MMSS: Multi-modal Sharable and Specific Feature Learning for RGB-D Object Recognition

Anran Wang[1], Jianfei Cai[1], Jiwen Lu[2], and Tat-Jen Cham[1]
[1] School of Computer Engineering, Nanyang Technological University, Singapore
[2] Department of Automation, Tsinghua University, Beijing, China

Abstract

Most of the feature-learning methods for RGB-D object recognition either learn features from color and depth modalities separately, or simply treat RGB-D as undifferentiated four-channel data, which cannot adequately exploit the relationship between different modalities. Motivated by the intuition that different modalities should contain not only some modal-specific patterns but also some shared common patterns, we propose a multi-modal feature learning framework for RGB-D object recognition. We first construct deep CNN layers for color and depth separately, and then connect them with our carefully designed multi-modal layers, which fuse color and depth information by enforcing a common part to be shared by features of different modalities. In this way, we obtain features reflecting shared properties as well as modal-specific properties in different modalities. The information of the multi-modal learning frameworks is back-propagated to the early CNN layers. Experimental results show that our proposed multi-modal feature learning method outperforms state-of-the-art approaches on two widely used RGB-D object benchmark datasets.

1. Introduction

Object recognition in everyday environments is a fundamental problem in computer vision. It remains a challenging task, especially for the scenarios with clutter and highly variable illumination. With the recent advent of low-cost RGB-D cameras such as Kinect, there is an increasing amount of visual data containing both color and depth information. It is expected to enhance the inference of objects or scenes as depth measurement is robust to light and color variation.

Many methods have been proposed for RGB-D object recognition. Early works mainly focus on the design of hand-crafted features in RGB-D image descriptors. For example, Lai *et al.* [18] used hand-crafted features including spin images [14] and SIFT descriptors [23] for depth images, while textons [21], color histograms [1] and SIFT descriptors were used for color images. Lai *et al.* also utilized the bag-of-words-based *Efficient Match Kernel* (EMK) to encode local hand-crafted features, and derived an image-level representation via integrating EMK features in different spatial parts. Using the encoded features, they evaluated the recognition performance of different classifiers: a linear support vector machine, a Gaussian kernel support vector machine and a random forest classifier. Bo *et al.* [5] developed a set of kernel features for depth images that model sizes, 3D shapes, and depth edges to further improve recognition performance. Although hand-crafted features in these works boosted object recognition accuracy, the feature design process required a strong understanding of domain-specific knowledge and cannot generalize to new data domains readily. Another shortcoming for hand-crafted features is that they can only capture a subset of the features

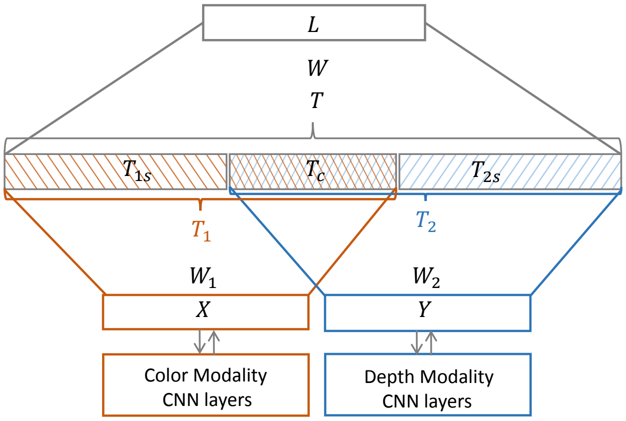

Figure 1. Illustration of our proposed multi-modal feature learning framework. The inputs X and Y are the activations of the second fully connected layers of the CNNs pre-trained on color and depth separately. The inputs are transformed by W_1 and W_2 respectively, and the transformed features T_1 and T_2 are enforced to share a common part T_c. The labeling information L is integrated in the learning process to enhance the discrimination.

that are discriminative for object recognition.

To reduce the dependency on hand-crafted features, several recent methods have been proposed for unsupervised learning of features from raw data directly, to be used in RGB-D object recognition. In particular, Bo *et al.* [6] proposed to use a *Hierarchical Matching Pursuit* (HMP) method [5] based on sparse codes derived not only from RGB-D images but also gray-scale intensities and surface normals, computed via K-SVD [2]. These features captured high-level information from local patches. Blum *et al.* [4] described a feature learning approach which learns dictionaries from RGB-D data based on K-means clustering of local-patch features, where the image patches are extracted around the interest points detected by SURF features [3]. Socher *et al.* [29] outlined a framework which integrated *Convolutional Neural Networks* (CNN) and *Recursive Neural Networks* (RNN) to learn features from color and depth separately, where the single-layer CNN is pre-trained in an unsupervised manner to produce lower-level features while the RNN learns higher-level features.

However, the relationship between different modalities have not been thoroughly investigated in these feature-learning methods for RGB-D object recognition. Most of the methods either learn features from color and depth modalities separately, or simply treat RGB-D as undifferentiated four-channel data. The major shortcoming for the separate learning is that the relation between the two modalities is ignored and the feature learning of one modality is not adjusted by the other modality. The major shortcoming for a simple four-channel learning is that the combination may not be physically meaningful and may not capitalize on different characteristics of the modalities.

To exploit the implicit dependence between different modalities, we therefore propose a multi-modal learning framework for RGB-D object recognition that treats color and depth as two modalities. At the heart of our method, we jointly explore two kinds of feature properties: shared common patterns of different modalities, and modal-specific patterns owned by individual modalities. We learn a compact and discriminative representation by transforming the data of each modality to a new feature domain with two parts: the common feature part shared by all modalities and the modal-specific part. By concatenating the shared and modal-specific features from all modalities, we obtain the final object representation which has the intended desirable properties. Supervised information is further integrated into the framework to enhance discriminative capability. CNN layers are constructed to form the input to our multi-modal feature learning framework, and the information of the multi-modal learning framework is back-propagated to the early CNN layers. The multi-modal feature learning and the back-propagation are iteratively performed until convergence. Fig. 1 shows the structure of the proposed multi-modal feature learning framework.

We wish to point out that our work is inspired by the multiview learning method proposed by Liu *et al.* in [22], which explores the consistency and complementarity properties contained in multiview data. However, their method is built based on nonnegative matrix factorization and focuses on semi-supervised learning, which cannot handle the data unseen in the training stage. In contrast, our multi-modal feature learning framework is based on matrix transformation which extracts both shared common patterns and modal-specific properties of different modalities, and is integrated with CNN based supervised deep learning for RGB-D object recognition. The method of Daumé [9] shares a similar idea with our method but focuses on transfer learning. Daumé proposed to augment data by containing general and specific versions of features. With the augmented data from both domains, their method trains a classifier which could classify data from the target domain, while leveraging information from both the source and target domains. In another recent work [34], Zhang *et al.* [34] proposed a new discriminative canonical correlation analysis (DCCA), which regards color as the main modality and depth as the auxiliary information for an object recognition task. They considered a transfer learning setting where color and depth images are used in training while only color images are used in testing, which is different in our task.

Recently, the generative power of CNN has been shown in many computer vision tasks [17, 10, 32, 15, 28, 11, 25, 30, 33, 16, 27]. There are also some approaches that use CNN for RGB-D object detection and scene labeling. For example, Couprie *et al.* [8] presented a multi-scale CNN framework for RGB-D scene labeling. Gupta *et al.* [12] proposed a CNN based approach which replaces the original depth map with three channels (horizontal disparity, height above ground, angle between point normal and inferred gravity) as the CNN input for RGB-D object detection and segmentation. However, the relationship between color and depth was ignored in these deep models. In addition, other deep structures have been used for different tasks on RGB-D data. Lenz *et al.* [20] proposed a deep learning method for robotic grasps detection. They used stacked auto-encoder structures and derived multi-modal features by encouraging each dimension of the learned features using information from only a subset of the modalities. Wang *et al.* [31] proposed a method for RGB-D scene labeling with stacked joint feature learning and encoding structures. It is worth to point out that, there is a concurrent work by Hu *et al.* [13] which shares a similar idea with us. They focus on the RGB-D activity recognition task by fusing multiple different hand-crafted features while we focus on RGB-D object recognition by integrating multi-modal feature learning with CNN.

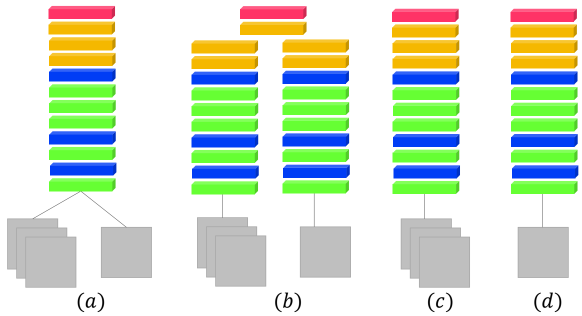

Figure 2. Different CNN structures for RGB-D data. Green, blue, yellow and red boxes indicate convolutional, pooling, fully-connected and softmax layers, respectively.

2. Proposed Approach

2.1. Conventional CNN-based Learning Structure

Of the various ways of using CNNs with RGB-D data, a straightforward approach is to combine RGB and depth data from the outset as a four-channel input to the convolutional neural network, as shown in Fig. $2(a)$ (akin to what is used in Couprie *et al.* [8] for scene labeling), where green, blue, yellow and red boxes indicate convolutional, pooling, fully-connected and softmax layers, respectively. Alternatively, discriminative features can be extracted independently from color and depth images by concatenating the activations of the second fully-connected layers of the two modalities, and feeding them into the last fully-connected layer with dense connections. From the final softmax layer, supervised information is back-propagated to the independent networks for both modalities. Such a structure is shown in Fig. $2(b)$.

2.2. Proposed Multi-modal Learning Structure

Rather than adopting the above two conventional learning structures that involve some simple fusions of color and depth data, in this paper we propose to further explore the relationship between the two modalities. We develop an architecture for multi-modal feature learning carried out in conjunction with convolutional neural networks. Specifically, we first pre-train CNNs on color and depth images separately, as shown in Fig. $2(c)$ and (d). Then, the activations of the second fully-connected layers of the two modalities are fed into the proposed multi-modal feature learning framework shown in Fig. 1.

Our main idea is that the desired features should reflect the agreement or shared properties between different modalities, while at the same time they should contain the modal-specific properties that are only captured by one of the modalities. To realize such a goal, we explicitly enforce the learned features of different modalities to share a common part. In addition, the weights between different modalities are simultaneously learned in our framework without prior knowledge of which modality is the most importance one.

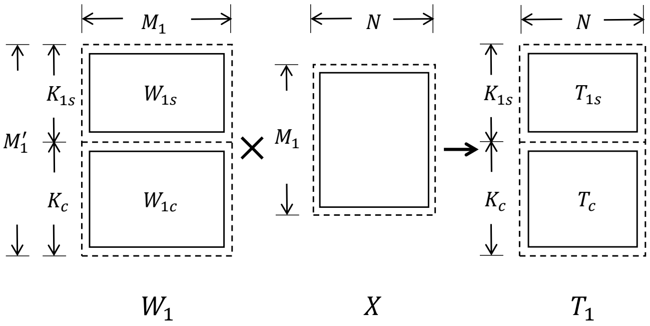

Figure 3. Illustration of the transformation matrix.

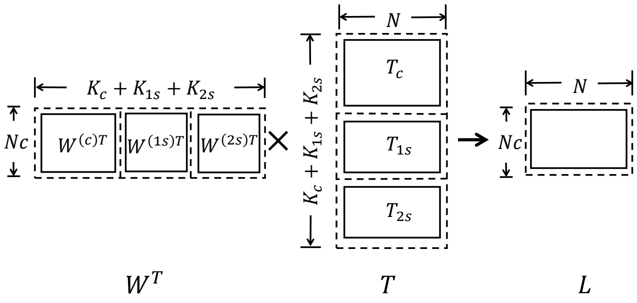

Figure 4. Illustration of the regression coefficient matrix.

In Fig. 1, $X = [x_1, x_2, \cdots, x_N] \in \mathbb{R}^{M_1 \times N}$ denotes the activations (with M_1 dimensions) of the second fully-connected layer of the CNN for color images in one data batch with N images. Similarly, $Y \in \mathbb{R}^{M_2 \times N}$ denotes the activations of the CNN for depth images in one data batch. Our objective is to learn a new feature representation T containing two sets of properties: 1) common properties shared by two modalities; and 2) modal-specific properties captured separately by individual modalities.

Let $T_1 \in \mathbb{R}^{{M_1}' \times N}$ and $T_2 \in \mathbb{R}^{{M_2}' \times N}$ denote the learned features for the color and depth modalities respectively. Here we enforce T_1 and T_2 to 1) share a common part $T_c \in \mathbb{R}^{K_c \times N}$, and 2) contain modal-specific parts $T_{1s} \in \mathbb{R}^{K_{1s} \times N}$ and $T_{2s} \in \mathbb{R}^{K_{2s} \times N}$ respectively, where ${M_i}' = K_c + K_{is}$, $i = 1, 2$, as illustrated in Fig. 3. The learned features for the two modalities are therefore: $T_1 = [T_{1s}; T_c]$ and $T_2 = [T_{2s}; T_c]$. We further denote $W_i \in \mathbb{R}^{{M_i}' \times M_i}$ as the transformation matrix for modality i. Our task is then to learn the appropriate transformation matrices W_i to obtain the features $T_1 = W_1 X$ and $T_2 = W_2 Y$. The final learned features are $T = [T_c; T_{1s}; T_{2s}]$.

Finally we require an additional matrix $W \in \mathbb{R}^{(K_c+K_{1s}+K_{2s})\times Nc}$ to map the T feature representation to actual labels for Nc number of classes, as illustrated in Fig. 4. Here we incorporate supervised learning by enforcing W^TT to be close to the ground truth label L.

2.3. Formulation

To learn features containing both shared and modal-specific properties, we formulate our cost function as

$$\begin{aligned}
&\min_{\{W_1,W_2,\alpha_1,\alpha_2,T_1,T_2,W\}} F = F_1 + F_2 + F_3 \\
&= \alpha_1(\|W_1X - T_1\|_F^2 + \left\|W_1^TT_1 - X\right\|_F^2 + \lambda_1 g(T_1)) \\
&+ \alpha_2(\|W_2Y - T_2\|_F^2 + \left\|W_2^TT_2 - Y\right\|_F^2 + \lambda_1 g(T_2)) \\
&+ \beta(\left\|W^TT - L\right\|_F^2 + \lambda_2\|W\|_{2,1}) \\
&\qquad \text{subject to} \quad \alpha_1 + \alpha_2 = 1, \alpha_1 \geq 0, \alpha_2 \geq 0
\end{aligned} \tag{1}$$

where $\|\cdot\|_F$ and $\|\cdot\|_{2,1}$ denote the Frobenius norm and $l_{2,1}$ norm. F_1 is the cost for regulating the color modality, in which: the first term enforces T_1 to be similar to the W_1-transformed X, the second term encourages the ability of T_1 to reconstruct X when back-transformed via W_1^T, while the third term g is the smooth L_1 penalty function [19]. Likewise, F_2 corresponds to the cost for regulating depth modality. Although the definitions of F_1 and F_2 seem to indicate that color and depth modalities are optimized independently, T_1 and T_2 are not in fact independent since they are explicitly required to share a common part T_c. By concatenating T_c and the modal-specific parts T_{1s}, T_{2s}, the final representation for each image is $T = [T_c; T_{1s}; T_{2s}]$. The third part, F_3 in (1), incorporates supervised information to enhance the discriminative power of the learned features, in which W is the regression coefficient matrix and the $l_{2,1}$ norm ensures W to be row-wise sparse, thus acting as a feature selector in T. Fig. 3 illustrates the matrix transformation in F_1, while Fig. 4 shows the regression coefficient matrix in F_3.

After we derive the learned matrices W, W_1 and W_2 in the training stage, the features of any test image can be directly computed as: $T_{1s} = W_{1s}X$, $T_{2s} = W_{2s}Y$, $T_c = (W_{1c}X + W_{2c}Y)/2$. With the multi-modal feature representation $T = [T_c; T_{1s}; T_{2s}]$, the final recognition result will be directly computed as W^TT.

2.4. Alternating Optimization

In this research, we employ the typical alternating optimization strategy to obtain a local optimal solution for (1). The pipeline of the algorithm is briefly described in Alg. 1. First, W, W_1, W_2 and T are initialized randomly, and α_i is initialized as 0.5. All these variables including W, W_i, T and α_i will be learned and updated in Alg. 1. Other parameters such as λ_1, λ_2 and β in (1) are set empirically.

Algorithm 1: Optimizing the proposed multi-modal feature learning framework

```
Input: Training set with two modalities: $X$, $Y$, the
       corresponding ground truth label $L$.
Output: Feature projection matrix: $W_1$, $W_2$.
        Regression coefficient matrix $W$.
Step 1 (Initialization):
    Initialize $W$, $W_1$, $W_2$, $T$, $\alpha_1$, $\alpha_2$.
Step 2 (Optimization):
    loop
        2.1. Fix $W$, $W_1$, $W_2$, $T$
             Update $\alpha_1$, $\alpha_2$ according to (5).
        2.2. Fix $W_1$, $W_2$, $T$, $\alpha_1$, $\alpha_2$,
             Update $W$ according to (7).
        2.3. Fix $W$, $W_1$, $W_2$, $\alpha_1$, $\alpha_2$,
             Update $T_c$ and $T_{is}$ according to (10).
        2.4. Fix $W$, $T$, $\alpha_1$, $\alpha_2$,
             Update $W_1$, $W_2$ according to (12).
    end loop until convergence
```

In Step 2.1, we fix W_i, W, T, and update α_i. α_1 and α_2 allows the different modalities to have different weights since they are unlikely to play the same role. When W_i, W, T are fixed, we can construct the following Lagrange function based on (1):

$$L(\alpha, \eta) = \alpha_1 C_1 + \alpha_2 C_2 + \beta C - \eta(\alpha_1 + \alpha_2 - 1). \tag{2}$$

where C_1, C_2 and C are the corresponding constant values in (1) due to fixing W_i, W, T. Unfortunately, the solution to (2) will be trivial. For example, if C_1 is less than C_2, then the solution of minimizing (2) will be: $\alpha_1 = 1$ and $\alpha_2 = 0$, which means only one modality will be used in the feature learning. Experimentally we found that this leads to suboptimal results. In order to utilize the information from different modalities, we modify our cost function to

$$\begin{aligned}
&\min_{\{W_1,W_2,\alpha_1,\alpha_2,T_1,T_2,W\}} F = F_1 + F_2 + F_3 \\
&= \alpha_1^p(\|W_1X - T_1\|_F^2 + \left\|W_1^TT_1 - X\right\|_F^2 + \lambda_1 g(T_1)) \\
&+ \alpha_2^p(\|W_2Y - T_2\|_F^2 + \left\|W_2^TT_2 - Y\right\|_F^2 + \lambda_1 g(T_2)) \\
&+ \beta(\left\|W^TT - L\right\|_F^2 + \lambda_2\|W\|_{2,1}) \\
&\qquad \text{subject to} \quad \alpha_1 + \alpha_2 = 1, \alpha_1 \geq 0, \alpha_2 \geq 0
\end{aligned} \tag{3}$$

where $p > 1$ is an additional parameter. By adding p, the objective becomes nonlinear for α_i and the two modalities will be constrained to obtain shared common pattern and modal-specific patterns in T, while at the same time keeping the most of the original information in T. In this way, the Lagrange function becomes

$$L(\alpha, \eta) = \alpha_1^p C_1 + \alpha_2^p C_2 + \beta C - \eta(\alpha_1 + \alpha_2 - 1). \tag{4}$$

By setting $\frac{\partial L(\alpha,\eta)}{\partial \alpha}$ and $\frac{\partial L(\alpha,\eta)}{\partial \eta}$ to 0, α_i can be updated as:

$$\alpha_i = \frac{(1/C_i)^{1/(p-1)}}{\sum_{i=1}^{2} (1/C_i)^{1/(p-1)}}. \tag{5}$$

In Steps 2.2-2.4, we update the other variables using the gradient descent algorithm, where the same learning rate γ is used. In particular, the regression coefficient matrix W is updated in Step 2.2. According to [24], the derivative of the cost function with respect to W can be expressed as

$$\frac{\partial F}{\partial W} = 2\beta(T(W^T T - L)^T + \lambda_2 EW) \tag{6}$$

where E is a diagonal matrix with $e_{kk} = 1/2\|w_k\|_2$, and w_k is the kth row of W. Then, W is updated according to the gradient descent rule:

$$W \leftarrow W - \gamma \frac{\partial F}{\partial W}. \tag{7}$$

In Step 2.3, the feature representation T is updated. Considering that T contains a common part T_c and modal-specific parts T_{1s} and T_{2s}, we update these three parts separately. In this way, the learned features are enforced to contain both shared common properties and modal-specific properties. The derivatives of F with respect to T_c and T_{1s}, and the mechanism for updating T_c and T_{1s} (and likewise for T_{2s}) are shown below:

$$\begin{aligned}\frac{\partial F}{\partial T_c} = {} & 2\alpha_1^p \left[(T_c - W_{1c}X) + W_{1c}(W_{1c}^T T_c - X) + \lambda_1 g'(T_c)\right] \\ & + 2\alpha_2^p \left[(T_c - W_{2c}Y) + W_{2c}(W_{2c}^T T_c - Y) + \lambda_1 g'(T_c)\right] \\ & + 2\beta W^{(c)}(W^{(c)T} T_c - L)\end{aligned} \tag{8}$$

$$\begin{aligned}\frac{\partial F}{\partial T_{1s}} = {} & 2\alpha_1^p \left[(T_{1s} - W_{1s}X) + W_{1s}(W_{1s}^T T_{1s} - X) + \lambda_1 g'(T_{1s})\right] \\ & + 2\beta W^{(1s)}(W^{(1s)T} T_{1s} - L).\end{aligned} \tag{9}$$

The common part and the modal-specific parts of T are updated according to the gradient descent rule:

$$T_c \leftarrow T_c - \gamma \frac{\partial F}{\partial T_c} \qquad T_{1s} \leftarrow T_{1s} - \gamma \frac{\partial F}{\partial T_{1s}}. \tag{10}$$

In Step 2.4, when T, W and α_i are fixed, W_i is updated in a similar way, e.g.

$$\frac{\partial F}{\partial W_1} = 2\alpha_1^p \left[(W_1 X - T_1)X^T + T_1(W_1^T T_1 - X)^T\right] \tag{11}$$

$$W_1 \leftarrow W_1 - \gamma \frac{\partial F}{\partial W_1}. \tag{12}$$

In our framework, X and Y are the activations of the second fully-connected CNN layers. The results of the multi-modal learning will then be back-propagated to the lower layers of CNN by

$$\frac{\partial F}{\partial X} = 2\alpha_1^p \left[W_1^T(W_1 X - T_1) - (W_1^T T_1 - X)\right]. \tag{13}$$

The multi-modal feature learning and the back-propagation are iteratively performed until convergence. Here we have shown the formulation for a two-modal problem. It can be straightforwardly extended to a multi-modal formulation by representing the learned features with the concatenation of the common part and the modal-specific parts from more modalities.

3. Experiments

To evaluate the effectiveness of our proposed multi-modal feature learning framework, we perform object recognition experiments on the RGB-D Object Dataset [18] and the 2D3D Dataset [7]. The details of the experiments and the results are described in the following sections.

3.1. Datasets and Experiment Setup

RGB-D Object Dataset: This dataset has 51 object classes and contains RGB-D images of 300 distinct objects taken from multiple views. They are commonplace objects such as cups, keyboards, fruits and vegetables. Each object is video-recorded with cameras mounted at three different elevation angles of approximately 30°, 45° and 60°. There are in total 207,920 RGB-D image frames, with roughly 600 images per object.

We conduct experiments for both category recognition and instance recognition. We adopt the same setup as [6], where images are sampled from every 5th frame of the videos. For the category recognition, we run the 10 random splits provided. For each split, one object from each class is sampled, resulting in 51 test objects. There are some 34,000 images for training and 6900 images for testing. For the instance recognition, we use images captured from elevation angles of 30° and 60° for training, and test on the images of the 45° angle (leave-sequence-out).

2D3D Dataset: This dataset consists of 154 objects in 14 different classes. Each object is recorded by a PMD^{TM} CamCube 2.0 time-of-flight camera with views at every 10° around the vertical axis, resulting in a total of 5544 RGB-D images. For category recognition, we adopt the setting of [7]. After excluding some classes with few examples, 6 objects of each class are used for training and the remaining objects are used for testing. For each training (testing) object, only 18 views out of 36 views are used. Eventually 82 objects in 1476 RGB-D images are regarded as training data, while 74 objects in 1332 RGB-D images are used for testing.

Architecture of CNNs: As we consider two modalities, for each modality we construct a smaller network than the one in [17] in order to ensure that the data of the two modalities can be placed in the GPU memory simultaneously. The input images are resized to 150 × 150. For the color modality, there are 96 kernels of size $7 \times 7 \times 3$ with stride 2, 96 kernels of size $5 \times 5 \times 96$ with stride 2, 112 kernels of size $3 \times 3 \times 96$ with stride 1, 128 kernels of size $3 \times 3 \times 112$ with stride 1, and 128 kernels of size $3 \times 3 \times 128$ with stride 1, for the filters of the 1st, 2nd, 3rd, 4th and 5th convolutional layers, respectively. The two fully-connected layers have the sizes of 1024 and 512 respectively. A dropout of 0.5 probability is used for the first fully-connected layer. For each 150×150 image, overlapping 142×142 images are cropped for data augmentation. There are max-pooling layers following the first, the second and the fifth convolutional layers. ReLu non-linearity [17] is applied to the output of every convolutional layer and every fully-connected layer. Note that when initializing CNNs by independently training with color and depth images as shown in Fig. 2(c) and (d), the final fully-connected layer has a size equal to the number of categories, which is then fed into the final softmax layer. We use the same architecture for both color and depth modalities, apart from the size of filters in the first convolutional layer (3 channels for color and 1 channel for depth).

Parameters setting: For our multi-modal learning framework, the dimension $M_k{}'$ of the transformed features is set to be the same as $M_k = 512$, although it could be different. Half of the $M_k{}'$-dimensional features are enforced to be the same between the two modalities, i.e. $K_c = 256$ and $K_{is} = 256$. The parameters p, β, λ_1, λ_2, γ are empirically set as 2, 1000, 1, 20, 0.001 respectively for all the experiments for both the RGB-D object and 2D3D datasets.

3.2. Results on RGB-D Object Dataset

Comparison with different baselines of using CNNs: We compare with five different CNN-based baselines: 1) CNN trained using RGB images only (Fig. 2(c)), named 'RGB CNN'; 2) CNN trained using depth images only (Fig. 2(d)), named 'Depth CNN'; 3) RGB-D used as the four-channel input to a CNN (Fig. 2(a)), named 'RGB-D CNN with 4-channel input'; 4) CNN with separate training for color and depth at the lower layers, followed by concatenating the activations of the second fully-connected layer (*fc7*) and feeding them into the last fully-connected layer (Fig. 2(b)), named 'RGB-D CNN connected at *fc7*'; 5) Similar setting with 4), but two modalities are concatenated at the fifth convolutional layer (*conv5*), named 'RGB-D CNN connected at *conv5*'.

The top part of Table 1 shows the recognition results of the five baselines on RGB-D Object Dataset. It can be seen that although simply adding depth as the fourth channel of the CNN input ('RGB-D CNN with 4-channel input') greatly improves the performance of those only using one modality ('RGB CNN' and 'Depth CNN'), extracting features separately from color and depth and connecting them at the later stage ('RGB-D CNN connected at *conv5*') performs better with significant gain. This is because separately learning features at the early stage for different modalities result in more independent features, which could prevent the CNN from primarily learning features for the predominant modality.

Following [6], we also use surface normals to replace the depth map as the input, which results in another three baselines: 6) CNN trained using surface normals only, named 'Surface Normals (SN) CNN'; 7) RGB and surface normals used as the six-channel input to a CNN, named 'RGB-SN CNN with 6-channel input'; 8) CNN with separate training for color and surface normals at the lower layers, followed by concatenating the activations of the second fully-connected layer and feeding them into the last fully-connected layer, named 'RGB-SN CNN connected at *fc7*'; 9) Similar setting with 8), but two modalities are concatenated at *conv5*, named 'RGB-SN CNN connected at *conv5*'. The comparison in Table 1 indicates that surface normals can better represent geometry information than the depth map. To further boost the performance, we use images of first 50 classes of ILSVRC2012 [26] to pretrain CNN layers of both color and surface normals, which leads to another baseline: 10) named 'RGB-SN CNN connected at *conv5* (pretrained)', achieving the best performance among all the baselines.

Table 1. Comparison of different baselines of using CNNs on RGB-D Object Dataset.

Method	Accuracy (%)
RGB CNN	74.6 ± 2.9
Depth CNN	75.5 ± 2.7
RGB-D CNN with 4-channel input	80.2 ± 1.9
RGB-D CNN connected at *fc7*	84.7 ± 2.1
RGB-D CNN connected at *conv5*	85.1 ± 2.0
Surface Normal (SN) CNN	76.3 ± 2.5
RGB-SN CNN with 6-channel input	80.7 ± 2.1
RGB-SN CNN connected at *fc7*	85.0 ± 2.4
RGB-SN CNN connected at *conv5*	85.5 ± 2.2
RGB-SN CNN connected at *conv5* (pretrained)	86.8 ± 2.1

Table 2. Comparison of our method with the best baseline. Here the CNNs in both methods are pretrained with a subset of ILSVRC2012 dataset.

Method	Accuracy (%)
RGB-SN CNN connected at *conv5* (pretrained)	86.8 ± 2.1
Ours	88.5 ± 2.2

Table 2 compares the results of our proposed multi-modal learning with the best baseline, 'RGB-SN CNN connected at *conv5* (pretrained)'. Note that our method also uses surface normals to replace depth images and uses pretrained CNN layers. It can be seen from Table 2 that our method outperforms the best baseline by 1.7% in recognition accuracy. This is mainly because our method extracts both shared common patterns and modal-specific patterns of different modalities, which cannot be achieved through simply connecting color and surface normals by a fully-connected layer.

Comparison with state-of-the-art methods: We also compare our method with state-of-the-art methods including: 1) Lai *et al.* [18]: using SIFT and spin images for depth, and SIFT, color histogram and texton histogram for color; 2) Blum *et al.* [4]: using convolutional k-means descriptors; 3) Socher *et al.* [29]: using Recursive Neural Network plus CNN; 4) Zhang *et al.* [34]: using transfer learning based method; 5) Bo *et al.* [6]: using sparse coding based feature learning with additional input channels such as gray-scale image and surface normals. The comparison results are shown in Table 3. It can be seen that our method achieves the best performance, outperforming state-of-the-art method in both category recognition and instance recognition.

The confusion matrix of our final results is shown in Fig. 5, whose diagonal elements represent the recognition accuracy for each category. Fig. 6 shows a few misclassification examples. For instance, in Fig. 6 (a) the light bulb is misclassified as a cap due to

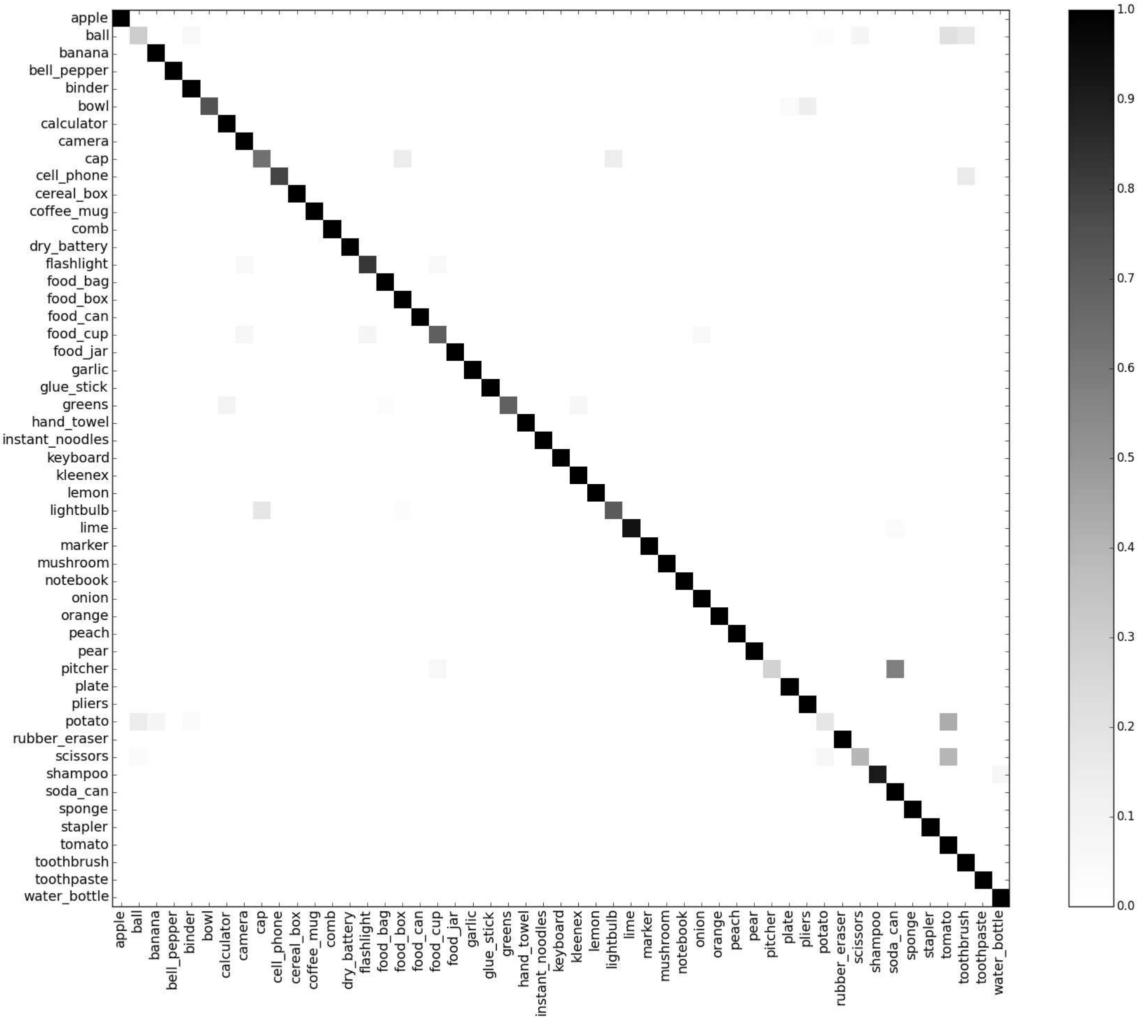

Figure 5. Confusion matrix of the category recognition results on RGB-D Object Dataset. The vertical axis shows the true labels and the horizontal axis shows the predicted labels.

Table 3. Comparison with state-of-the-art methods on RGB-D Object Dataset.

Method	Category (%)	Instance (%)
Lai *et al.* [18]	81.9 ± 2.8	73.9
Blum *et al.* [4]	86.4 ± 2.3	90.4
Socher *et al.* [29]	86.8 ± 3.3	-
Zhang *et al.* [34]	-	86.6
Bo *et al.* [6]	87.5 ± 2.9	92.8
Ours	88.5 ± 2.2	94.0

similar geometrical shapes. Likewise, the tomato is misclassified as a potato due to strong similarities in both color and shape. In Fig. 6 (b), the cellphone is misclassified as toothbrush as there is a missing part of the surface normals, which is misleading.

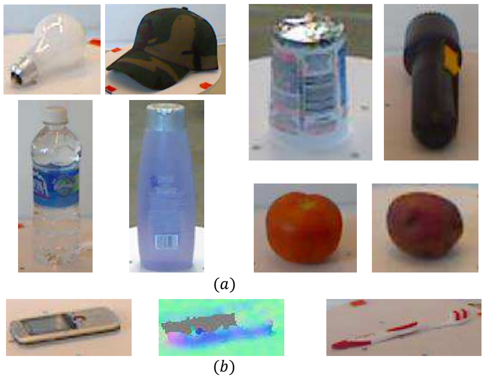

Figure 6. Misclassification examples. Misclassifications are due to similar color, texture or geometry shape.

Table 4. Comparison on 2D3D Dataset.

Method	Accuracy (%)
Browatzki *et al.* [7]	82.8
Bo *et al.* [6]	91.0
RGB-SN CNN connected at *conv5* (pretrained)	89.2
Ours	91.3

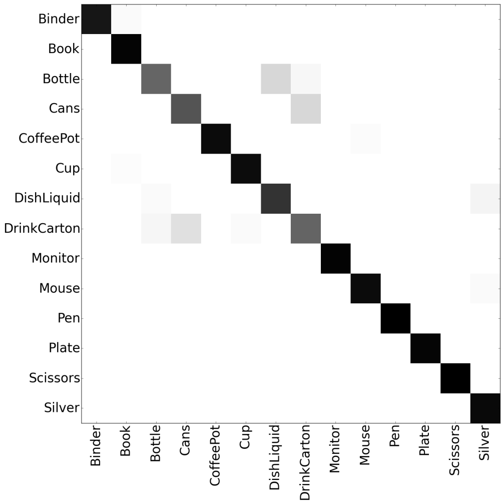

Figure 7. Confusion matrix of the category recognition results on 2D3D Dataset.

3.3. Results on 2D3D Dataset

We use the CNNs pretrained as described in Sec 3.2 and fine-tuned on RGB-D Object Dataset for the initialization. Table 4 shows the comparison between our method and the best baseline approach, 'RGB-SN CNN connected at *conv5* (pretrained)'. This table also shows the comparison of our method and state-of-the-art methods that reported results on this dataset, including Bo *et al.* [6] and Browatzki *et al.* [7], which use multiple descriptors such as 3D shape context and depth buffer for depth and multiple descriptors such as SURF and self similarity features for color. Similar remarks as those for the results on RcGB-D Object Dataset can be made here, i.e. our multi-modal feature learning method achieves superior performance to state-of-the-art methods. The confusion matrix is shown in Fig. 7.

3.4. Parameter Analysis

In our method, there are some important parameters. One is

$$R = \frac{K_c}{K_c + K_{is}}c \tag{14}$$

which ranges from 0 to 1 and controls the percentage of the shared features occupying the transformed features (note that $K_{1s} = K_{2s}$ in our setting). Fig. 8 shows how the category recognition performance on RGB-D Object Dataset split 1 varies with different R. When R is too small, the recognition accuracy is relatively low since there is only a small portion of common features between the two modalities, which cannot fully exploit the shared properties. On the other hand, when R is too large, the modal-specific features will vanish.

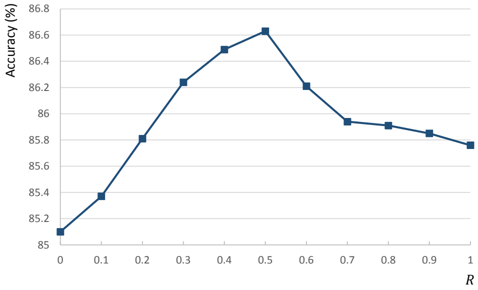

Figure 8. The effect of choosing different R.

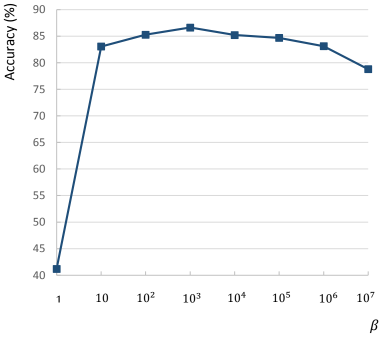

Figure 9. The effect of choosing different β.

Another important parameter is β, which balances the relation between the feature reconstruction constraints and the supervised constraints. Fig. 9 shows the accuracy performance under different β values on RGB-D Object Dataset. It can be seen that an excessively small or large weight will result in a performance drop, especially when a small weight is set on the supervised cost.

4. Conclusion

In this paper, we have proposed a CNN-based multi-modal feature learning framework for RGB-D object recognition task. Instead of fusing color and depth data from the outset or concatenating separately learned features before the classification, we extract both features with sharedc common patterns and features with modal-specific patterns in a joint framework. The experimental results show that our method integrated with CNN layers greatly boosts the performance. Our method outperforms state-of-the-art approaches on both of RGB-D Object Datast and 2D3D Dataset.

Acknowledgment

This research, which is carried out at BeingThere Centre, is supported by Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office. The research is also in part supported by MOE Tier 1 RG 138/14.

References

[1] A. E. Abdel-Hakim and A. A. Farag. Csift: A sift descriptor with color invariant characteristics. In *CVPR*, pages 1978–1983, 2006.

[2] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *TSP*, 54(11):4311–4322, 2006.

[3] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV*, pages 404–417. 2006.

[4] M. Blum, J. T. Springenberg, J. Wulfing, and M. Riedmiller. A learned feature descriptor for object recognition in rgb-d data. In *ICRA*, pages 1298–1303, 2012.

[5] L. Bo, X. Ren, and D. Fox. Depth kernel descriptors for object recognition. In *IROS*, pages 821–826, 2011.

[6] L. Bo, X. Ren, and D. Fox. Unsupervised Feature Learning for RGB-D Based Object Recognition. In *ISER*, pages 387–402, 2012.

[7] B. Browatzki, J. Fischer, B. Graf, H. Bulthoff, and C. Wallraven. Going into depth: Evaluating 2d and 3d cues for object classification on a new, large-scale object dataset. In *ICCV workshop*, pages 1189–1195, 2011.

[8] C. Couprie, C. Farabet, L. Najman, and Y. LeCun. Indoor semantic segmentation using depth information. *arXiv preprint arXiv:1301.3572*, 2013.

[9] H. Daumé III. Frustratingly easy domain adaptation. *ACL*, page 256, 2007.

[10] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, pages 184–199. 2014.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[12] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In *ECCV*, pages 345–360. 2014.

[13] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. In *CVPR*, pages 5344–5352, 2015.

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

[15] L. Kang, P. Ye, Y. Li, and D. Doermann. Convolutional neural networks for no-reference image quality assessment. In *CVPR*, pages 1733–1740, 2014.

[16] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[18] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view RGB-D object dataset. In *ICRA*, pages 1817–1824, 2011.

[19] Q. V. Le, A. Karpenko, J. Ngiam, and A. Y. Ng. Ica with reconstruction cost for efficient overcomplete feature learning. In *NIPS*, pages 1017–1025, 2011.

[20] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, 2015.

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

[22] J. Liu, Y. Jiang, Z. Li, Z.-H. Zhou, and H. Lu. Partially shared latent factor learning with multiview data. *Neural Networks and Learning Systems, IEEE Transactions on*, 2014.

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

[24] F. Nie, H. Huang, X. Cai, and C. H. Ding. Efficient and robust feature selection via joint l2, 1-norms minimization. In *NIPS*, pages 1813–1821, 2010.

[25] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *CVPR Workshop*, pages 512–519, 2014.

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *CoRR*, abs/1409.0575, 2014.

[27] P. Sermanet, S. Chintala, and Y. LeCun. Convolutional neural networks applied to house numbers digit classification. In *ICPR*, pages 3288–3291, 2012.

[28] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[29] R. Socher, B. Huval, B. Bhat, C. D. Manning, and A. Y. Ng. Convolutional-recursive deep learning for 3d object classification. In *NIPS*, 2012.

[30] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, pages 1701–1708, 2014.

[31] A. Wang, J. Lu, G. Wang, J. Cai, and T.-J. Cham. Multimodal unsupervised feature learning for rgb-d scene labeling. In *ECCV*, pages 453–467. 2014.

[32] X. Zeng, W. Ouyang, M. Wang, and X. Wang. Deep learning of scene-specific classifier for pedestrian detection. In *ECCV*, pages 472–487, 2014.

[33] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev. Panda: Pose aligned networks for deep attribute modeling. In *CVPR*, pages 1637–1644, 2014.

[34] Q. Zhang, G. Hua, W. Liu, Z. Liu, and Z. Zhang. Can visual recognition benefit from auxiliary information in training? In *ACCV*, 2014.