two added examples from the unlabeled pool U with incorrect labels can largely affect the performance of the two linear SVMs in the next round iteration.

3.3. Fusing RGB-D Object Dataset

The fusing RGB-D object database [41] consists of 8 classes of everyday objects (cup, bottle, doll, teddy bear, remote control, shoe, stapler, and pot, shown in Fig. 13), each with 10 objects per class. For each object, 12 images are captured by recording 2 camera positions, 3 object poses and 2 illumination conditions. Thus there are 960 image pairs in the database. We randomly split the database into training and test sets

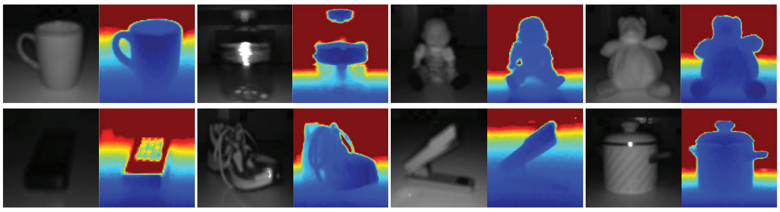


Figure 13: Object examples from the fusing RGB-D object database. Each object (RGB + depth) shown here belongs to a different category.

Methods	Labeled Size	Depth	RGB	Combine
GIFT [41]	480	80.4	77.1	84.1
3D SPMK(L=2) [42]	480	72.0	–	–
KDES [1]	480	81.0	**82.8**	**88.3**
enhanced CKM	480	83.0	77.7	87.4
HMP [3]	480	**84.3**	78.1	87.1
enhanced CNN-RNN	480	74.9	75.5	81.4
CNN-SPM-RNN	480	79.4	80.4	85.0

Table 5: Comparison of results on the fusing RGB-D dataset. Since this database has a very small scale of images, we only show the supervised results.

with five different objects per class in each according to [41]. Since there are a very small scale of images, we only evaluate the performance of the supervised setting.

The results in Table 5 show that the well-designed kernel features of KDES can achieve the state-of-the-art performance on the very small scale database. The main reason is that those unsupervised feature learning methods like enhanced CKM, HMP, CNN-RNN and CNN-SPM-RNN require a lot of training examples to achieve the discriminating abilities. And for depth modality, such effects by the small scale of training set are even more remarkable for enhanced CNN-RNN and CNN-SPM-RNN.

4. Conclusion

This paper proposes a semi-supervised learning framework based on co-training for RGB-D object recognition, which can exploit the two distinct views (RGB and depth) to boost the performance by learning from the unlabeled data. Through the analysis of the state-of-the-art features along with an unbiased evaluation, we find out that the proposed CNN-SPM-RNN features are very powerful to represent the RGB-D objects. The experiments demonstrate the effectiveness of our method, especially for large scale RGB-D datasets, since both the CNN-SPM-RNN features and the semi-supervised learning model can benefit from more available data. Furthermore, our method is lowly sensitive to the selection of the parameter values such as the labeled training size, the iteration number, etc. We believe our method can be a useful tool for many vision applications, e.g, RGB-D dataset annotation and robot navigation, by solving the expensive and difficult task of manually labeling massive data.

In addition, this paper also provides a meaningful guideline how to better represent RGB-D objects. For large scale RGB-D object datasets, e.g., the Washington RGB-D datset, those unsupervised feature learning methods such as enhanced CKM, HMP, enhanced CNN-RNN and CNN-SPM-RNN can achieve higher recognition performance. While for small scale RGB-D object datasets, e.g., the fusing RGB-D dataset, the well-designed kernel descriptors on top of various attributes such as object shape, size, edges, etc. are the best choice to depict objects.

In future work, we will try to use on-line learning algorithms to improve the efficiency of the semi-supervised learning framework.

5. Acknowledgements

This work is funded by the National Basic Research Program of China (Grant No. 2012CB316302), National Natural Science Foundation of China (Grant No. 61322209 and Grant No. 61175007), the National Key Technology R&D Program (Grant No. 2012BAH07B01). The authors thank all the anonymous reviewers for their very helpful comments to improve the paper.

References

[1] L. Bo, X. Ren, D. Fox, Depth kernel descriptors for object recognition, in: IROS, 2011.

[2] M. Blum, J. T. Springenberg, J. Wulfing, M. Riedmiller, A learned feature descriptor for object recognition in rgb-d data, in: ICRA, 2012.

[3] L. Bo, X. Ren, D. Fox, Unsupervised feature learning for rgb-d based object recognition, International Symposium on Experimental Robotics (ISER), June.

[4] R. Socher, B. Huval, B. Bath, C. D. Manning, A. Ng, Convolutional-recursive deep learning for 3d object classification, in: NIPS, 2012.

[5] B. Browatzki, J. Fischer, B. Graf, H. Bulthoff, C. Wallraven, Going into depth: Evaluating 2d and 3d cues for object classification on a new, large-scale object dataset, in: ICCV Workshops, 2011.

[6] K. Lai, L. Bo, X. Ren, D. Fox, A large-scale hierarchical multi-view rgb-d object dataset, in: Robotics and Automation (ICRA), 2011.

[7] A. Janoch, S. Karayev, Y. Jia, J. T. Barron, M. Fritz, K. Saenko, T. Darrell, A category-level 3d object dataset: Putting the kinect to work, in: Consumer Depth Cameras for Computer Vision, Springer, 2013, pp. 141–165.

[8] M. Sun, S. S. Kumar, G. R. Bradski, S. Savarese, Object detection, shape recovery, and 3d modelling by depth-encoded hough voting, Computer Vision and Image Understanding 117 (9) (2013) 1190–1202.

[9] M. Zia, M. Stark, K. Schindler, Are cars just 3d boxes? - jointly estimating the 3d shape of multiple objects, in: CVPR, 2014.

[10] D. Held, J. Levinson, S. Thrun, S. Savarese, Combining 3d shape, color, and motion for robust anytime tracking, in: Robotics: Science and Systems (RSS), 2014.

[11] M. Luber, L. Spinello, K. O. Arras, People tracking in rgb-d data with on-line boosted target models, in: Intelligent Robots and Systems (IROS), 2011.

[12] W. Choi, C. Pantofaru, S. Savarese, Detecting and tracking people using an rgb-d camera via multiple detector fusion, in: ICCV Workshops, 2011.

[13] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, A. Fitzgibbon, Scene coordinate regression forests for camera relocalization in rgb-d images, in: CVPR, 2013.

[14] J. Sturm, N. Engelhard, F. Endres, W. Burgard, D. Cremers, A benchmark for the evaluation of rgb-d slam systems, in: IROS, 2012.

[15] B. Ni, G. Wang, P. Moulin, Rgbd-hudaact: A color-depth video database for human daily activity recognition, in: Consumer Depth Cameras for Computer Vision, Springer, 2013, pp. 193–208.

[16] J. Sung, C. Ponce, B. Selman, A. Saxena, Unstructured human activity detection from rgbd images, in: ICRA, 2012.

[17] J. Xiao, A. Owens, A. Torralba, Sun3d: A database of big spaces reconstructed using sfm and object labels, in: ICCV, 2013.

[18] N. Silberman, R. Fergus, Indoor scene segmentation using a structured light sensor, in: ICCV workshop, 2011.

[19] K. Lai, L. Bo, X. Ren, D. Fox, Sparse distance learning for object recognition combining rgb and depth information, in: ICRA, 2011.

[20] D. Yarowsky, Unsupervised word sense disambiguation rivaling supervised methods, in: Proceedings of the 33rd annual meeting on Association for Computational Linguistics, 1995.

[21] E. Riloff, J. Wiebe, T. Wilson, Learning subjective nouns using extraction pattern bootstrapping, in: Proceedings of the seventh conference on Natural language learning, 2003.

[22] C. Rosenberg, M. Hebert, H. Schneiderman, Semi-supervised self-training of object detection models, in: Seventh IEEE Workshop on Applications of Computer Vision, 2005.

[23] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: COLT, 1998.

[24] M.-F. Balcan, A. Blum, K. Yang, Co-training and expansion: Towards bridging theory and practice, in: NIPS, 2004.

[25] A. Blum, S. Chawla, Learning from labeled and unlabeled data using graph mincuts, in: ICML, 2001.

[26] A. Blum, J. Lafferty, M. R. Rwebangira, R. Reddy, Semi-supervised learning using randomized mincuts, in: ICML, 2004.

[27] B. Pang, L. Lee, A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts, in: Proc. of the 42nd annual meeting on Association for Computational Linguistics, 2004.

[28] R. Socher, C. C. Lin, C. Manning, A. Y. Ng, Parsing natural scenes and natural language with recursive neural networks, in: ICML, 2011.

[29] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, in: ECCV, 2014.

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

[31] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: NIPS, 2012.

[32] Y. Cheng, X. Zhao, K. Huang, T. Tan, Semi-supervised learning for rgb-d object recognition, in: ICPR, 2014.

[33] A. Coates, A. Y. Ng, H. Lee, An analysis of single-layer networks in unsupervised feature learning, in: International Conference on Artificial Intelligence and Statistics, 2011.

[34] C. Redondo-Cabrera, R. J. Lopez-Sastre, J. Acevedo-Rodriguez, S. Maldonado-Bascon, Surfing the point clouds: selective 3d spatial pyramids for category-level object recognition, in: CVPR, 2012.

[35] D. G. Lowe, Distinctive image features from scale-invariant keypoints, IJCV 60 (2) (2004) 91–110.

[36] A. E. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3d scenes, PAMI 21 (5) (1999) 433–449.

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

[38] L. Bo, X. Ren, D. Fox, Kernel descriptors for visual recognition, in: NIPS, 2010.

[39] L. Bo, C. Sminchisescu, Efficient match kernel between sets of features for visual recognition., in: NIPS, 2009.

[40] L. Bo, X. Ren, D. Fox, Hierarchical matching pursuit for image classification: architecture and fast algorithms, in: NIPS, 2011.

[41] A. Bar-Hillel, D. Hanukaev, D. Levi, Fusing visual and range imaging for object class recognition, in: ICCV, 2011.

[42] C. Redondo-Cabrera, R. J. Lopez-Sastre, J. Acevedo-Rodriguez, S. Maldonado-Bascon, Recognizing in the depth: selective 3d spatial pyramid matching kernel for object and scene categorization.