Multi-modal Unsupervised Feature Learning for RGB-D Scene Labeling

Anran Wang[1], Jiwen Lu[2], Gang Wang[1,2], Jianfei Cai[1], and Tat-Jen Cham[1]

[1] Nanyang Technological University, Singapore
[2] Advanced Digital Sciences Center, Singapore

Abstract. Most of the existing approaches for RGB-D indoor scene labeling employ hand-crafted features for each modality independently and combine them in a heuristic manner. There has been some attempt on directly learning features from raw RGB-D data, but the performance is not satisfactory. In this paper, we adapt the unsupervised feature learning technique for RGB-D labeling as a multi-modality learning problem. Our learning framework performs feature learning and feature encoding simultaneously which significantly boosts the performance. By stacking basic learning structure, higher-level features are derived and combined with lower-level features for better representing RGB-D data. Experimental results on the benchmark NYU depth dataset show that our method achieves competitive performance, compared with state-of-the-art.

Keywords: RGB-D scene labeling, unsupervised feature learning, joint feature learning and encoding, multi-modality.

1 Introduction

Scene labeling is an integral part of scene understanding and involves densely assigning a category label to each pixel in an image. Most previous scene labeling work dealt with outdoor scenarios [1,2,3,4,5,6]. Comparatively, indoor scenes are more challenging due to a number of factors: relative poor light condition, messy object distribution, and large variance of features for objects in different scene types. However, low-cost RGB-D cameras such as the Kinect can be used on indoor scenes to provide both color and depth measurements, leading to improvements in accuracy and robustness of labeling.

Hand-crafted features were used in several previous works on RGB-D scene labeling. These include the use of SIFT [7], KDES (kernel descriptors) [8] and other sophisticated features [9]. However, the accuracy of such feature extractors is highly dependent on variations in hand-crafting and combinations, and thus hard to systematically extend to different modalities. In addition, features are often designed for RGB and depth independently, with the shared information between RGB and depth left unexploited. Inspired by the recent success of unsupervised feature learning technique in many applications including object recognition [10] and action recognition [11], we propose to adapt the existing unsupervised feature learning technique to directly learn features from multi-modal

D. Fleet et al. (Eds.): ECCV 2014, Part V, LNCS 8693, pp. 453–467, 2014.

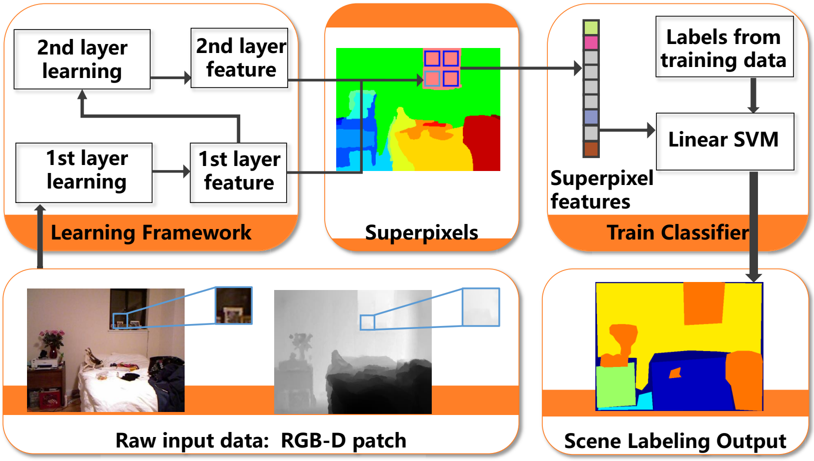


Fig. 1. Our framework for RGB-D indoor scene labeling. Our method learns features from raw RGB-D input with two-layer stacking structure. Features of the two layers are concatenated to train linear SVMs over superpixels for labeling task.

raw data in RGB-D indoor scene labeling so as to avoid the problem of handcrafting features. To the best of our knowledge, very few works have applied feature learning for RGB-D indoor scene labeling. Recently, supervised feature learning method [12], convolutional neural networks (CNN), is used for RGB-D feature learning. In another work [13], pixels of patches are encoded with selected example patches. Both of these two methods obtain limited performance.

The approach proposed in this paper attempts to learn visual patterns from RGB and depth in a joint manner via an unsupervised learning framework. This is illustrated in Figure 1. At the heart of our unsupervised learning algorithm, we perform feature learning and feature encoding jointly in a two-layer stacked structure. A dense sampling of patches is initially obtained from RGB-D images, forming the input into the learning structure. The output of the learning is a collection of superpixels, in which each superpixel represents a combination of features obtained from all patches whose centers fall into the catchment region of the superpixel. Subsequently, linear SVMs are trained to map superpixel features to scene labels.

2 Related Work

2.1 Scene Labeling

Early work on scene labeling focused on outdoor color imagery, and typically used CRF or MRF. The nodes of the graphical models were pixels [14,15], superpixels [1,4] or a hierarchy of regions [2]. Local interactions between nodes were captured by pairwise potentials, while unary potentials were used to represnt image observations, via features such as SIFT [16] and HOG [17]. An alternative inference framework was presented in [3], in which a very efficient recursive neural network (RNN) was used to greedily merge neighboring superpixels according to a learned scoring function. In a departure from the earlier approaches involving hand-crafted feature extraction, Grangier et al. [5] used convolutional networks for scene labeling. Farabet et al. [6] later adopted multiscale convolutional networks to automatically learn low and high-level textures as well as shape features from raw pixels, and further proposed the "purity" of class distributions as an optimization goal, in order to maximize the likelihood that each segment contained only one object. They achieved state-of-the-art performance on the commonly used Stanford Background [18] and SIFT Flow datasets [19].

Indoor scene labeling is a harder problem, but is recent more accessible with the advent of affordable RGB-D cameras such as Kinect. Silberman and Fergus [7] released a large-scale RGB-D dataset containing 7 scene types and 13 semantic labels. They employed RGB-D SIFT and 3D location priors as features and used MRFs to ensure contextual consistency. Koppula et al. [20] achieved high accuracy on semantic labeling of point clouds through a mixed integer optimization method. They however require the extraction of richer geometry features from 3D+RGB point clouds rather than the more limited height field from a single RGB-D image, and also depend on a computationally intensive optimization process with long running time. In an extension to Silberman and Fergus's work, Ren et al. [8] evaluated six kernel descriptors and chose four. Additionally, more comprehensive geometry features of superpixels were added to further boost performance. With these features, they achieved state-of-the-art performance on the NYU depth dataset V1. Recently, Cadena and Kosecka [9] proposed various new features including entropy for associating superpixel boundaries to vanishing points, and neighborhood planarity. A CRF is applied to the superpixels to obtain final scene labels. These methods mentioned require manual fine-tuning in feature design and also in the way that different features are combined. To reduce the dependency on hand-crafted features, Couprie et al. [12] applied the convolutional neural network method of Farabet et al. [6] to indoor RGB-D scene labeling. The depth data was treated as an additional channel besides RGB, and a multiscale convolutional network was used to ensure the features captured a larger spatial context. Although this method was demonstrated to be effective for outdoor scenes, the performance on RGB-D indoor scenes is much less satisfactory. Pei et al. [13] learned features by projecting raw pixels of patches onto selected example patches. Such an encoding method may not be powerful enough since the input raw pixel values are usually redundant and noisy.

2.2 Feature Learning

Feature learning has been applied to action recognition [11], handwritten digits recognition [21] and image classification [10,22,23]. It is also a central aspect of the RGB-D labeling framework in this paper, in which we jointly consider the two modalities of color and depth.

A number of previous work also applied feature learning to data with multiple modalities. Potamianos et al. [24] applied it to audio-visual speech recognition. Ngiam et al. [25] proposed a framework to train deep networks over multiple modalities (video and audio) using RBM (Restricted Boltzmann Machines) as basic learning units. Their method focused on learning better features for one modality when multiple modalities were present. Socher et al. [26] treated color and depth information as two modalities in object classification problem. Each modality was processed separately, wherein low-level features were extracted using a single-layer CNN and combined using RNN. Finally features from two modalities were concatenated together. However as their framework was designed only for determining a single label for each image, and did not involve classifying different regions in an image, it was not suitable for the scene labeling task in this paper.

3 Approach

3.1 Single-Layer Feature Learning Structure

Our approach is based on the unsupervised feature learning algorithm [10], which is to minimize the following objective function

$$\underset{W}{\text{minimize}} \left\|W^T W Z - Z\right\|_2^2 + \lambda_1 g(WZ) \qquad (1)$$

where Z is a set of d-dimensional raw input data vectors, i.e. $Z = [z_1, \cdots z_m] \in \mathbb{R}^{d \times m}$, $W \in \mathbb{R}^{d' \times d}$ is the transform matrix which projects Z into a d'-dimensional feature space, g is the smooth L_1 penalty function [10], and λ_1 is a tradeoff factor. Eq. (1) essentially is to seek the transformation matrix W that can minimize the reconstruction error (first term) and the penalty of the approximated orthonormal constraint (second term). The transform matrix $W \in \mathbb{R}^{d' \times d}$ is often chosen to be overcomplete, i.e. $d' > d$, for better performance, as demonstrated in the study [27]. Note that Z has gone through the whitening preprocess, i.e. the input data vectors are linearly transformed to have zero mean and identity covariance [10]. Such unsupervised feature learning method has been proven to be successful in the application of object recognition [10].

Here, we adopt Eq. (1) to learn multi-modality features for RGB-D scene labeling. Instead of learning W for color and depth information separately, we consider different modalities jointly and their relationship is implicitly reasoned. In particular, let $X = [x_1, \cdots x_m] \in \mathbb{R}^{d_1 \times m}$ denote the input RGB vectors, and $Y = [y_1, \cdots y_m] \in \mathbb{R}^{d_2 \times m}$ denote the input depth vectors. Then, Z in Eq. (1) is simply formed by cascading color and depth information as $Z = [X; Y] \in \mathbb{R}^{d \times m}$ $(d = d_1 + d_2)$.

Moreover, the previous methods [11,28] show that better performance can be achieved by further applying feature encoding over the learned features to build "bag of words" type features. However, they perform feature learning and feature encoding separately. It is clear that there is inconsistency between these two components, i.e. feature learning is not optimized for feature encoding and vice versa. Thus, in this paper, we propose to perform feature learning and feature encoding in a joint framework with the following objective function:

$$\begin{aligned} \underset{W,V,U}{\text{minimize}} \quad & \left\|W^T W Z - Z\right\|_2^2 + \lambda_1 g(WZ) + \lambda_2 \left\|WZ - UV\right\|_2^2 + \lambda_3 |V|_1 \\ \text{subject to} \quad & \|u_k\|_2 \leq 1, \ k = 1, 2, \ldots, K. \end{aligned} \tag{2}$$

where $U = [u_1, \cdots u_K] \in \mathbb{R}^{d' \times K}$ represents the dictionary which has K bases, and V denotes the feature encoding coefficients. Compared with Eq. (1), the newly added two terms in Eq. (2) aim to find sparse feature representation for the learned feature WZ. At the same time, there is a L2-norm constraint for u_k to avoid trivial solutions which just scale down V and scale up U. By jointly learning W, V and U in Eq. (2), we integrate feature learning and feature encoding into a coherent framework. With the optimized W, transformed data WZ could be encoded by more descriptive dictionary U and the final features V become more efficient.

Optimization Process. In the proposed unsupervised feature learning Eq. (2), we need to optimize W, U and V together. We solve this problem by updating three variables iteratively. W, U and V are initialized randomly. Given a training data matrix Z, we first fix U and V, the cost function can then be minimized by using the unconstrained optimizer (e.g. L-BFGS [29], CG [29]) to update W. When fixing W and U, similar to the sparse coding work [22], Eq. (2) becomes a linear regression problem with regularization on the coefficients, which can be solved efficiently by optimization over each coefficient v_m with the feature-sign search algorithm [30]. At last, when W and V are fixed, it becomes a least square problem with quadratic constraints, which can be easily solved. The optimization process is shown in Algorithm 1.

3.2 Hierarchical Structure

What we present in section 3.1 is just one-layer feature learning structure. Considering that there exists multi-level information in visual data such as intensity, edge, object, etc [31], it is often preferred to learn hierarchical features so as to describe low-level and high-level properties simultaneously. In our case, we can stack the single-layer feature learning structure to capture the higher-level features. Particularly, we first learn the low-level features using the single-layer structure. Then, the output of the low-level structure is treated as raw data input for the higher level. Considering the output of the first-layer learning structure is of high dimension, PCA is used to reduce its dimension so that the same structure can be reused for the high-layer feature learning. In the stacked structure, the input Z of higher level would contain lower-level features from the two modalities produced by the lower-level feature learning.

```
Input: Raw data from multiple modalities: $Z$
Output: Transformation matrix $W$, Dictionary $U$, Sparse encoding $V$
Step 1: Initialization.
$W$, $U$ and $V$ are randomly initialized;
Step 2: Iteratively optimize over $W$, $U$ and $V$.
while $iter \leq max\_iter$ do
    Fix $U$ and $V$:
    Solved by unconstrained optimizer L-BFGS and update $W$

    Fix $W$ and $U$:
    A linear regression problem over $V$ with L1 norm regularization on the coefficients.
    Optimized by feature-sign search algorithm and update $V$

    Fix $W$ and $V$:
    A least square problem with quadratic constraints over $U$
    Optimized by Lagrange dual and update $U$
end
```

Algorithm 1: Optimization process

3.3 Application in RGB-D Scene Labeling

In the RGB-D scene labeling application, when the input data has large size, the learning process becomes less efficient. To address this, we make use of small patch features to represent big patches. Our main framework for RGB-D labeling is as follows. We first run our unsupervised learning on randomly sampled small patches ($s \times s$) to learn the optimal transform matrix W and the dictionary U. Then, for each densely sampled big patch ($S \times S$, $S > s$), with the obtained W and U we derive the feature vector V for its overlapped $s \times s$ small patches. Features of $S \times S$ patches are then obtained by concatenating all its overlapped $s \times s$ patches' features together. Finally, superpixel technique is incorporated to ensure that pixels in the same superpixel take the same label.

Fig. 2 shows the detailed first-layer feature extraction process. In particular, we extract input raw data from two different modalities (color and depth). We convert the color image to grayscale. At the beginning, m $s \times s$ RGB-D small image patches are randomly sampled. For each $s \times s$ small patch, X is s^2-d raw color data by flatting the patch into a vector. The same goes for raw depth data Y. Concatenating them together, we have Z, $2s^2$-d data. For each $S \times S$ big patch, there are $(S - s + 1)^2$ $s \times s$ small patches. After the unsupervised feature learning process, a small patch is then represented by a sparse vector V (K-dimensional) computed from W and U. Concatenating the features of $(S - s + 1)^2$ small patches together, we obtain the features of a big patch. To avoid over-fitting caused by the high dimensionality of the big patch features, we use max-pooling to reduce the dimensionality.

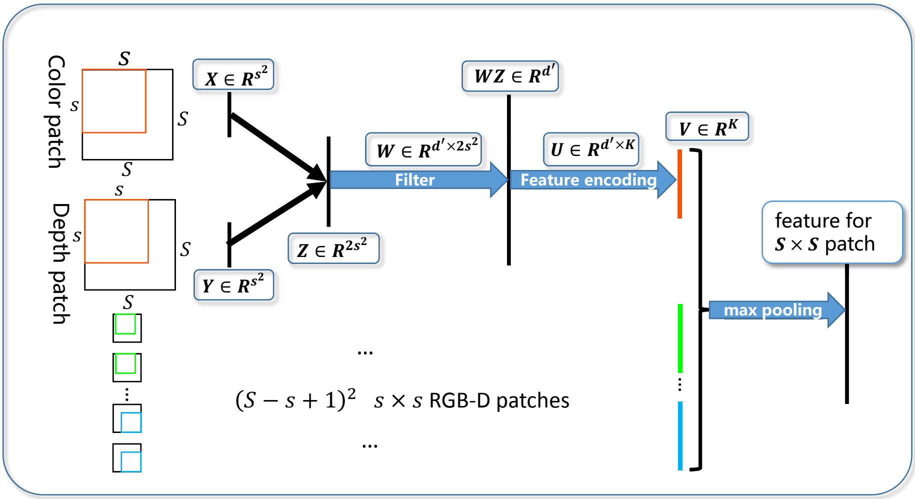


Fig. 2. Detailed illustration of feature extraction of the first layer: $s \times s$ color and depth patches are flatted to vectors X and Y. Z is the feature vector obtained by concatenating X and Y. With learned filter matrix W and dictionary U, the sparse encoding coefficients V can be derived, which represents the feature of a $s \times s$ patch. By concatenating the features of $(S - s + 1)^2$ $s \times s$ small patches, we get the feature of a $S \times S$ big patch.

To capture higher-level features, we stack two above single-layer structure together. Fig. 3 shows the two-layer feature learning structure, where the output features of the first layer are used as the input for the second layer. Specifically, the first-layer output feature vectors are further processed through dimension reduction by PCA so that the vectors could be resized to $S \times S$ data patches. Same as the first layer, $s \times s$ small patches in these $S \times S$ big patches are sampled as training data of the second layer. After the learning process of the second layer, these $S \times S$ patches are represented by the concatenated features of their $s \times s$ patches. At last, the features from the two different layers are concatenated together as the final representation of the raw patches.

In our patch size setting, we set S as 10 and s as 7 for both layers. The input data is normalized between the two modalities. We choose the dictionary size K as 1024. With learned W and U, the output of the first layer is 1024-d V. After PCA transformation, it is rescaled as 100-d data. The 100-d data is then resized to 10×10 patches, where the overlapping 7×7 patches are the training input for the second layer. By concatenating 16 1024-d features, we get a 16384-d feature vector for a 10×10 patch. Then, max-pooling is used to reduce the dimension to 1024-d for one layer. Concatenating the features of the two layers, we finally obtain a 2048-d feature for each 10×10 patch.

After feature learning process, scene labeling is done using the learned patch features. Considering that predicting the pixel-wise labeling independently could

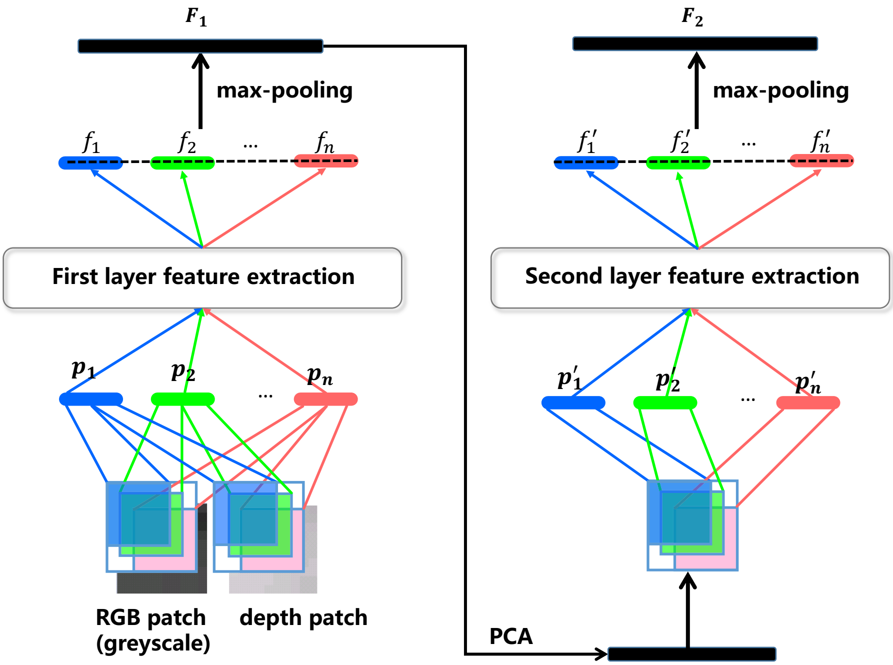


Fig. 3. Left: the unsupervised learning structure of the first layer. Right: the second layer structure. F_1 is the first-layer feature. F_2 is the second-layer feature.

be noisy and pixels with same color in local regions should take the same label, we oversegment RGB-D images using the gPb hierarchical segmentation method [32], where we follow the adaption to RGB-D images proposed by Ren et al. [8] to linearly combine the Ultrametric Contour Maps (UCM) results. The 10×10 patches are obtained by densely sampling over a grid with a unit distance of eight pixels. Finally, each superpixel is represented by averaging the features of all the patches whose centers are located in the region.

4 Experiment

Dataset. The benchmark dataset, the NYU depth dataset [7,33] including version 1 and version 2, are used for evaluation. The V1 dataset contains 2347 RGB-D images captured in 64 different indoor scenes labeled with 12 categories plus an unknown class. The V2 dataset consists of 1449 images captured in 464 different scenes.

Training Details. The parameters λ_1, λ_2 and λ_3 in Eq. (2) are empirically set to 0.1, 0.5 and 0.15. For each layer, we randomly sample 20000 7×7 patches as training data. We run 50 iterations to learn W, U in our unsupervised learning framework. Each iteration takes about 17 minutes on average on a PC with Intel i5 3.10GHz CPU and 8G memory. For a superpixel, we calculate the mean values of all its 10×10 patches' features. With the labelled superpixels in the training list, we train a 1-vs-all linear SVM for each category. For NYU depth dataset V1 [7], we use 60% data for training and 40% data for testing which is the same as that of [8]. For NYU depth dataset V2 [33], we use the training/testing splits provided by the dataset: 795 images for training and 654 images for testing.

We produce a confusion matrix whose diagonal represents the pixel-level labeling accuracy of each category. The average value of the diagonal of the confusion matrix is used as the performance metric. Note that different oversegmentation levels lead to different scene labeling results. We report the best performance of different oversegmentation levels. We would also like to point out that in this research we focus on feature learning and thus we did not further apply contextual models such as MRFs to smooth the class labels. For fair comparison, we only report the results of other methods without further smoothing.

Table 1. Class-average accuracy comparison of different methods on the NYU depth dataset V1

Results on V1		
Single feature	Ours	61.71%
	gradient KDES [8]	51.84%
	color KDES [8]	53.27%
	spin/surface normal KDES [8]	40.28%
	depth gradient KDES [8]	53.56%
Combined feature	Silberman and Fergus [7]	53.00%
	Pei et al. [13]	50.50%
	Ren et al. [8]	71.40%
	Combining our features with Ren's	72.94%

Result Comparisons on Dataset V1. Table 1 shows the average labeling results of different methods on the NYU detph dataset V1. We compare the result of our two-layer feature learning method with: 1) the result of Silberman and Fergus [7]; 2) the result of Pei et al. [13]; 3) the result of single kernel descriptor(KDES) [8]; 4) the result of Ren et al. [8] (combining four KDESs and geometry features); 5) the result of combining the features of our method and Ren's.

It can be seen from Table 1 that our method significantly outperforms the method of Silberman and Fergus, as they mainly use SIFT features on color and depth images. Our method also outperforms the method of Pei et al. [13], as they use selected patches which are usually redundant and noisy in encoding. However, our result does not outperform that of Ren et al. [8]. We argue that Ren et al. [8] evaluated six kernel based features, integrated four of them: gradient, color, depth gradient, spin/surface normal, and developed a sophisticated

Table 2. Class-average accuracy results of our method with different settings on the NYU depth dataset V1

Result on V1	
Our method (first layer)	54.76%
Sparse coding after feature learning	45.67%
k-means feature encoding after feature learning	22.32%
Separate learning from two modalities with our cost function	50.74%
Our method (second layer)	52.90%

Table 3. Individual class label accuracy on the NYU depth dataset V1 with only one-layer features. Second column: learning from color modality alone. Third column: learning from two modalities separately. Forth column: joint learning from two modalities. The bold numbers are to indicate the cases that extra depth features hurt the performance. In contrast, the performance is boosted when jointly learning from two modalities for all the categories.

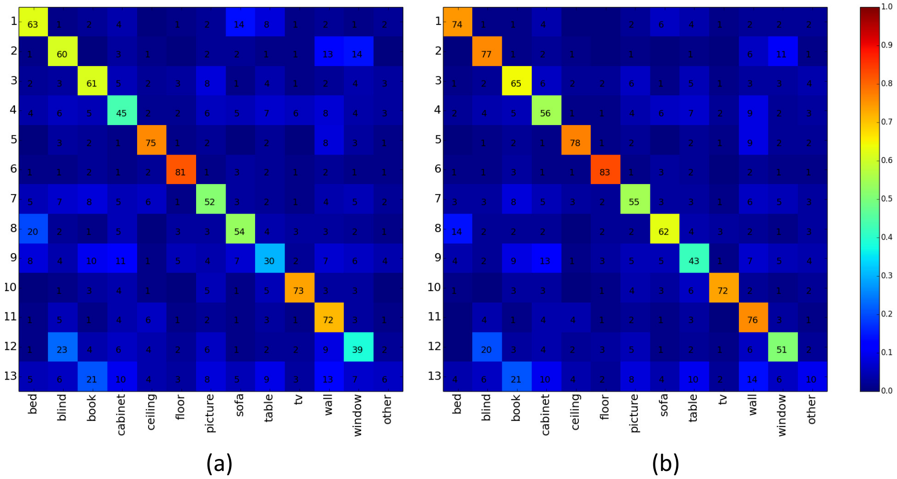
	Learning only from color modality	Separate learning from two modalities	Joint learning from two modalities
bed	58.08%	**57.11%** ↓	62.55%
blind	56.63%	**55.19%** ↓	60.40%
book	54.88%	**47.97%** ↓	60.99%
cabinet	34.66%	38.40%	44.77%
ceiling	61.77%	79.52%	75.36%
floor	67.70%	83.20%	81.37%
picture	35.56%	47.35%	51.71%
sofa	43.99%	57.37%	54.48%
table	19.27%	23.68%	30.40%
tv	47.56%	59.83%	73.15%
wall	70.75%	**69.73%** ↓	71.62%
window	33.69%	36.50%	38.73%
other	3.70%	3.87%	6.28%

method to carefully select the best combination of the four features. In contrast, we just learn a single type of features directly from raw pixel values. If we compare our result with that of each single descriptor of [8], our method achieves superior performance. Compared to [8], our method does not need any detailed hand-crafting of features. Moreover, by combining our and Ren's features together, the classification accuracy can be further improved, suggesting that our features capture visual patterns which cannot be captured by those of [8].

Detailed Evaluations on Different Settings. Here we give detailed evaluations on our method with only one-layer features under different settings. In particular, we compare the following five setups: 1) our method with the features learned from the first layer; 2) separate learning: conducting feature

Table 4. Class labeling accuracy on the NYU depth dataset V2.

	Ground	Structure	Furniture	Props	class average
Ours	90.1%	81.4%	46.3%	43.3%	65.3%
Couprie et al. [12]	87.3%	45.3%	35.5%	86.1%	63.5%
Cadena and Kosecka [9]	87.3%	60.6%	33.7%	74.8 %	64.1%



(a)

	bed	blind	book	cabinet	ceiling	floor	picture	sofa	table	tv	wall	window	other
1	63	1	2	4		1	2	14	8	1	2	1	2
2	1	60		3	1		2	2	1	1	13	14	
3	2	3	61	5	2	3	8	1	4	1	2	4	3
4	4	6	5	45	2	2	6	5	7	6	8	4	3
5		3	2	1	75	1	2		2		8	3	1
6	1	1	2	1	2	81	1	3	2	1	2	1	1
7	5	7	8	5	6	1	52	3	2	1	4	5	2
8	20	2	1	5		3	3	54	4	3	3	2	1
9	8	4	10	11	1	5	4	7	30	2	7	6	4
10		1	3	4	1		5	1	5	73	3	3	
11	1	5	1	4	6	1	2	1	3	1	72	3	1
12	1	23	4	6	4	2	6	1	2	2	9	39	2
13	5	6	21	10	4	3	8	5	9	3	13	7	6

(b)

	bed	blind	book	cabinet	ceiling	floor	picture	sofa	table	tv	wall	window	other
1	74	1	1	4			2	6	4	1	2	2	2
2		77	1	2	1		1		1		6	11	1
3	1	2	65	6	2	2	6	1	5	1	3	3	4
4	2	4	5	56	1	1	4	6	7	2	9	2	3
5		1	2	1	78	1	2		2	1	9	2	2
6	1	1	3	1	1	83	1	3	2		2	1	1
7	3	3	8	5	3	2	55	3	3	1	6	5	3
8	14	2	2	2		2	2	62	4	3	2	1	3
9	4	2	9	13	1	3	5	5	43	1	7	5	4
10	1		3	3		1	4	3	6	72	2	1	2
11		4	1	4	4	1	2	1	3		76	3	1
12		20	3	4	2	3	5	1	2	1	6	51	2
13	4	6	21	10	4	2	8	4	10	2	14	6	10

Fig. 4. The confusion matrices of: (a) our results with one-layer structure; (b) our results with two-layer structure.

learning with Eq. (1) to get filter matrix W and then performing sparse coding to encode filtered data WZ; 3) conducting feature learning with Eq. (1) and then use k-means clustering result as hard quantization to encode filtered data WZ; 4) learning features from two modalities separately with our cost function; 5) our method with the features learned from the second layer alone. Table 2 shows the results of the five different setups.

Comparing the results of methods 1, 2 and 3 in Table 2, we can see that joint feature learning and encoding performs much better than the methods using separate processing. Particularly, for method 2, we run 50 iterations to update W and then conduct sparse coding for 50 iterations to encode WZ. Compared with the way of iteratively updating W and U for 50 iterations in method 1, method 2 cannot guide W to help find descriptive U. For method 3, important feature information is lost when quantized by k-means.

Comparing the results of methods 1 and 4 in Table 2, we can see that joint learning from two modalities outperforms separate learning. This is because separate learning ignores the correlation information between the two modalities, for which the extra features learned from depth alone might hurt the perfor-

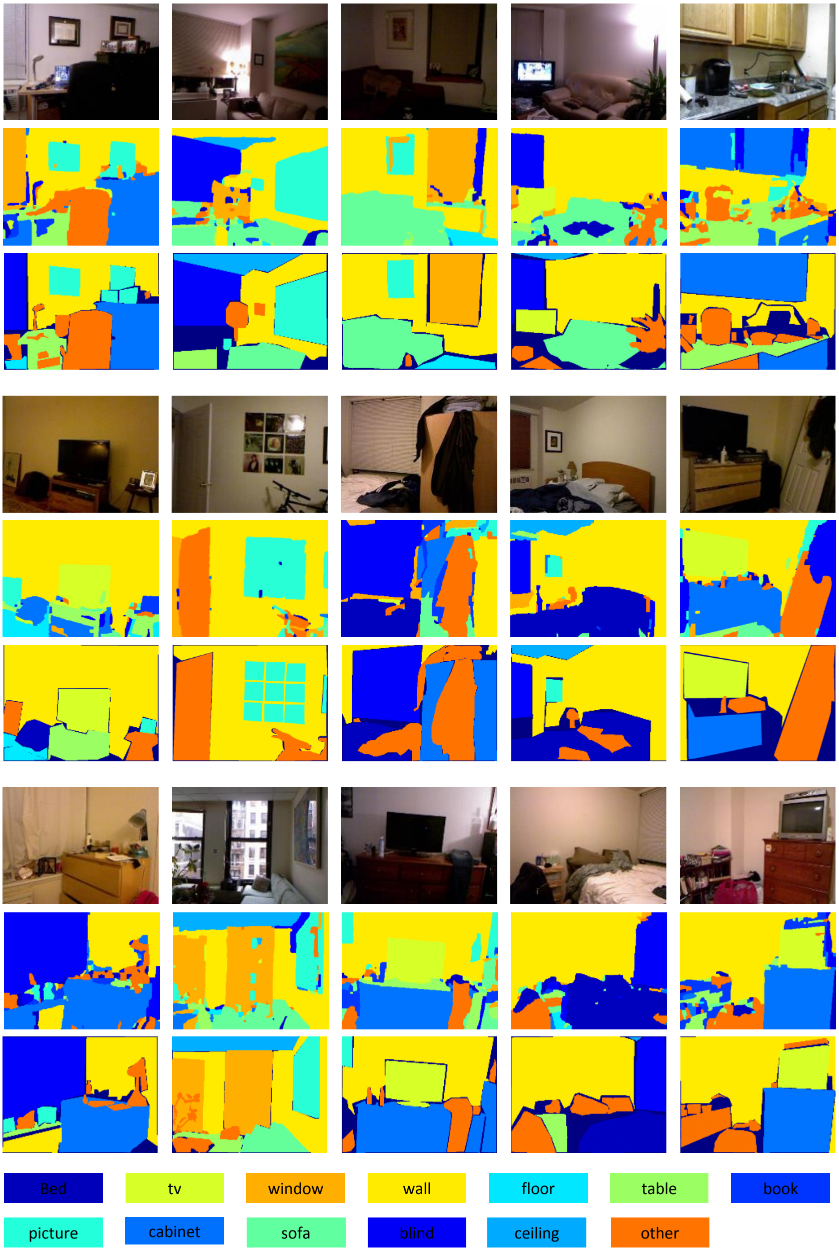


Fig. 5. 15 example results. Rows 1st, 4th and 7th: color images. Rows 2nd, 5th and 8th: the results of combining our features and Ren's features. Rows 3rd, 6th and 9th: ground truth. Note that since we focus on feature learning, we did not use CRFs or MRFs to smooth the labels. So the results might look a bit noisy.

Table 5. Labeling accuracy on the NYU depth dataset V2. Second column: the results of Couprie et al. [12]. Third column: the results of our method with two-layer structure.

	Couprie et al. [12]	Ours
bed	38.1%	47.6%
objects	8.7%	12.4%
chair	34.1%	23.5%
furnit.	42.4 %	16.7%
ceiling	62.6%	68.1%
floor	87.3%	84.1%
deco.	40.4%	26.4%
sofa	24.6%	39.1%
table	10.2%	35.4%
wall	86.1%	65.9%
window	15.9%	52.2%
books	13.7%	45.0%
TV	6.0%	32.4%
class average	36.2%	42.2%

mance, as shown in Table 3. On the contrary, our algorithm implicitly infers the correlation between the two modalities and could find better combination of them, which leads to better performance for all the classes (see Table 3).

Comparing the results of methods 1 and 5 in Table 2, we can see that the high-level features captured by the second layer alone are not sufficient. Only when combining with low-level features together, we can achieve a performance improvement of 7% (see Table 1), compared with using the first-layer features alone. The detailed comparison of confusion matrixes between one-layer learning and two-layer learning is shown in Fig. 4.

Result Comparisons on Dataset V2. We also compare our results on NYU depth dataset V2 with the following two existing works that have reported results on the dataset V2: 1) Couprie et al. [12]; 2) Cadena and Kosecka [9]. [9] includes a lot of hand-crafted appearance and geometry features. Couprie et al. [12] automatically learns features from raw data input which is similar to our method. Table 4 shows the labeling accuracy results on the NYU depth dataset V2, where the four structural classes, structure, floor, furniture and prop, are often used for comparison. It can be seen that our method achieves the best average accuracy, although our method performs poorly on the prop class, which contains many small table items.

Considering that both [12] and our method are feature-learning based approaches, we give a further comparison between them using the 13 fine categories defined in [12]. Table 5 shows the comparison results. It can be seen that the performance of [12] is not satisfactory, although their hierarchical convolutional neural network system succeeds in scene labeling on outdoor color image dataset. Our average accuracy outperforms theirs by 6%.

Fig. 5 shows some examples of pixel labeling results. The visualization results demonstrate that the learned local features can well represent objects in the scene. Note that since our work focuses on feature learning, we did not use CRFs or MRFs to smooth class labels.

5 Conclusion

In this paper, we presented an unsupervised feature learning method that learns features from RGB-D data for scene labeling task. We pose it as a multi-modality learning problem containing color and depth. Our method considers unsupervised feature learning and feature encoding problem together and implicitly infers the relationship between two modalities. By stacking the learning framework, our method could learn hierarchical features. Linear SVMs are trained on superpixels to produce the final labeling. We carried experiments on NYU depth dataset V1 and V2 and get comparable results with state-of-the-art methods including those use hand-crafted features and those learns features from raw data.

Acknowledgment. This research, which is carried out at BeingThere Centre, is supported by Singapore MoE AcRF Tier-1 Grant RG30/11 and the 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 RG84/12 and SERC 1321202099.

References

1. Micusik, B., Kosecka, J.: Semantic segmentation of street scenes by superpixel co-occurrence and 3d geometry. In: ICCV Workshops, pp. 625–632 (2009)
2. Lempitsky, V.S., Vedaldi, A., Zisserman, A.: Pylon model for semantic segmentation. In: NIPS, pp. 1485–1493 (2011)
3. Socher, R., Lin, C.C., Manning, C., Ng, A.Y.: Parsing natural scenes and natural language with recursive neural networks. In: ICML, pp. 129–136 (2011)
4. Galleguillos, C., Rabinovich, A., Belongie, S.: Object categorization using co-occurrence, location and appearance. In: CVPR, pp. 1–8 (2008)
5. Grangier, D., Bottou, L., Collobert, R.: Deep convolutional networks for scene parsing. In: ICML Deep Learning Workshop (2009)
6. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. IEEE Trans. Pattern Anal. Mach. Intell. 35(8), 1915–1929 (2013)
7. Silberman, N., Fergus, R.: Indoor scene segmentation using a structured light sensor. In: ICCV Workshops, pp. 601–608 (2011)
8. Ren, X., Bo, L., Fox, D.: Rgb-(d) scene labeling: Features and algorithms. In: CVPR, pp. 2759–2766 (2012)
9. Cadena, C., Košecka, J.: Semantic parsing for priming object detection in rgb-d scenes. In: Workshop on Semantic Perception, Mapping and Exploration (2013)
10. Le, Q.V., Karpenko, A., Ngiam, J., Ng, A.Y.: Ica with reconstruction cost for efficient overcomplete feature learning. In: NIPS, pp. 1017–1025 (2011)
11. Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: CVPR, pp. 3361–3368 (2011)

12. Couprie, C., Farabet, C., Najman, L., LeCun, Y.: Indoor semantic segmentation using depth information. arXiv preprint arXiv:1301.3572 (2013)
13. Pei, D., Liu, H., Liu, Y., Sun, F.: Unsupervised multimodal feature learning for semantic image segmentation. In: IJCNN (2013)
14. He, X., Zemel, R., Carreira-Perpindn, M.: Multiscale conditional random fields for image labeling. In: CVPR (2004)
15. Shotton, J., Winn, J.M., Rother, C., Criminisi, A.: *textonBoost*: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 1–15. Springer, Heidelberg (2006)
16. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV, pp. 1150–1157 (1999)
17. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893 (2005)
18. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV (2009)
19. Liu, C., Yuen, J., Torralba, A.: Sift flow: Dense correspondence across scenes and its applications. PAMI (2011)
20. Koppula, H.S., Anand, A., Joachims, T., Saxena, A.: Semantic labeling of 3d point clouds for indoor scenes. In: NIPS, pp. 244–252 (2011)
21. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Computation 18(7), 1527–1554 (2006)
22. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: CVPR, pp. 1794–1801 (2009)
23. Kumar, A., Rai, P., Daumé III, H.: Co-regularized multi-view spectral clustering. In: NIPS, pp. 1413–1421 (2011)
24. Potamianos, G., Neti, C., Luettin, J., Matthews, I.: Audio-visual automatic speech recognition: An overview. Issues in Visual and Audio-Visual Speech Processing 22, 23 (2004)
25. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep learning. In: ICML, pp. 689–696 (2011)
26. Socher, R., Huval, B., Bhat, B., Manning, D.: C., Ng, A.Y.: Convolutional-Recursive Deep Learning for 3D Object Classification. In: NIPS, pp. 665–673 (2012)
27. Coates, A., Ng, A.Y., Lee, H.: An analysis of single-layer networks in unsupervised feature learning. In: International Conference on Artificial Intelligence and Statistics, pp. 215–223 (2011)
28. Wang, H., Ullah, M.M., Klaser, A., Laptev, I., Schmid, C., et al.: Evaluation of local spatio-temporal features for action recognition. In: BMVC, pp. 124.1–124.11 (2009)
29. Schimidt, M.: minfunc. (2005)
30. Lee, H., Battle, A., Raina, R., Ng, A.Y.: Efficient sparse coding algorithms. In: NIPS, pp. 801–808 (2006)
31. Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: ICML, pp. 609–616 (2009)
32. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. PAMI, 898–916 (2011)
33. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part V. LNCS, vol. 7576, pp. 746–760. Springer, Heidelberg (2012)