Semi-Supervised Learning for RGB-D Object Recognition

Yanhua Cheng, Xin Zhao, Kaiqi Huang, and Tieniu Tan
Center for Research on Intelligent Perception and Computing
National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
Email:{yanhua.cheng, xzhao, kqhuang, tnt}@nlpr.ia.ac.cn

***Abstract*—Conventional supervised object recognition methods have been investigated for many years. Despite their successes, we are still suffering from two limitations: (1) various information of an object is represented by artificial features only derived from RGB images; (2) a lot of manually labeled data is required by supervised learning. To address those limitations, we propose a new semi-supervised learning framework based on RGB and depth (RGB-D) images to improve object recognition. In particular, our framework has two modules: (1) RGB and depth images are represented by convolutional-recursive neural networks to construct high level features, respectively; (2) co-training is exploited to make full use of unlabeled RGB-D instances due to the existing two independent views. Experiments on the standard RGB-D object dataset demonstrate that our method can compete against with other state-of-the-art methods with only 20% labeled data.**

I. Introduction

Object recognition plays a very important role in computer vision community and has plentiful appealing applications. Despite its great potentials, many latent factors (e.g., large variations of object appearance and different object viewpoints, as shown in Fig. 1) can significantly influence the final recognition results since the appearance features [1] based on RGB images are not distinctive enough. In order to improve the recognition performance, many researchers have paid attentions on object representation by capturing other useful information of objects (e.g., shape and spatial geometry information). One of the representative shape features is the Histogram of Oriented Gradient (HOG) [2]. Other work supposes to consider the spatial region information by using the segmentation based methods [3].

Regardless of the impressive characteristics and performance of the above methods, there is an inevitable limitation that they construct the features all generated from RGB images by utilizing their 2D information. As the RGB images themselves have various variations, the obtained shape and spatial geometry features may not be reliable as well. Recently, due to the development of sensing technology, depth cameras (e.g., Kinect) utilize infrared (IR) projector and IR camera to obtain the depth images of the scenes while capture the RGB images via RGB camera at the same time. Such sensor data is called RGB-D as shown in Fig. 1. Since the imaging mechanisms are quite different, the RGB and depth images are generated independently without the influence to each other. Depth images can provide adequate shape and spatial geometry information of the scenes including objects [4]–[6].

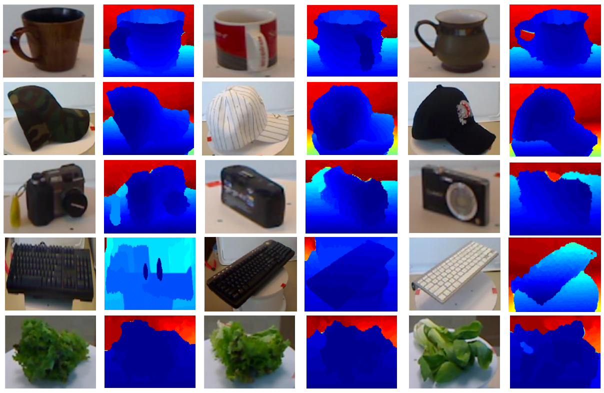

Fig. 1. RGB-D images from Kinect. Each row belongs to a category. RGB images can capture color and texture information, while depth images provide pure geometry and shape cues for us.

Recently, much work has been developed to combine RGB and depth images to improve object recognition [4], [5], [7]–[9]. All these methods focus on learning a good feature representation to make full use of the depth modality. Hand-crafted features such as SIFT [4] or special size, shape and geometry features [7] are applied to depth channel. Instead of hand-crafted features, unsupervised feature learning is one alternative to learn discriminative features from raw vision data, e.g., convolutional k-means descriptors (CKM) [8] and hierarchical matching pursuit (HMP) [9], which has achieved large improvements in RGB-D object recognition.

In this paper, we apply unsupervised convolutional-recursive neural networks (CNN-RNN) to learn a set of basis features for RGB and depth images respectively. Compared to CKM [8] and HMP [9], CNN-RNN is more fast and does not need additional features such as surface normals. The most relevant work to ours is [5]. However, they simply take the depth image as the fourth channel of the RGB image for each instance and learn the combined features. Instead, we will explore the complementarity between RGB and depth information, thus learn two feature sets separately.

For classifier construction, all these work [4], [5], [7]–[9] employ a supervised learning way. For a supervised pattern recognition task, it is important to have an adequate number of labeled data to train the model. However, gathering sufficient labeled data is difficult especially when we suffer from a large scale recognition problem [4]. An effective method to deal with this problem is utilizing relatively plentiful and cheap unlabeled data. This is the basic idea of the semi-supervised learning methods [10]. To be a semi-supervised paradigm, co-training [11] is one of the most representative semi-supervised learning methods. The co-training algorithm needs two distinct views of the data. It assumes that each view is described by an independent feature set. Furthermore, the two feature sets can provide useful and complementary information about one instance. When these conditions meet, co-training can be very successful to learn from unlabeled data [11].

Before the emergence of low-cost depth sensors, objects only consist of one view, i.e., RGB view. To meet the conditions of co-training, some researches manufacture a feature split from RGB images, such as shape or edge information as the second view. While this kind of feature split can obtain some success demonstrated in [12] and [13], the learning ability of co-training is very limited. Co-training may degrade the performance as well when the feature split is not so well designed, as showed in [13]. Fortunately, with new depth cameras that can record high quality RGB and depth images, we can extract a natural independent split of features. Thus, in this paper we propose to use this natural feature split via co-training to increase the capability of recognition.

The contributions of this paper are as follows:

- We propose a semi-supervised learning framework for RGB-D object recognition. To the best of our knowledge, this is the first semi-supervised learning solution to combine RGB and depth information for object recognition.
- We use unsupervised convolutional-recursive neural networks to learn high level feature representations for both RGB and depth images. We learn the features from the raw RGB-D data efficiently.
- On the standard RGB-D Object Database, experimental results show that our approach obtains a promising performance with only 20% labeled data, compared to other state-of-the-art methods.

The rest of this paper is organized as follows: Section II discusses the related work. Section III proposes our semi-supervised learning framework, where unsupervised feature learning and the co-training algorithm are described in detail. We give experiments and present the comparison of our semi-supervised learning method with the state-of-the-art methods in Section IV. Finally, in Section V, we draw a conclusion and discuss future work.

II. Related Work

With the advent of new depth sensing technology, much work has been proposed to combine color and depth images to improve object recognition. We briefly review the existing feature representation methods and semi-supervised learning approaches that are related to our method.

Feature Representation: For standard object recognition, well-designed features based on orientation histograms such as SIFT [14] or HOG [2] are proved to be successful. However, these hand-crafted features just can capture a small set of recognition cues, for example, SIFT is sensitive to corners and edges but ignores color information. To adapt to new data modality like RGB-D images, Lai *et al.* [4] simply try to extend SIFT to depth channel and Bo *et al.* [7] use kernel descriptors to describe size, 3D shape, and depth edges.

Instead of those hand-crafted features, unsupervised feature learning is one alternative to learn powerful image representations. Since Hinton *et al.* [15] introduced deep belief networks, a wide spectrum of unsupervised deep learning methods have been employed: Denoising Autoencoders [16], Deep Boltzmann Machines [17], Hierarchical Sparse Coding [18] and K-Means based feature learning [19]. While these methods have achieved great improvements in object recognition, they mainly focused on RGB images, especially gray images. Recently, Blum *et al.* [8] introduced convolutional k-means descriptors (CKM) for feature learning from RGB-D images. Convolutional k-means descriptors are extracted from a set of detected SURF interest points. Meanwhile, Bo *et al.* [9] proposed hierarchical matching pursuit method (HMP), which uses sparse coding to learn hierarchical feature representations in an unsupervised way. The most relevant work to ours is [6]. They also adopt the unsupervised feature learning method based on convolutional neural nets (CNN) [20] and recursive neural nets (RNN) [21]. However, the depth image is simply taken as the fourth channel of the RGB image for each instance in [5]. Furthermore, the RGB and depth features are directly concatenated to compose a high-dimensional instance representation.

Semi-Supervised Learning: It addresses the problem of learning a better classifier by combining a small set of labeled data and large amount of unlabeled data. Generally, collecting labeled examples is difficult and time consuming, while unlabeled examples are relatively easy and cheap to collect. Many semi-supervised learning methods [10] have been introduced to use the unlabeled data, including self-training, co-training, Expectation-Maximization (EM) and graph based methods.

In this paper, we are particularly interested in the co-training method first proposed in [11]. Co-training requires two distinct views of the data, and it has been proved very successful to learn from unlabeled data under two assumptions. One is the conditional independence assumption, which means that the two feature sets F_1 and F_2 extracted from the two different views of the same instance should be conditionally independent given the class label. The other is the sufficient assumption of the two views. That is to say, both views of the data can achieve good classification accuracy when enough labeled training samples are provided. We utilize co-training to learn from large amount of unlabeled RGB-D data to improve the capability of object recognition, since new depth cameras can provide two natural independent views (RGB and depth images) of each instance.

III. Our Approach

In this section we describe our semi-supervised framework for RGB-D object recognition. We simply discuss RGB-D data with Kinect at first. Then, we describe the unsupervised feature learning method via convolutional-recursive neural networks.

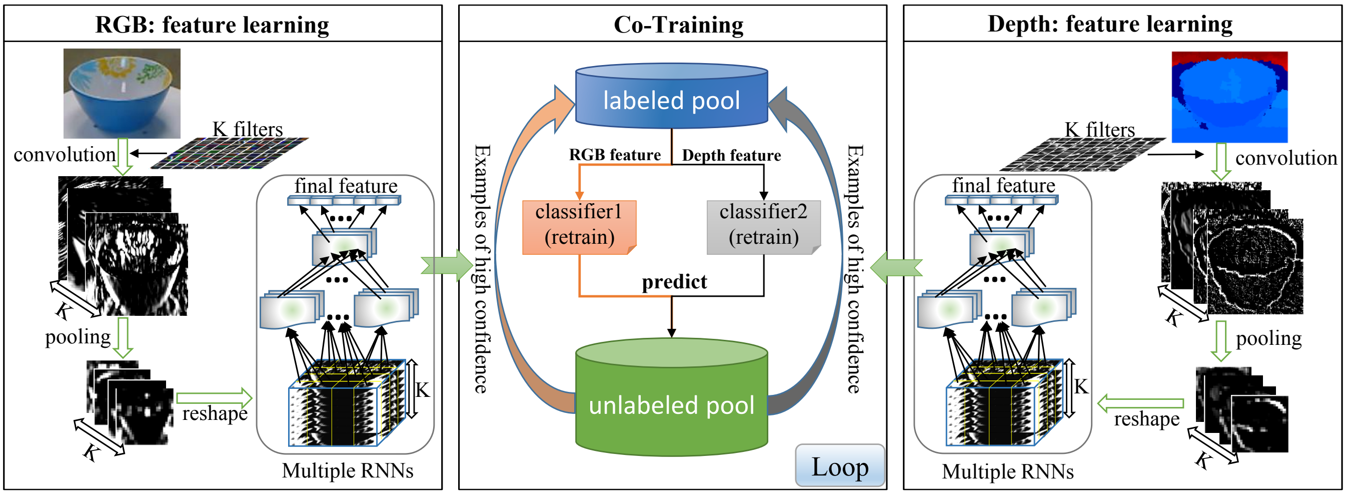

Fig. 2. An overview of our framework. We first use unsupervised CNN-RNN model to learn feature representations of RGB and depth images separately. Then, we employ co-training to combine the labeled and unlabeled RGB-D data iteratively to improve the performance of both the RGB classifier and the depth classifier. Finally, the output of the two classifiers can be combined to predict the test instances.

Finally, co-training is proposed to learn from the unlabeled data in order to promote both of the SVM classifiers (RGB classifier and depth classifier). Fig. 2 outlines our approach.

A. RGB-D data with Kinect

Kinect has become an important and popular 3D sensor. It consists of an infrared (IR) projector, an IR camera and a RGB camera. The IR camera and the IR projector form a stereo pair and can be used to triangulate points in 3D space, while the RGB camera can capture texture and color of the physical space. The imaging mechanism guarantees the independence of RGB and depth. With Kinect, we can obtain high quality RGB and depth images simultaneously, as shown in Fig. 1. The natural two independent and complementary views (RGB and depth) of the same instance are crucial for our method.

B. Unsupervised Feature Learning

We use unsupervised convolutional-recursive neural networks to learn high order feature representations for both RGB and depth images. There are three main steps in the feature extraction module. First, Both RGB and depth filters are simply learned by k-means clustering, as described in [22]. Then, we use a single convolutional layer to convolve the learned filters over the input image in order to extract low level features. Finally, the pooled convolutional responses of each image are input into multiple recursive neural networks with fixed tree structures to compose high level features. We apply the feature learning procedure to both RGB and depth images separately.

1) Filters by K-Means Clustering: We randomly sample sub-patches from training sets. Each sub-patch has dimension of $w \times w$ and has d channels (RGB: d=3; depth: d=1). The selected sub-patches are contrast normalized and whited before k-means clustering to obtain the filters. The procedure is done for RGB and depth images separately. In this paper, we set the number of the filters $K = 128$ for both RGB and depth images. As shown in Fig. 3, the resulting RGB and depth filters obtain high response values at image edges. This is chiefly because that object boundaries with large discontinuities of color or intensity can be remained while the other regions are likely smoothed by k-means clustering.

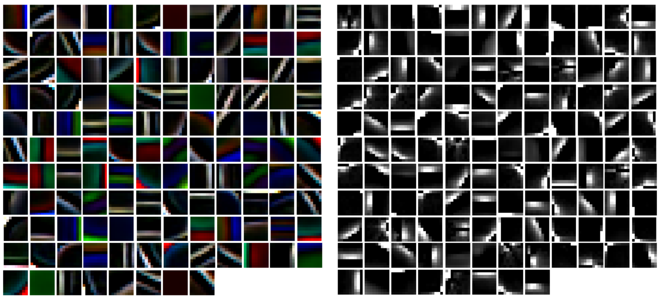

Fig. 3. K-means clustering based filters for RGB(**left**) and depth(**right**) images (best viewed in color). RGB filters can capture standard edge and color features, while depth filters can obtain much sharper edges.

2) A Single CNN Layer: Before we convolve the learned filters over the input image, we resize the image to $p \times p$. The convolutional responses of the input image have dimension of $(p - w + 1) \times (p - w + 1)$, followed by rectification with absolute values and local contrast normalization. We then use standard practice to reduce the dimensionality of the convolutional responses by average pooling with square regions of size q and step size s. This means that, the size of our final convolutional feature applied to each image is $K \times r \times r$ $(r = (p - w + 1 - q)/s + 1)$. We also apply this same procedure to RGB and depth images separately.

3) Multiple RNNs with Fixed-Tree Structures: Recursive neural network [21] was proposed to predict hierarchical tree structures for scene images and learn the hierarchical feature representations. The tree structure in [21] depends on input images. Later, Richard *et al.* [5] find that balanced fixed-tree RNN can also obtain approximately good performance for object recognition, while it can be designed very fast, despite lack of much flexibility. In this paper, we use multiple balanced fixed-tree RNNs to compose the convolutional features to high level representations.

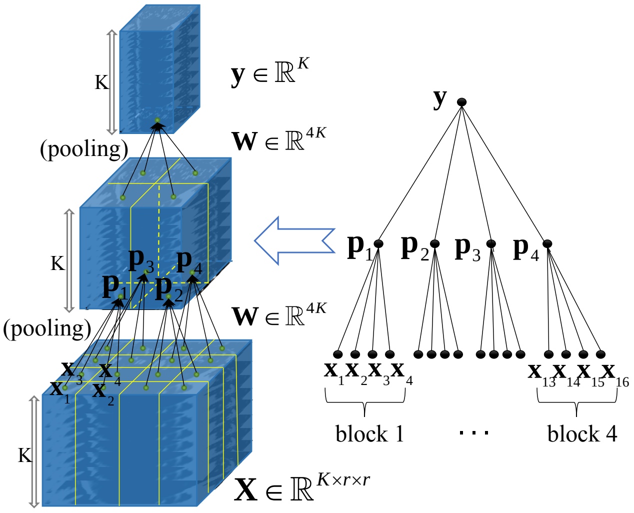
K
$\mathbf{y} \in \mathbb{R}^K$
(pooling)
$\mathbf{W} \in \mathbb{R}^{4K}$
K
$\mathbf{p}_3$ $\mathbf{p}_4$
$\mathbf{p}_1$ $\mathbf{p}_2$
(pooling)
$\mathbf{W} \in \mathbb{R}^{4K}$
$\mathbf{x}_3$ $\mathbf{x}_4$
$\mathbf{x}_1$ $\mathbf{x}_2$
K
$\mathbf{X} \in \mathbb{R}^{K\times r\times r}$
$\mathbf{y}$
$\mathbf{p}_1$ $\mathbf{p}_2$ $\mathbf{p}_3$ $\mathbf{p}_4$
$\mathbf{x}_1\mathbf{x}_2\mathbf{x}_3\mathbf{x}_4$
$\mathbf{x}_{13}\mathbf{x}_{14}\mathbf{x}_{15}\mathbf{x}_{16}$
block 1
· · ·
block 4

Fig. 4. An Example of a single fixed-tree RNN. Right is the fixed-tree RNN structure, where a list of adjacent column vectors in a block $b = 2 \times 2$ are merged into a parent vector with the same weight matrix at each layer. Left is the sketch map that demonstrates how the fixed-tree RNN organises the input convolutional responses to a high order representation.

The output $\mathbf{X} \in \mathbb{R}^{K\times r\times r}$ of the convolutional layer for each image is given to each RNN tree. Each RNN yields a function f that transforms an input 3D matrix $\mathbf{X}$ to a new representation $\mathbf{y} \in \mathbb{R}^K$, as shown in Fig. 4. A list of adjacent column vectors in a block of size b are merged into a parent vector $\mathbf{p}$ with the same weight $\mathbf{W}$ at each layer. Each column vector $\mathbf{x}_i$ in a block has the dimension of K. Then we can compute each parent vector in the RNN tree in Equation 1.

$$\mathbf{p} = f(\mathbf{W} \times [\mathbf{x}_1{}^\top, \mathbf{x}_2{}^\top, \cdots, \mathbf{x}_b{}^\top]^\top) \tag{1}$$

where $\mathbf{x}_i \in \mathbb{R}^K, \mathbf{W} \in \mathbb{R}^{K\times bK}$. A nonlinear function such as tanh can be chosen as f. Finally, we simply use N initialized RNN trees with random weights, and concatenate the output of each tree to the final representation. That is to say, the final representation of each input image is $\mathbf{z} \in \mathbb{R}^{NK}$, as shown in Equation 2. The procedure of obtaining high level representations through multiple RNNs is also illustrated in Fig. 2.

$$\begin{aligned} \mathbf{z} &= [\mathbf{y}_1{}^\top, \mathbf{y}_2{}^\top, \cdots, \mathbf{y}_j{}^\top, \cdots, \mathbf{y}_N{}^\top]^\top \\ \mathbf{y}_j &= RNN_j(\mathbf{X}) \end{aligned} \tag{2}$$

where $\mathbf{X} \in \mathbb{R}^{K\times r\times r}, \mathbf{y}_j \in \mathbb{R}^K, \mathbf{z} \in \mathbb{R}^{NK}$.

C. Co-Training Approach

Since new sensing technology such as Kinect can provide two independent views of the same instance, co-training is particularly suitable for our case. We use co-training to increase the accuracy of RGB-D object recognition based on a large amount of unlabeled data, together with a small set of labeled data.

Algorithm 1 illustrates our co-training approach. In this paper, we use two linear SVM classifiers (C_{RGB} and C_{depth}) to model RGB images and depth images. We first train the two classifiers with two separate feature sets, F_{RGB} and F_{depth} from the labeled training sets L. The trained classifiers C_{RGB} and C_{depth} are then applied to predict the examples from the unlabeled training sets U. The most confidently predicted instances of each class by the two classifiers are transferred from U to L for the next round training. The algorithm runs until it reaches the maximum number of iteration or the unlabeled pool U is empty. When this algorithm ends, the two classifiers C_{RGB} and C_{depth} are returned.

Algorithm 1 Co-Training Approach

```
Input:
    $F_{RGB}$: RGB image features;
    $F_{depth}$: depth image features;
    L: a set of labeled training examples;
    U: a set of unlabeled training examples;
    I: the maximum number of iteration
Output:
    $C_{RGB}$:RGB classifier; $C_{depth}$: depth classifier
$i \leftarrow 0$;
repeat
    $C_{RGB} \leftarrow train(F_{RGB}, L)$;
    $C_{depth} \leftarrow train(F_{depth}, L)$;
    $C_{RGB} \rightarrow predict(F_{RGB}, U)$, for each predicted class $c_j$, choose $n_j$ most confident examples and add them to L;
    $C_{depth} \rightarrow predict(F_{depth}, U)$, for each predicted class $c_j$, choose $n_j$ most confident examples and add them to L;
    $i$++;
until $i > I$ or U is empty
return $C_{RGB}$ and $C_{depth}$;
```

The core idea behind co-training is the following: As the two classifiers C_{RGB} and C_{depth} are trained using independent feature sets, when one classifier labels an example, the other classifier can use it as a random example. In the next round training, the other classifier can benefit from this additional example. In this way, the performance of the two classifiers can be improved by learning from large amount of unlabeled data via co-training. The idea is explained in [13] as well.

At the inference time, we apply C_{RGB} and C_{depth} to the corresponding features separately. The category of the input instance is determined by the weighted sum of the two probability scores ($P_{C_{RGB}}$ and $P_{C_{depth}}$) generated by the two classifiers, defined in Equation 3. The coefficient α is determined by cross-validation.

$$c = \arg_{c_i \in \chi} Max(\alpha P^{c_i}_{C_{RGB}} + (1-\alpha) P^{c_i}_{C_{depth}}) \tag{3}$$

IV. Experiments

We evaluate our semi-supervised learning approach on recent RGB-D object recognition database [4]. This database consists of 300 household instances in 51 categories. There are about 600 RGB-D image pairs for each instance taken from different viewpoints. We take every fifth frame from each instance and give around total 41,877 RGB-depth image pairs.

Methods	Labeled	Depth	RGB	Combine
Linear SVM [4]	35k	53.1 ± 1.7	74.3 ± 3.3	81.9 ± 2.8
Kernal SVM [4]	35k	64.7 ± 2.2	74.5 ± 3.1	83.8 ± 3.5
Random Forest [4]	35k	66.8 ± 2.5	74.7 ± 3.6	79.6 ± 4.0
CNN-RNN [5]	35k	78.9 ± 3.8	80.8 ± 4.2	86.8 ± 3.3
Depth Kernel [7]	35k	78.8 ± 2.7	77.7 ± 1.9	86.2 ± 2.1
CKM [8]	35k	–	–	86.4 ± 2.3
SP+HPM [9]	35k	81.2 ± 2.3	82.4 ± 3.1	87.5 ± 2.9
SSL	7k	77.7 ± 1.4	81.8 ± 1.9	87.2 ± 1.1

TABLE I

COMPARISON OF OUR SSL ($L = 20\%$) TO MULTIPLE RECENT RESULTS

In this paper we focus on the task of category recognition. Following the experimental setting in [4], we use 10 random train/test splits. For each split, we leave one object instance out from each category for testing, and take the remaining objects as training sets. That is to say, there are 51 objects with around 6,120 images in the test sets, while there are $300 - 51 = 249$ objects with around $41,877 - 6,120 = 35,757$ images in the training sets for each split. We then randomly choose several images of each object from the training sets to add the labels while leave the remaining unlabeled, for example, we split the training sets with 20% labeled and 80% unlabeled (The procedure of splitting the training sets are repeated 10 times and the average result is used in our experiments). We use co-training to iteratively improve the performance of both the RGB classifier and the depth classifier, based on the initial 20% labeled training sets together with 80% unlabeled training sets. In every iteration of co-training, both the RGB classifier and the depth classifier choose the most confident example. This means that, the labeled training pool increases by at most 2×51 examples at every turn. The output classifiers are combined to predict the test sets based on confidence scores with the weighting coefficient $\alpha = 0.65$. We report the accuracy averaged over 10 random train/test splits.

The parameter settings of our unsupervised feature learning method are as follows: We randomly sample 500,000 subpatches from the training sets to obtain $K = 128$ filters for RGB and depth images separately via k-means clustering. The size of each RGB filter is $9 \times 9 \times 3$, compared to 9×9 for depth filter. We then resize each image to 148×148, convolve the filters over the image and perform average pooling, resulting in a 3D matrix $128 \times 27 \times 27$ output of the convolutional layer for each image. Finally, the pooled convolutional response is given into 128 random initialized RNN trees to compose the final feature representation. In this case, the final representation of each image has dimensionality of 128×128.

A. Performance Evaluation

In this section, we compare our semi-supervised learning method (SSL) to the published state-of-the-art results [4], [5], [7]–[9]. Here we label 20% (around 7,000 images) of the training sets and remain the rest (around $28,000$ images) as unlabeled for SSL, and the iteration number of co-training is set to 400. All the other methods are under condition that 100% of the training sets are labeled to train their classifiers.

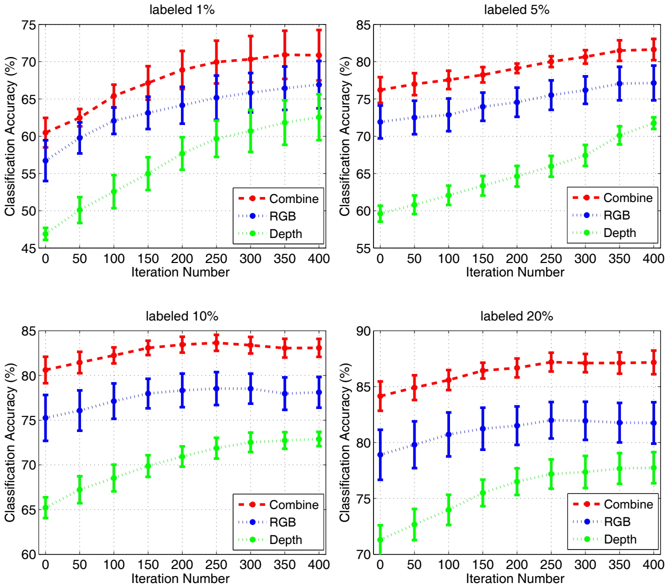

Fig. 5. Accuracy vs. number of iterations for SSL ($L = 1\%, 5\%, 10\%, 20\%$)

Table I shows the comparison results. Lie *et al.* [4] use many hand-crafted features such as SIFT, texton histogram, color histogram and efficient match kernel (EMK) to model the visual appearance, together with spin images, EMK, width, depth, and height as shape features. Bo *et al.* [7] apply multiple kernel descriptors to computing various features, including gradient, 3D shape, spin, size, kernel PCA and local binary pattern kernel descriptors. Instead of hand-crafted features, unsupervised feature learning methods are employed in [5], [8], [9] to learn the features. Our method outperforms all methods except [9], who performs 0.3% better in terms of the final recognition accuracy. However, they take advantage of additional information such as surface normals and gray scale images on top of RGB and depth images to assist the recognition task. In contrast, we only learn from the raw RGB and depth images and use a small set of labeled training sets. Furthermore, the standard deviation of recognition accuracy over the 10 test splits is the smallest via our method, which means that our method is more stable than the other methods. The main reason is that our semi-supervised learning method can exclude the training examples that are very hard to recognize for both the RGB classifier and the depth classifier. These too hard examples may degrade the performance of the two classifiers.

B. Model Analysis

Our semi-supervised learning method is closely related to the iteration number I and the labeled training size L. In order to keep all the experiments fair and balance the examples of each category in the labeled training pool, we choose the most confident examples of each predicted category in every iteration for the two classifiers. Now we analyze the influence of I and L in the following subsection.

1) Influence of Iteration Number: We plot average accuracy curves (with standard deviation) of SSL with different number of iterations for the labeled size $L = 1\%$, 5%, 10% and 20%

in Fig. 5. In every iteration, we select the most confident example in each predicted class for both the RGB and depth classifiers. In our experiments, the iteration number I varies from 0 to 400. When $I = 0$, it means that we directly use the labeled training sets to train both the RGB and depth SVM classifier, and use them to predict the test sets. As the iteration number increases, the performance of the SSL is improved very quickly until I reaches a relative high value. The main reason is that the examples selected from the unlabeled training sets by the SSL at the beginning can greatly promote the two classifiers. When enough examples are transferred from the unlabeled pool to the labeled pool, the SSL can converge to a very robust result. This characteristic keeps the SSL effective and practical because we can decide the final iteration number from a wide range and require a good performance at the same time.

2) Influence of Labeled Training Size: The labeled training size L is also an important factor of our semi-supervised learning method. For convenience, we carry out experiments on the second train/test split, changing the value of L from 1% to 30%. As well, we take the output of the 400th iteration as the final result for each labeled size L. Fig. 6 demonstrates how the growth size of the labeled training sets influence our SSL method. It's very reasonable that the SSL performs better when we use more labeled training sets, since both of the two initialized SVM classifiers can be more reliable to choose confident examples from unlabeled pool with correct labels. We can find that SSL can be competitive with the other state-of-the-art methods with only 10% labeled, and exceeds all the others except for [9] with 20% labeled. When labeled size L is larger than 20%, the increases are very small or nearly none. The main reason is that the SSL benefits less from the unlabeled data via co-training when there are much labeled data.

V. Conclusion and Future Work

This paper proposes a semi-supervised learning framework to improve RGB-D object recognition. In this framework, we apply convolutional-recursive neural networks to learn optimal appearance and spacial geometry features from RGB-D data. The two independent feature sets are used to iteratively improve the classification performance by utilizing unlabeled data via co-training. Our method can take good advantage of the complementarities of RGB and depth information in every iteration. We evaluate the proposed SSL method on a large RGB-D dataset and demonstrate that our method is competitive to other state-of-the-art methods with only a small set of labeled data. Results show that our method is able to well describe RGB-D objects and reduce the dependence on large annotated training sets in recognition procedure.

In the future, we will use on-line learning algorithms to improve the efficiency of the semi-supervised learning framework and experimentally analyze its performance on larger datasets as well.

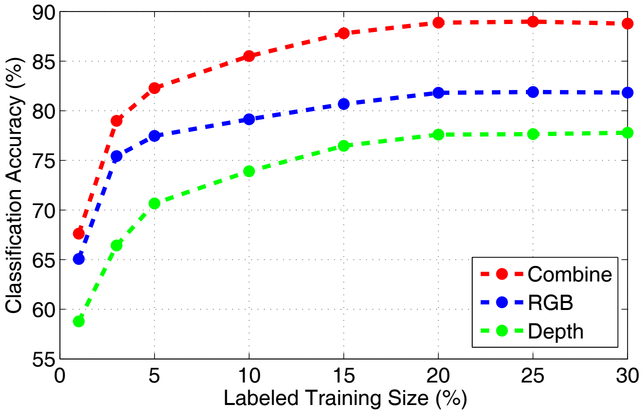

Fig. 6. Accuracy vs. labeled training size for SSL ($I = 400$)

References

[1] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Proc. ECCV*, 2004.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. CVPR*, 2005.

[3] L. Zhu, Y. Chen, Y. Lin, C. Lin, and A. Yuille, "Recursive segmentation and recognition templates for image parsing," *TPAMI*, vol. 34, no. 2, pp. 359–371, 2012.

[4] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *Proc. ICRA*, 2011.

[5] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Ng, "Convolutional-recursive deep learning for 3d object classification," in *Proc. NIPS*, 2012.

[6] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Proc. ECCV*, 2012.

[7] L. Bo, X. Ren, and D. Fox, "Depth kernel descriptors for object recognition," in *Proc. IROS*, 2011.

[8] M. Blum, J. T. Springenberg, J. Wulfing, and M. Riedmiller, "A learned feature descriptor for object recognition in rgb-d data," in *Proc. ICRA*, 2012.

[9] L. Bo, X. Ren, and D. Fox, "Unsupervised feature learning for rgb-d based object recognition," *ISER, June*, 2012.

[10] X. Zhu, "Semi-supervised learning literature survey," Computer Sciences, University of Wisconsin-Madison, Tech. Rep. 1530, 2005.

[11] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. COLT*, 1998.

[12] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *Proc. CIKM*, 2000.

[13] A. J. Joshi and N. P. Papanikolopoulos, "Learning to detect moving shadows in dynamic environments," *TPAMI*, vol. 30, no. 11, pp. 2055–2063, 2008.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[15] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, 2006.

[16] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. ICML*, 2008.

[17] R. Salakhutdinov and G. E. Hinton, "Deep boltzmann machines," in *Proc. AISTAS*, 2009.

[18] K. Yu, Y. Lin, and J. Lafferty, "Learning image representations from the pixel level via hierarchical sparse coding," in *Proc. CVPR*, 2011.

[19] A. Coates and A. Ng, "The importance of encoding versus training with sparse coding and vector quantization," in *Proc. ICML*, 2011.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.

[21] R. Socher, C. C. Lin, A. Ng, and C. Manning, "Parsing natural scenes and natural language with recursive neural networks," in *Proc. ICML*, 2011.

[22] A. Coates, A. Y. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. AISTATS*, 2011.