Unsupervised Joint Feature Learning and Encoding for RGB-D Scene Labeling

Anran Wang, *Student Member, IEEE*, Jiwen Lu, *Senior Member, IEEE*, Jianfei Cai, *Senior Member, IEEE*, Gang Wang, *Member, IEEE*, and Tat-Jen Cham

***Abstract*— Most 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 propose an unsupervised *joint feature learning and encoding* (JFLE) framework for RGB-D scene labeling. The main novelty of our learning framework lies in the joint optimization of feature learning and feature encoding in a coherent way, 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. Moreover, to explore the nonlinear intrinsic characteristic of data, we further propose a more general *joint deep feature learning and encoding* (JDFLE) framework that introduces the nonlinear mapping into JFLE. The experimental results on the benchmark NYU depth dataset show that our approaches achieve competitive performance, compared with the state-of-the-art methods, while our methods do not need complex feature handcrafting and feature combination and can be easily applied to other data sets.**

***Index Terms*— RGB-D scene labeling, unsupervised feature learning, joint feature learning and encoding, multi-modality.**

I. 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 [7], [8], [10], [25], [28], [39]. Comparatively, indoor scenes are more challenging due to a number of factors: relatively 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, which leads to improvements in accuracy and robustness of the scene labeling task.

Hand-crafted features were used in several previous works on RGB-D scene labeling. These include the use of SIFT [36], LBP, HOG, texton [17], spin image [36], class-specific location [17], planarity [2], height above the ground, 3D shape contexts, size features [11], and some sophisticated features such as KDES (*kernel descriptors*) [33]. However, the labelling accuracy of such feature extractors is highly dependent on variations in hand-crafting and combinations, and thus hard to systematically extend to different datasets or 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 [21] and action recognition [22], in this paper we propose to adapt the existing unsupervised feature learning technique to directly learn features from multi-modal raw data for RGB-D indoor scene labeling so as to avoid the limitations of hand-crafted features. Several works have applied feature learning for RGB-D indoor scene labeling. In [31], pixels of patches are encoded with example patches selected from input data. Another work [4] uses multi-scale *convolutional neural networks* (CNN), a supervised feature learning method, for RGB-D feature learning. Both methods obtained limited performance for indoor scene labeling. Recently, two supervised methods [6], [12] which are based on pre-trained CNN achieve current state-of-the-art performance. Although both supervised and unsupervised methods can be used for feature learning, in this research we focus on unsupervised approach, which derives features without using label information and thus it can work for datasets lack of sufficient labelled data.

In particular, the approach proposed in this paper attempts to learn visual patterns from RGB and depth in a joint manner via an unsupervised learning framework. At the heart of our unsupervised learning algorithm, we perform feature learning and feature encoding jointly in a two-layer stacked structure, called *joint feature learning and encoding* framework (JFLE), which is very different from the conventional way of feature learning followed by feature encoding (e.g. BOW, sparse coding), i.e. two separate processes. Moreover, to explore the nonlinear intrinsic characteristics of data, we further extend the JFLE framework to a more general framework called *joint deep feature learning and encoding* (JDFLE), which uses a deep model with stacked nonlinear layers to model the input data. Fig. 1 illustrates the proposed overall framework.

Manuscript received February 25, 2015; revised June 27, 2015; accepted July 24, 2015. Date of publication August 11, 2015; date of current version August 18, 2015. This work was supported in part by the Singapore National Research Foundation under its International Research Centre at the Singapore Funding Initiative, and administered by the Interactive Digital Medi Programme Office, in part by the Ministry of Education (MOE) Tier 1 under Grant RG 138/14, in part by the Singapore MOE Tier 2 under Grant ARC28/14, and in part by the Agency for Science, Technology and Research through the Science and Engineering Research Council, Singapore, under Grant PSF1321202099. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Vladimir Stankovic.

A. Wang, J. Cai, and T.-J. Cham are with the School of Computer Engineering, Nanyang Technological University, Singapore 639798 (e-mail: awang001@e.ntu.edu.sg; asjfcai@ntu.edu.sg; astjcham@ntu.edu.sg).

J. Lu is with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: elujiwen@gmail.com).

G. Wang is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: wanggang@ntu.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2015.2465133

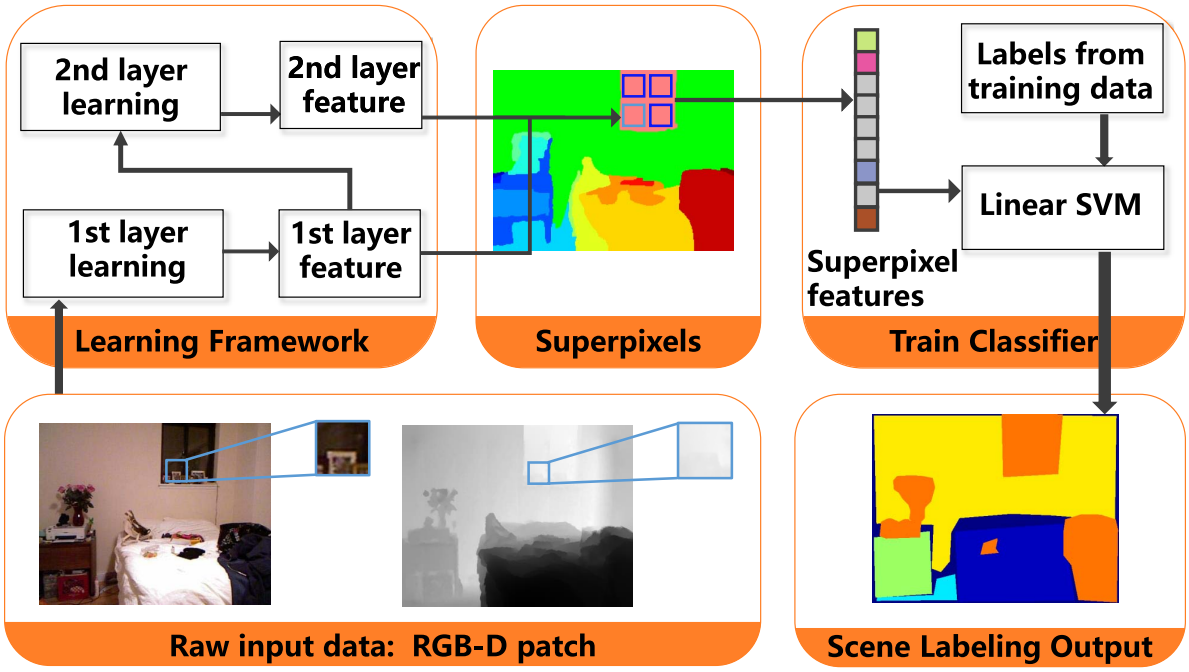


Fig. 1. Our framework for RGB-D indoor scene labeling. Our method learns features from raw RGB-D input with two-layer stacking structure. Features of the two layers are concatenated to train linear SVMs over superpixels for labeling task.

The input to the learning structure (either JFLE or JDFLE) is a set of patches densely sampled from RGB-D images, and the learning output is the set of corresponding path features, which are then combined to generate superpixel features. Finally, linear SVMs are trained to map superpixel features to scene labels.

The contributions of this paper are threefold. First, we propose an unsupervised joint feature learning and feature encoding framework (JFLE) that makes feature learning and feature encoding help each other in a coherent manner. Second, we further develop a more general JDFLE framework which replaces the linear mapping in JFLE by multiple nonlinear sub-layers. Third, we apply our joint frameworks on RGB-D scene labeling. We conduct extensive experiments to demonstrate the effectiveness of the proposed joint frameworks.

We would like to point out that part of this research (specifically, the JFLE framework) has been report in [40]. Compared with the conference version, this journal submission contains substantial extension including the newly developed JDFLE framework that is a more general nonlinear learning model, more experiments and comparisons, and more detailed descriptions.

The rest of this paper is organized as follows. Section II introduces related works on scene labeling and feature learning. Section III describes the JFLE method in detail. Section IV presents the more general JDFLE framework. Section V gives the experimental results and Section VI concludes the paper.

II. Related Work

A. 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 [13], [35], superpixels [8], [28] or a hierarchy of regions [25]. Local interactions between nodes were captured by pairwise potentials, while unary potentials were used to represnt image observations, via features such as SIFT [27] and HOG [5]. An alternative inference framework was presented in [39], 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.* [10] used convolutional networks for scene labeling. Farabet *et al.* [7] later adopted multiscale convolutional networks to automatically learn low and high-level texture and 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 contains only one object. They achieved state-of-the-art performance on the commonly used Stanford Background [9] and SIFT Flow datasets [26].

Indoor scene labeling is a harder problem, but it has become more accessible recently with the advent of affordable RGB-D cameras such as Kinect. Silberman and Fergus [36] 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.* [18] achieved high accuracy on semantic labeling of point clouds via 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.* [33] evaluated

six kernel descriptors and eventually chose four of them. Besides, more comprehensive geometry features of superpixels were added to further boost the performance. With the additional features, state-of-the-art performance on the NYU depth dataset V1 has been achieved. Cadena and Košecka [2] proposed various new features including entropy for associating superpixel boundaries to vanishing points, and neighborhood planarity. Gupta *et al.* [11] proposed an object boundary detection method which naturally combines color and depth information. With better bottom-up segmentation, they extract generic and class-specific features to encode superpixels. Recently, Müller and Behnke [29] proposed to fuse color, depth information with a random forest approach. They built a CRF model not only based on spatial relations in 2D but also geometry relations in 3D. Khan *et al.* [17] proposed a CRF framework with long range interaction based on geometry features as higher order potentials besides the unary and pairwise potentials. Hermans *et al.* [14] proposed a CRF-based scene labeling method and created a consistent 3D semantic reconstruction of indoor scenes based on 2D results. All these methods mentioned above require manual fine-tuning in feature design and also in the combination of different features.

To reduce the dependency on hand-crafted features, several feature learning methods have been proposed to learn features for RGB-D scene labeling. Pei *et al.* [31] 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. Lai *et al.* [20] proposed an unsupervised method for labeling point clouds of tabletop objects with RGB-D videos as the input. They presented a hierarchical sparse coding technique for learning features from 3D point cloud data. Recently, convolutional neural network has consistently achieved superior performance in computer vision tasks. Couprie *et al.* [4] applied the *convolutional neural network* (CNN) method of Farabet *et al.* [7] 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 capture 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. Gupta *et al.* [12] applied CNN framework for color image detection and semantic segmentation of RGB-D images. They encoded depth image with 3-channel HHA (horizontal disparity, height above ground, and the angle the pixels local surface normal makes with the inferred gravity direction) which could make use of the pre-trained CNN with color images. Eigen and Fergus [6] proposed to use an end-to-end CNN structure to predict depth, surface normals and semantic labels. Instead of extracting CNN features from superpixels, they directly predicted pixel-level labels. To our knowledge, their method achieved state-of-the-art performance. These CNN-based methods are supervised, while our method is unsupervised.

Our proposed method belongs to the category of using learned features for RGB-D scene labelling. The uniqueness of our method lies in the joint optimization of feature learning and feature encoding while the existing approaches either only focus on one of them or perform them separately.

B. Feature Learning

Feature learning has been applied to action recognition [22], handwritten-digit recognition [15] and image classification [19], [21], [43]. In this paper, we investigate how to apply feature learning for RGB-D scene labeling, where we 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.* [32] applied it to audio-visual speech recognition. Ngiam *et al.* [30] 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.* [38] 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. Jhuo *et al.* [16] proposed an unsupervised feature learning for RGB-D image classification. The structure is constructed based on [21]. As their frameworks were designed only for determining a single label for each image, it was not suitable for the scene labeling task in this research. In addition, the key of our proposed method is performing feature learning and encoding in a joint framework, whose optimization target is different from theirs.

In addition, we also introduce nonlinear mapping as the extension to our previous conference version. There have been works about nonlinear mapping. For example, Xie *et al.* [42] proposed to optimize the dictionary and sparse coding on a Riemannian manifold instead of Euclidean space. Here we use a general deep structure to learn the nonlinear mapping, instead of based on some prior assumption.

III. Joint Feature Learning and Encoding

In this section, we describe our proposed basic feature learning framework, called *joint feature learning and encoding* (JFLE), which performs joint feature learning and encoding in one optimization framework.

A. Single-Layer Feature Learning Structure

Our approach is based on the unsupervised feature learning algorithm [21], which is to minimize the following objective function

$$\min_{W} \left\| W^T W Z - Z \right\|_2^2 + \lambda_1 g(WZ) \tag{1}$$

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 [21], 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 smooth L1 penalty on learned features WZ (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 [3]. 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 [21]. Such unsupervised feature learning method has been proven to be successful in the application of object recognition [21].

The previous methods [22], [41] show that better performance can be achieved by further applying feature encoding over the learned features to build "bag of words" type of features. However, they perform feature learning and feature encoding separately, which might cause inconsistency between the two components since feature learning is not optimized for feature encoding and feature encoding is also not optimized for feature learning.

Therefore, motivated by the above observation, in this paper we propose to perform feature learning and feature encoding in a joint framework with the following objective function:

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

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, where the two parts are optimized to help each other. With the optimized W, transformed data WZ could be encoded by more descriptive dictionary U and the final features V become more efficient.

In particular, for RGB-D scene labeling we learn multi-modality features. Instead of learning W for color and depth information separately, we consider different modalities jointly and their relationship is implicitly reasoned. Specifically, 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 $Z = [X; Y] \in \mathbb{R}^{d \times m}$ ($d = d_1 + d_2$).

B. 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 [34], CG [34]) to update W. When fixing W and U, similar to the sparse coding work [43], 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 [23]. 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.

Algorithm 1 Optimization Process

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

C. Hierarchical Feature Learning

What we present in section III-A is just one-layer feature learning structure. Considering that there exists multi-level information in visual data such as intensity, edge, object, etc [24], it is often preferred to learn hierarchical features so as to describe low-level and high-level properties simultaneously. In our case, we 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 the 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.

D. Application on RGB-D Scene Labeling

For 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

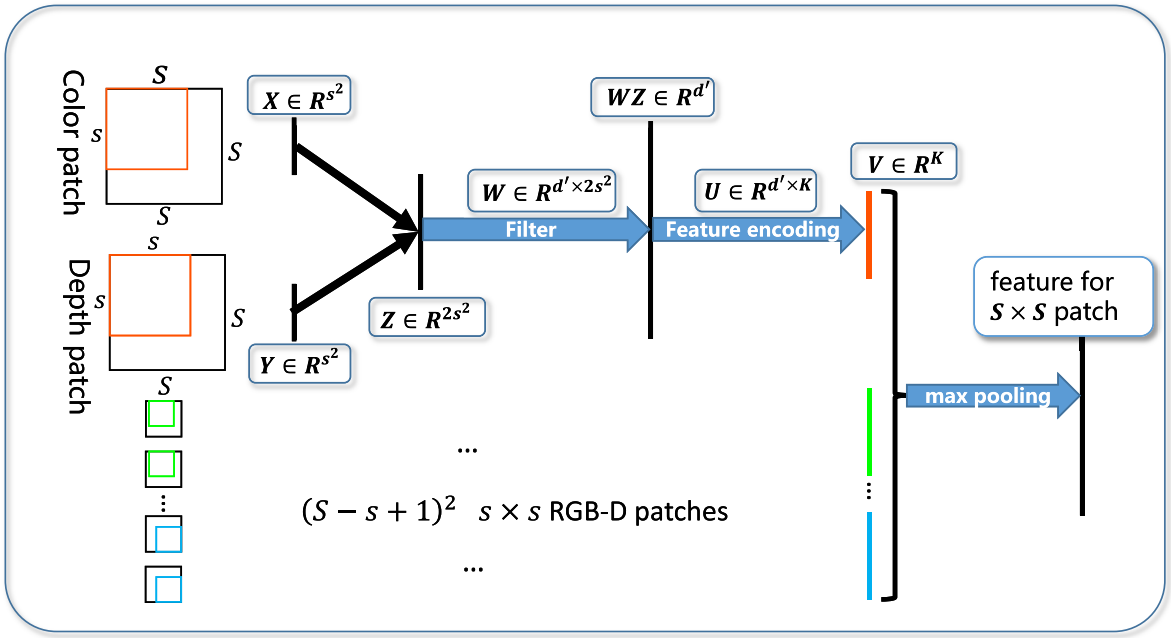


Fig. 2. Detailed illustration of feature extraction of the first layer: $s \times s$ color and depth patches are flatted to vectors X and Y. Z is the input vector obtained by concatenating X and Y. With learned filter matrix W and dictionary U, the sparse encoding coefficients V can be derived, which 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.

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.

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 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 [1], where we follow the adaption to RGB-D images proposed by Ren *et al.* [33] 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.

IV. Joint Deep Feature Learning and Encoding

By simultaneously performing feature learning and feature encoding, JFLE can obtain more efficient transformation

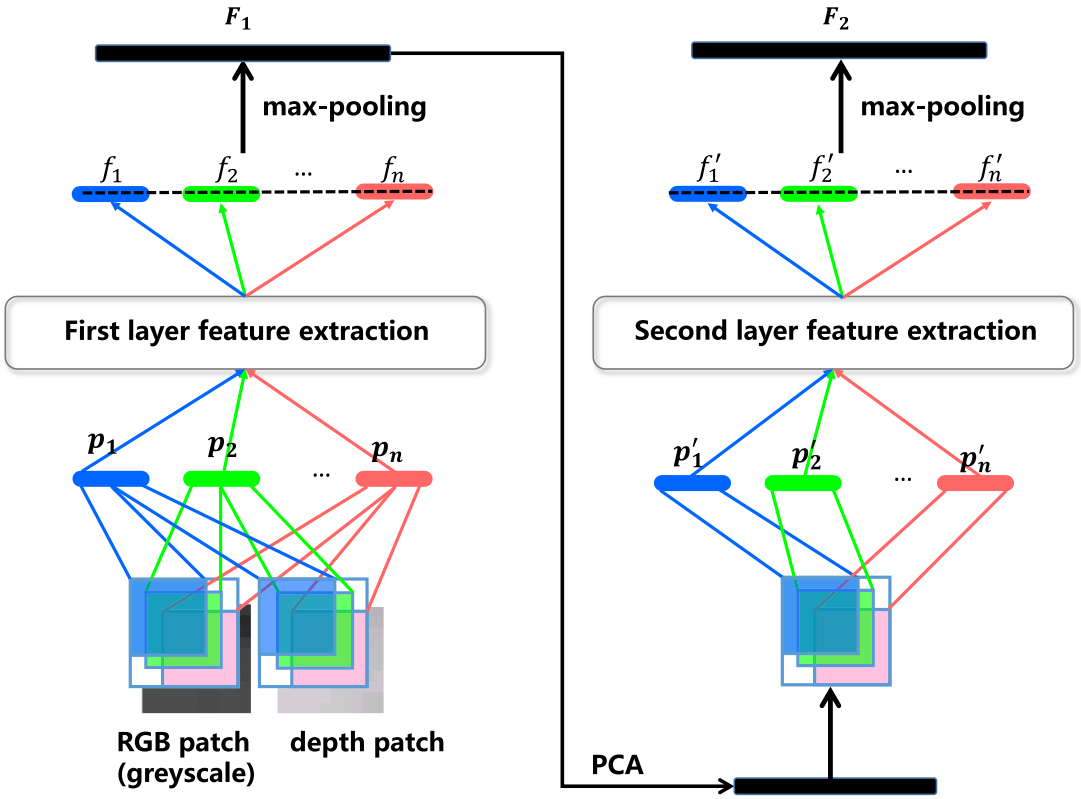


Fig. 3. Left: the unsupervised learning structure of the first layer. Right: the second layer structure. F_1 is the first-layer feature. F_2 is the second-layer feature.

matrix W and more descriptive dictionary U, compared with separate learning and encoding. Note that JFLE is under the assumption that the transformed data are vectors in an Euclidean space, meaning that each vector can be represented by the linear combination of a small number of vectors in dictionary. However, the original RGB-D data might need more complex description of the nonlinear manifold, as indoor objects usually have large variation in appearance. The linear assumption might be inappropriate for modeling them as it ignores the important intrinsic characteristic of data.

To address this issue, the kernel methods are usually employed. However, the design of the kernels is nontrivial and fixed kernels also make some assumptions on the data. In this paper, we take a further step to introduce stacked nonlinear layers to model a flexible nonlinear mapping. By doing so, we extend our framework of joint feature learning and feature encoding to a more general nonlinear model called *joint deep feature learning and encoding* (JDFLE), which incorporates intrinsic characteristic of the input data.

A. Formulation

Fig. 4 illustrates one single-layer structure of JDFLE. Instead of just using one projection matrix to model the mapping, we construct a neural network to learn features of objects. Raw RGB-D data are passed through multiple nonlinear transformation sub-layers. For example, passing the input data Z through one nonlinear sub-layer results in $A^{(1)}(Z) = t(W_1 Z + b_1)$, where W_1 denotes the projection matrix to be learned, b_1 denotes the bias, and t is a nonlinear activation function. In this way, data are modeled in a more complex manifold, the discriminative nature could be captured better, and the sparse coding process would not loss the important information. Consistent with the linear framework

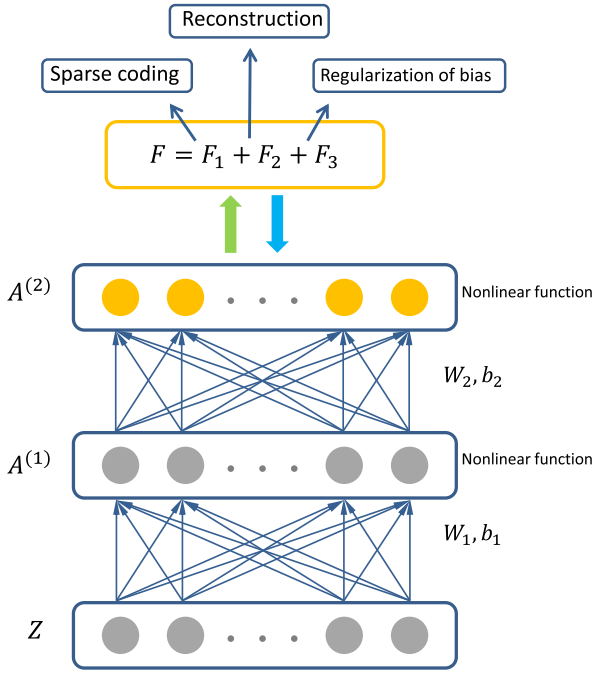


Fig. 4. Illustration of one single-layer structure in the proposed nonlinear framework. For given raw data of an RGB-D patch, we map it with several nonlinear sub-layers (here we have 2 sub-layers): each sub-layer contains linear projection matrix W, bias b and a nonlinear activation function. With the constraints of feature encoding (sparse coding), and the reconstruction constraints for each sub-layer (reconstruction and regularization of bias b), better dictionary and feature mapping could be learned.

of JFLE, the nonlinear framework of JDFLE also performs feature learning and feature encoding in a joint way, while the key difference is that the feature learning of JDFLE maps the original data to a nonlinear feature space which could be better sparse coded.

The notation of the output of one nonlinear sub-layer can be expressed as

$$A^{(m)}(Z) = t(W_m A^{(m-1)}(Z) + b_m) \tag{3}$$

with $A^{(0)}(Z) = Z$. With the nonlinear features, we perform feature learning and feature encoding jointly by adding feature encoding constraints on the top sub-layer output, and at the same time apply the reconstruction constraints and the bias regularization constraints on each sub-layer. Specifically, we minimize the following objective function:

$$\begin{aligned}\min_{W,b,V,U} \; & F = F_1 + F_2 + F_3 \\ &= \left(\left\| A^{(M)}(Z) - UV \right\|_2^2 + \xi_1 \|V\|_1 \right) \\ &\quad + \lambda \sum_{m=1}^{M} \left(\left\| W_m^T W_m A^{(m-1)}(Z) - A^{(m-1)}(Z) \right\|_2^2 \right. \\ &\qquad \left. + \xi_2 g\left(W_m A^{(m-1)}(Z) \right) \right) \\ &\quad + \xi_3 \sum_{m=1}^{M} \|b_m\|_2^2 \\ \text{subject to } & \|u_k\|_2 \le 1, \quad k = 1, 2, \ldots, K. \end{aligned} \tag{4}$$

where F_1 denotes the constraints of feature encoding (i.e. the sparse coding criteria for the nonlinear-mapped data), F_2 denotes the reconstruction constraints for each sub-layer (i.e. for sub-layer m, data can be reconstructed using the transposed projection matrix W_m^T), F_3 denotes the regularization constraints for the bias b_m for each sub-layer, M is the total number of sub-layers, and ξ_1, ξ_2, ξ_3 and λ are tradeoff parameters.

B. Optimization

To solve the optimization problem in (4), we iteratively optimize $\{W, b\}$, U and V. Particularly, when fixing W and b to optimize U or V, the operation is the same as that in JFLE (see Algorithm 1). When fixing U and V, we optimize W and b of each sub-layer by stochastic sub-gradient descent process. The gradients of the objective function F with respective to W_m, b_m for $m = 1, 2, \cdots, M$ can be derived as

$$\frac{\partial F}{\partial W_m} = \Phi_1^{(m)} A^{(m-1)^T} + \Phi_2^{(m)} \tag{5}$$

$$\frac{\partial F}{\partial b_m} = \Phi_1^{(m)} + \xi_3 b_m \tag{6}$$

with

$$\Phi_1^{(M)} = 2(A^{(M)} - UV) \otimes t'(H^{(M)}) \tag{7}$$

$$\Phi_2^{(M)} = C^{(M)} \tag{8}$$

$$\Phi_1^{(m)} = W_{m+1}^T \Phi_1^{(m+1)} \otimes t'(H^{(m)}) \tag{9}$$

$$\Phi_2^{(m)} = C^{(m)} + \left(\sum_{k=m+1}^{M} R^{(k)'} A^{(m)} \right) \otimes t'(H^{(m)}) A^{(m-1)^T} \tag{10}$$

where $\otimes$ denotes the element-wise multiplication operation, and $C^{(m)}$, $R^{(m)}$ and $H^{(m)}$ are defined as

$$\begin{aligned} C^{(m)} &= 2W_m A^{(m-1)} (W_m^T W_m A^{(m-1)} - A^{(m-1)})^T \\ &\quad + 2W_m (W_m^T W_m A^{(m-1)} - A^{(m-1)}) A^{(m-1)^T} \\ &\quad + \xi_2 g'(W_m A^{(m-1)}) A^{(m-1)^T} \end{aligned} \tag{11}$$

$$\begin{aligned} R^{(m)} &= \left\| W_m^T W_m A^{(m-1)}(Z) - A^{(m-1)}(Z) \right\|_2^2 \\ &\quad + \xi_2 g(W_m A^{(m-1)}(Z)) \end{aligned} \tag{12}$$

$$H^{(m)} = W_m A^{(m-1)}(Z) + b_m. \tag{13}$$

With the gradients, we iteratively update W_m and b_m until convergence:

$$\mathrm{W}_m = \mathrm{W}_m - \mu \frac{\partial F}{\partial W_m} \tag{14}$$

$$\mathrm{b}_m = \mathrm{b}_m - \mu \frac{\partial F}{\partial b_m} \tag{15}$$

We use $tanh$ as the activation function t:

$$t(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{16}$$

$$t'(x) = \tanh'(x) = 1 - \tanh^2(x). \tag{17}$$

Same as that in JFLE, $g(x)$ is defined as

$$g(x) = \sqrt{\xi^2 + x^2} \tag{18}$$

where ξ is a parameter.

What we present above is just one single-layer nonlinear structure (with multiple sub-layers) of the JDFLE. We further stack the single-layer nonlinear structure to have a hierarchical structure to capture multi-level information in RGB-D data, similar to that for JFLE described in section III-C.

V. Experiments

A. Dataset

We use the benchmark dataset - the NYU depth dataset [36], [37], including version 1 and version 2, 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.

B. Implementation Details

For each stacked layer, we randomly sample 20000 7×7 patches as training data. We run 50 iterations to learn the feature mapping and dictionary U in our unsupervised learning frameworks for both JFLE and JDFLE.

In the JFLE framework, the size of W is 200×98. The parameters λ_1, λ_2 and λ_3 in Eq. (2) are empirically set to 0.1, 0.5 and 0.15. Each iteration takes about 17 minutes on

TABLE I
CLASS-AVERAGE ACCURACY COMPARISON OF DIFFERENT METHODS ON THE NYU DEPTH DATASET V1

Single	JFLE	61.71%
	gradient KDES [33]	51.84%
	color KDES [33]	53.27%
	spin/surface normal KDES [33]	40.28%
	depth gradient KDES [33]	53.56%
Combined	Silberman and Fergus [36]	53%
	Pei *et al.* [31]	50.50%
	Ren *et al.* [33]	71.40%
	Combining ours with Ren's	72.94%

TABLE II
CLASS-AVERAGE ACCURACY RESULTS OF OUR METHOD WITH DIFFERENT SETTINGS ON THE NYU DEPTH DATASET V1

JFLE (first layer)	54.76%
JFLE (second layer)	52.90%
Sparse coding after feature learning	45.67%
k-means encoding after feature learning	22.32%
Separate learning from two modalities	50.74%

average on a PC with Intel i5 3.10GHz CPU and 8G memory. ξ is set as 10^{-5} for both JFLE and JDFLE.

For JDFLE framework, we pass raw data through 2 nonlinear mapping sub-layers ($M = 2$) for one single layer. The sizes of W_1 and W_2 are set to 98×150 and 150×200 respectively. Two such single-layer structures are stacked as shown in Fig. 3, similar to the JFLE framework. The parameters λ, ξ_1, ξ_2, ξ_3 in Eq. (4) are set to 2, 0.03, 0.1, 0.1. Each iteration takes about 21 minutes on average.

For a superpixel, we calculate the mean values of all its 10×10 patches' features. With the labeled superpixels in the training list, we train a 1-vs-all linear SVM for each category. For NYU depth dataset V1 [36], we use 60% data for training and 40% data for testing which is the same as that of [33]. For NYU depth dataset V2 [37], 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 elements represent the pixel-level labeling accuracy for 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, except the method of Khan *et al.* [17], as the higher-order graphical model is one of their major contributions.

C. Results

1) Comparisons on Dataset V1: Table I shows the average labeling results of different methods on the NYU detph dataset V1. We compare the result of our two-layer JFLE method with: 1) the result of Silberman and Fergus [36]; 2) the result of Pei *et al.* [31]; 3) the result of single kernel descriptor (KDES) [33]; 4) the result of Ren *et al.* [33] (combining four KDESs and geometry features); 5) the result of combining the features of our JFLE method and Ren's.

It can be seen from Table I 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.* [31], as they use selected patches which are usually redundant and noisy in encoding. However, our result does not outperform that of Ren *et al.* [33]. We argue that Ren *et al.* [33] evaluated six kernel based features, integrated four of them: gradient, color, depth gradient, spin/surface normal. For six kernel descriptors, each of them has several hyperparameters to be tuned. The authors also conducted sophisticated evaluation process to select which of them to be used for classification and finally chose four of them. The weights between these descriptors also need to be tuned to obtain the best combination. 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 [33], our method achieves superior performance. Compared to [33], 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 significant complementary visual patterns which cannot be captured by those of [33].

a) Evaluating different feature learning and encoding settings: Here, we give detailed evaluations on our linear method with only one-layer feature learning and feature encoding structure under different settings. In particular, we compare the following five setups: 1) our JFLE method with the features learned from the first layer; 2) our JFLE method with the features learned from the second layer alone; 3) separate learning: conducting feature learning with (1) to get filter matrix W and then performing sparse coding to encode filtered data WZ; 4) conducting feature learning with (1) and then use k-means clustering result as hard quantization to encode filtered data WZ; 5) learning features from two modalities separately with our JFLE cost function. Table II shows the results of the five different setups.

Comparing the results of methods 1, 3 and 4 in Table II, we can see that joint feature learning and encoding performs much better than the methods using separate processing. Particularly, for method 3, 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 3 cannot guide W to help find descriptive U. For method 4, important feature information is lost when quantized by k-means.

Comparing the results of methods 1 and 5 in Table II, we can see that joint learning from two modalities outperforms separate learning. This is because separate learning ignores the complimentary information between the two modalities, for which the extra features learned from depth alone might

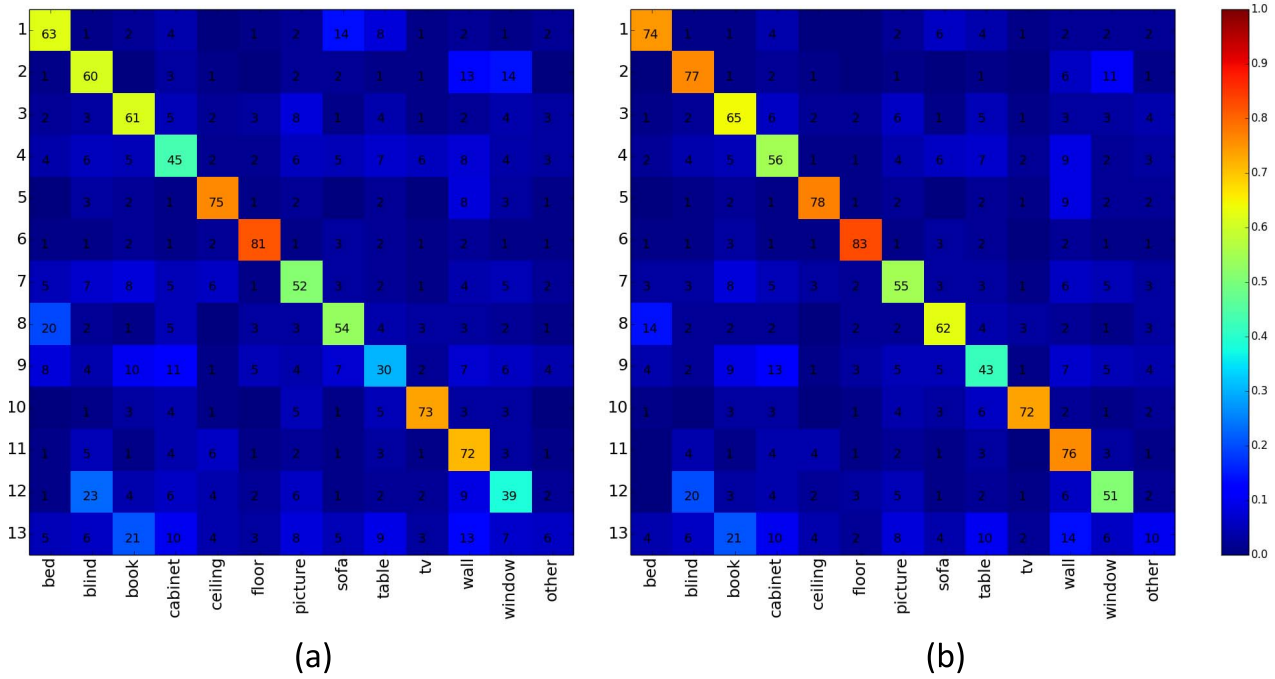


Fig. 5. The confusion matrices of: (a) our results with one-layer JFLE structure; (b) our results with two-layer JFLE structure.

TABLE III
INDIVIDUAL CLASS LABEL ACCURACY ON THE NYU DEPTH DATASET V1 WITH ONLY ONE-LAYER FEATURES. 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%

hurt the performance, as shown in Table III. 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 III).

Comparing the results of methods 1 and 2 in Table II, 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 I), compared with using the first-layer features alone. The detailed comparison of confusion matrixes between one-layer JFLE and two-layer JFLE is shown in Fig. 5.

TABLE IV
LABELING ACCURACY ON THE NYU DEPTH DATASET V1 FOR FIRST-LAYER, SECOND-LAYER, BOTH-LAYER FEATURE LEARNING AND FEATURE ENCODING STRUCTURE. SECOND COLUMN: THE RESULTS OF OUR JFLE FRAMEWORK. THIRD COLUMN: THE RESULTS OF OUR JDFLE FRAMEWORK

	JFLE	JDFLE
First layer	54.76%	59.46%
Second layer	52.90%	57.18%
Two layers	61.71%	63.28%
Combined with Ren's	72.94%	73.68%

b) Efficacy of JDFLE: To illustrate the efficacy of the JDFLE framework, the comparison with results of the JFLE framework on NYU dataset V1 is shown in Table IV. We can see that the JDFLE framework improves the performance of first-layer, second-layer, both-layer structures compared to JFLE. This is because we learn more general model for data through several nonlinear projections, and the simultaneously learned dictionary becomes more efficient to encode the data. In Fig. 6, several visual examples are shown to illustrate the improvement made by JDFLE framework.

2) Comparisons on Dataset V2: We also compare our results on NYU depth dataset V2 with the following existing works that have reported results on the dataset V2: 1) Couprie *et al.* [4], which automatically learns features from raw data input, similar to our method;

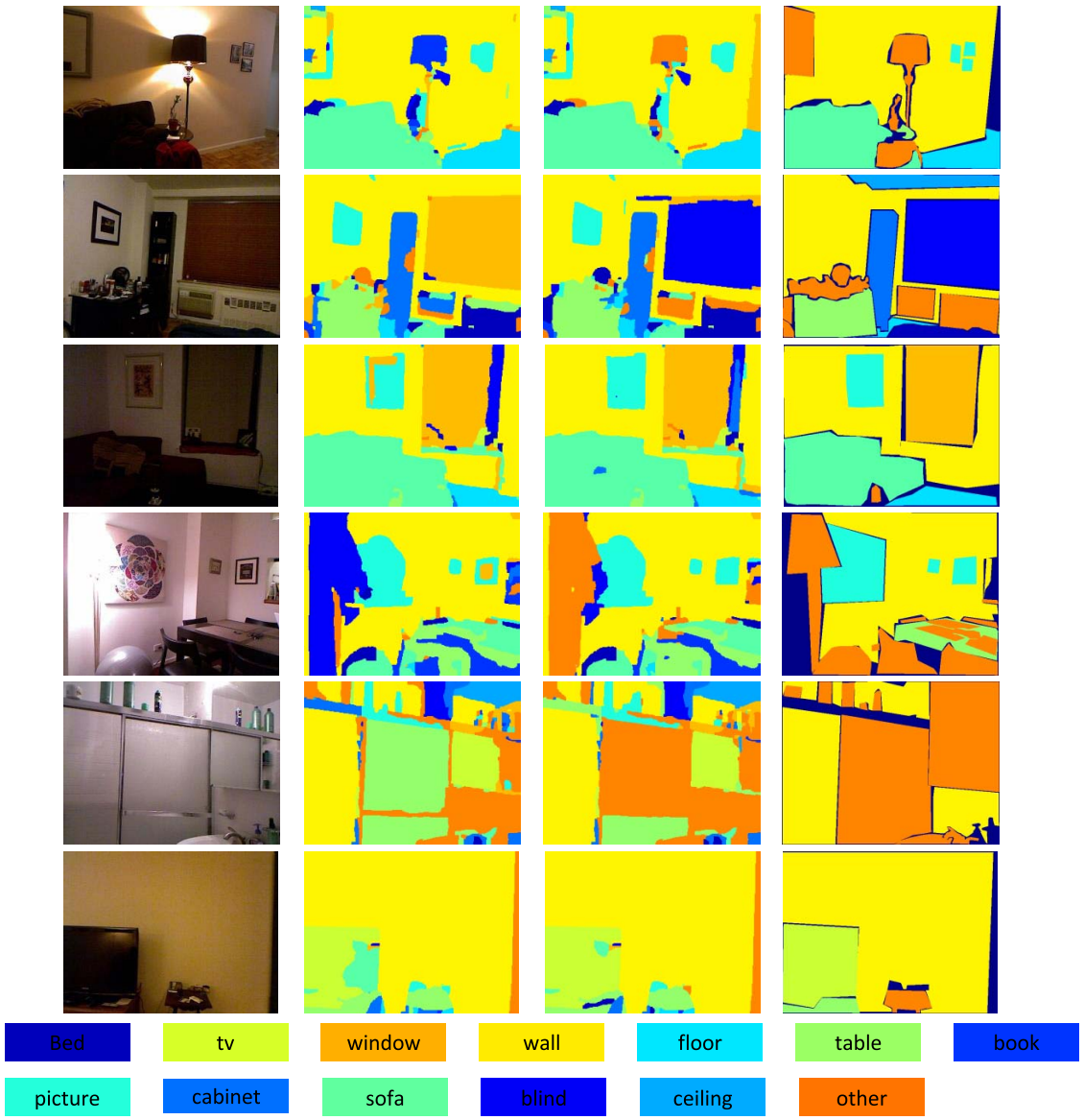


Fig. 6. Visual comparisons of our JFLE and JDFLE results. From left to right: original color images, the results of combining our JFLE features and Ren's features, the results of combining our JDFLE features and Ren's features, ground truth.

TABLE V
INDIVIDUAL CLASS LABELING PERFORMANCE OF 4-CLASS SETTING ON THE NYU DEPTH DATASET V2

	Ground	Furniture	Props	Structure
Couprie *et al.* [4]	87.3	45.3	35.5	86.1
Cadena and Kosecka [2]	87.3	60.6	33.7	74.8
Khan *et al.* [17]	87.1	54.7	32.6	88.2
JFLE	90.1	46.3	43.3	81.4
JDFLE	87.3	50.7	47.4	82.2

2) Cadena and Košecka [2]; 3) Khan *et al.* [17]. Cadena and Košecka [2] and Khan et al. [17] both include a lot of hand-crafted appearance and geometry features. Table V, VI, VII show the individual class labeling accuracy results on the NYU depth dataset V2 with the same 4-class, 13-class, 40-class settings. For 40-class setting, we also show the Jaccard index of each class. The overall performance summaries are shown in Table VIII and IX including: overall pixel-level accuracy (Pix. Acc.) and average pixel-level labeling accuracy for all classes (Per-Class Acc.) for 4-class, 13-class, 40-class settings, the mean pixel-frequency weighted Jaccard index (Freq. Jaccard), and the flat mean Jaccard index (Av. Jaccard) for the 40-class setting. It can be seen that our JFLE and JDFLE methods in general achieve the competitive accuracy compared to [4], [2], and [17]. We would like to point out that a very recent work [6] by Eigen and Fergus reported a better performance on NYU depth dataset V2, e.g. its pixel-level accuracy for 40-class setting has reached 62.9%. However, their method is based on CNN with supervised information required, while we use unsupervised feature learning, i.e. deriving features without using any label information. A direct comparison with their method is unfair as the problem settings are different. Compared with JFLE, JDFLE also achieves improved performance on NYU depth dataset V2.

TABLE VI
INDIVIDUAL CLASS LABELING PERFORMANCE OF 13-CLASS SETTING ON THE NYU DEPTH DATASET V2

	bed	objects	chair	furnit.	ceiling	floor	deco.	sofa	table	wall	window	books	tv
Couprie *et al.* [4]	38.1	8.7	34.1	42.4	62.6	87.3	40.4	24.6	10.2	86.1	15.9	13.7	6
Khan *et al.* [17]	32.3	-	-	-	64.7	75.8	-	58.6	47.9	77.5	54	38.3	45.7
JFLE	47.6	12.4	23.5	16.7	68.1	84.1	26.4	39.1	35.4	65.9	52.2	45	32.4
JDFLE	48.5	19.5	26.2	24.8	63.2	89.5	27.3	37	36.3	76.4	49.3	43.2	33.1

TABLE VII
INDIVIDUAL CLASS LABELING PERFORMANCE OF 40-CLASS SETTING ON THE NYU DEPTH DATASET V2

JACCARD INDEX	wall	floor	cabinet	bed	chair	sofa	table	door	window	bookshelf	picture	counter	blinds	desk	shelves	curtain	dresser	pillow	mirror	floor mat
JFLE	60.1	64.1	27.5	46.1	29	29.4	20.2	6.5	23.6	13.3	24.8	31.2	31	4.3	1.1	16	15.1	11.8	9	14.7
JDFLE	61.4	66.4	38.2	43.9	34.4	33.8	22.6	8.3	27.6	17.6	27.7	30.2	33.6	5.1	2.7	18.9	16.8	12.5	10.7	13.8
	clothes	ceiling	books	fridge	television	paper	towel	shower curtain	box	whitebd	person	night stand	toilet	sink	lamp	bathtub	bag	other struct	other furnit	other prop
JFLE	2.7	46.1	3.6	2.9	3.2	2.6	6.2	6.1	0.8	28.2	5	6.9	32	20.9	5.4	16.2	0.2	4.8	1.1	21.7
JDFLE	3.3	47.9	2.5	4.3	5.8	4	6.4	1	1.3	19.2	6	8.9	33.3	21.7	6.7	26.5	0.7	5.6	2.7	24.6
PIXEL ACCURACY	wall	floor	cabinet	bed	chair	sofa	table	door	window	bookshelf	picture	counter	blinds	desk	shelves	curtain	dresser	pillow	mirror	floor mat
JFLE	76.7	87.1	35.3	61	41.7	44.9	29.9	10.3	43.9	24.6	41.7	51.4	47.4	8.4	2.7	27	22.1	24.6	19.5	25.8
JDFLE	77.4	89.1	48.7	57.2	48.7	50.9	32.8	12.8	49.9	31.8	45.6	48.8	50.2	9.7	6.6	31	24.1	25.5	22.6	23.5
	clothes	ceiling	books	fridge	television	paper	towel	shower curtain	box	whitebd	person	night stand	toilet	sink	lamp	bathtub	bag	other struct	other furnit	other prop
JFLE	8.2	69.6	9.2	6.3	5.1	7.6	15.1	7.5	3.6	32.4	14.2	11.9	56.4	32.6	20.4	18.2	0.4	8.1	1.8	25.4
JDFLE	9.6	71	6.3	8.9	8.9	11.4	14.9	1.3	5.7	21.9	16.2	15	56.9	33.4	24.5	29.6	2.1	9.2	4.3	28.6

TABLE VIII
LABELING PERFORMANCE OF 4-CLASS AND 13-CLASS SETTINGS ON THE NYU DEPTH DATASET V2

4-Class	Pix. Acc.	Per-Class Acc.
Couprie *et al.* [4]	64.5	63.5
Cadena and Kosecka [2]	64.9	64.1
Khan *et al.* [17]	69.2	65.6
JFLE	67.8	65.3
JDFLE	69.7	66.9
13-Class	Pix. Acc.	Per-Class Acc.
Couprie *et al.* [4]	52.4	36.2
Khan *et al.* [17]	†	
JFLE	47.4	42.2
JDFLE	52.8	44.1

TABLE IX
LABELING PERFORMANCE OF 40-CLASS ON THE NYU DEPTH DATASET V2

40-Class	Pix. Acc.	Per-Class Acc.	Freq. Jaccard	Av. Jaccard
JFLE	48.9	27	35.8	17.4
JDFLE	51.6	29.2	38.1	19

TABLE X
LABELING ACCURACY ON THE NYU DEPTH DATASET V2 FOR FIRST-LAYER, SECOND-LAYER, BOTH-LAYER FEATURE LEARNING AND FEATURE ENCODING STRUCTURE USING OUR PROPOSED JFLE AND JDFLE FRAMEWORKS

	JFLE	JDFLE
First layer	60.2%	62.4%
Second layer	58.7%	61.3%
Two layers	65.3%	66.9%

Table X shows the comparison between 4-class average class accuracy performance of JFLE and JDFLE with first-layer, second-layer, both-layer structures.

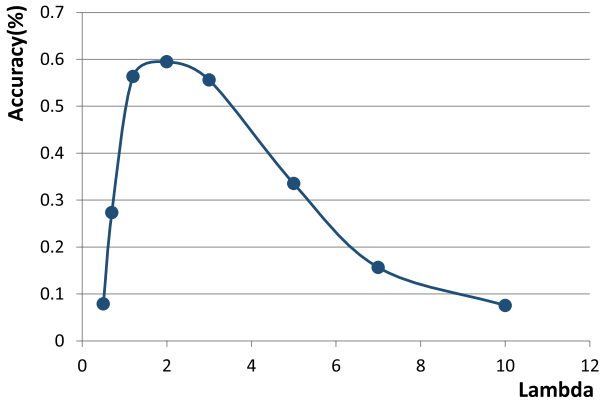


Fig. 7. Scene labeling accuracy under different λ values with single-layer JDFLE structure on NYU depth dataset V1.

TABLE XI
THE RESULTS OF DIFFERENT SETTINGS OF JDFLE

	Accuracy
Two nonlinear sub-layers	59.46%
One nonlinear sub-layers	58.21%

Fig. 8 and Fig. 9 show some examples of pixel labeling results on NYU depth dataset V1 and V2 respectively. 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.

3) Parameter Analysis:

a) Different settings of JDFLE: Different nonlinear structures could be constructed to learn the nonlinear mapping of the input RGB-D data. In Table IV, we show the result with two nonlinear sub-layers (i.e. $M = 2$) to map the data. We also

†Khan *et al.* [17] used a different label set. See the details in Table VI.

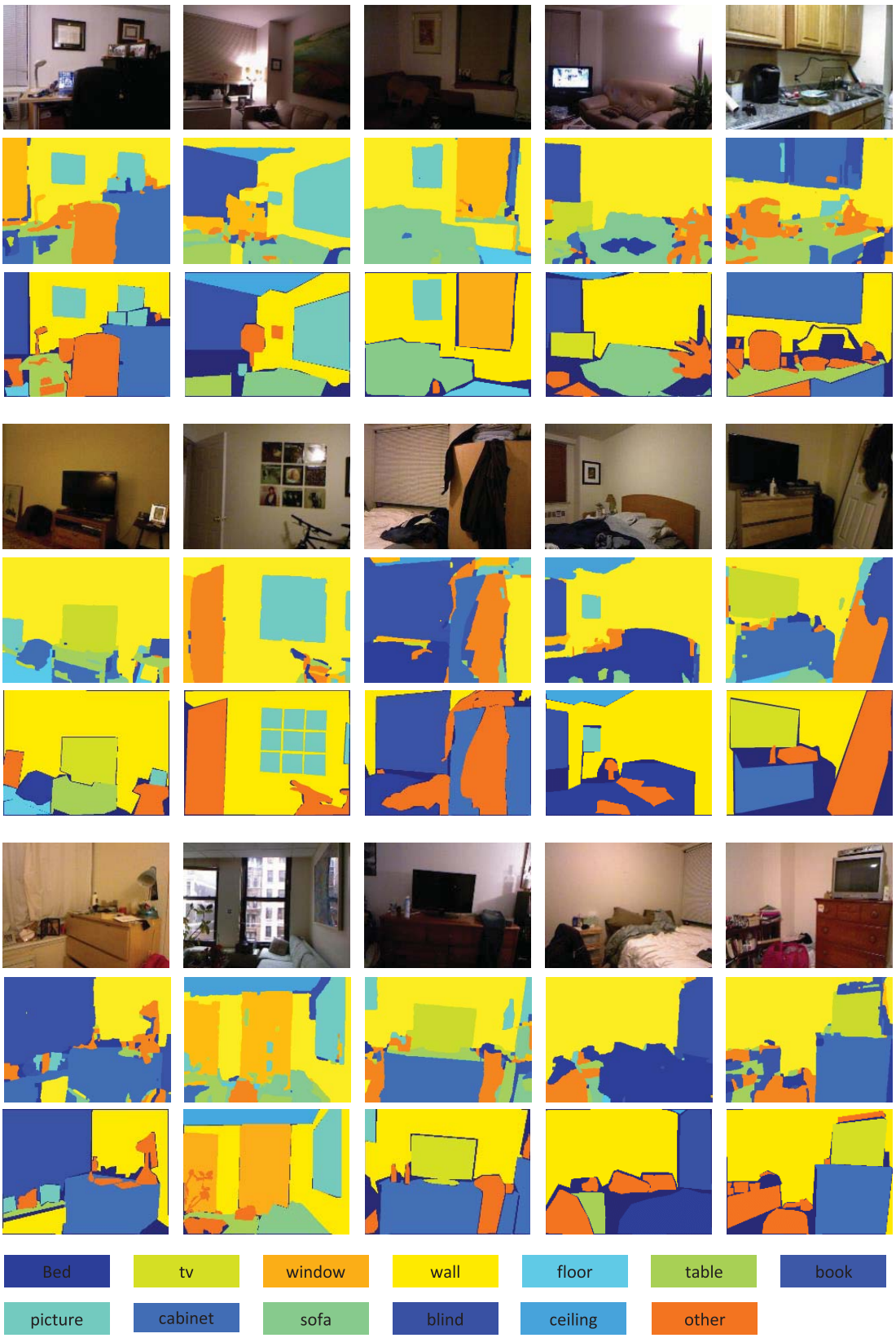


Fig. 8. 15 example results on NYU depth dataset V1. Rows 1st, 4th and 7th: color images. Rows 2nd, 5th and 8th: the results of combining our JDFLE features and Ren's features. Rows 3rd, 6th and 9th: ground truth. Note that since we focus on feature learning, we did not use CRFs to smooth the labels. So the results might look a bit noisy.

test more shallow network with $M = 1$. The results are shown in Table XI. It can be seen that with only one nonlinear sub-layer, the result is getting worse.

b) Effect of different λ: Fig. 7 shows the accuracy results under different λ in Eq. (4) with single-layer JDFLE structure on NYU depth dataset V1. Recall that λ denotes

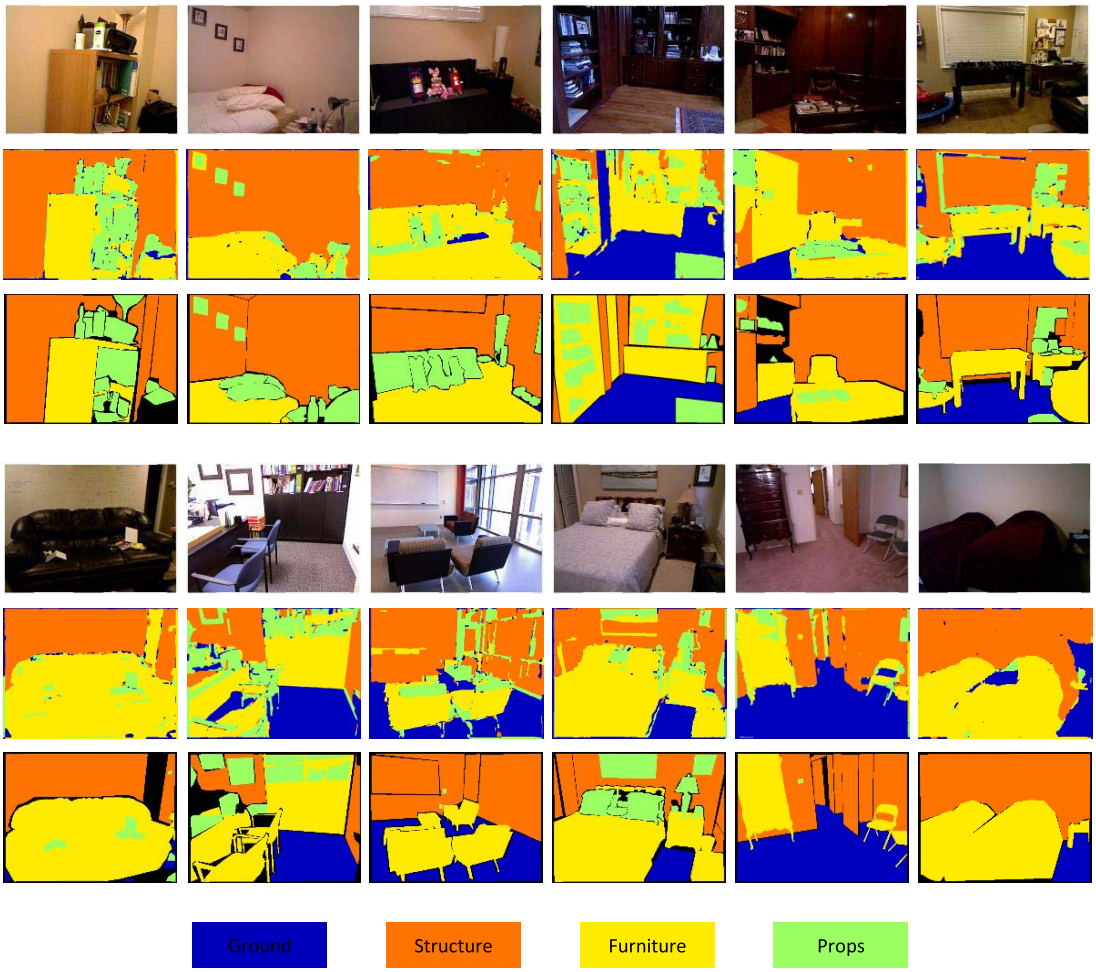


Fig. 9. 12 example results on NYU depth dataset V2. Rows 1st and 4th: color images. Rows 2nd and 5th: the results of our JDFLE features. Rows 3rd and 6th: ground truth.

the tradeoff between feature learning and feature encoding. If λ is too small, feature encoding becomes dominate while weak feature learning will not be able to learn discriminative features. On the other hand, if λ is too big, feature learning becomes dominate while weak feature encoding will not be able to produce effective sparse representation. Fig. 7 clearly shows such a tradeoff, which suggests an optimal λ value around 2.

VI. Conclusion

In this paper, we have presented an unsupervised feature learning framework that learns features from RGB-D data for scene labeling task. Our method considers unsupervised feature learning and feature encoding problems together and implicitly infers the relationship between two modalities. Furthermore, we have also extended the framework with stacked nonlinear projections to make it more general. 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 obtained comparable results with state-of-the-art methods including those using hand-crafted features and those learning features from raw data.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[2] C. Cadena and J. Košecka, "Semantic parsing for priming object detection in RGB-D scenes," in *Proc. 3rd Workshop Semantic Perception, Mapping Exploration*, 2013, pp. 1–6.

[3] A. Coates, A. Y. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2011, pp. 215–223.

[4] C. Couprie, C. Farabet, L. Najman, and Y. LeCun. (2013). "Indoor semantic segmentation using depth information." [Online]. Available: http://arxiv.org/abs/1301.3572

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. CVPR*, Jun. 2005, pp. 886–893.

[6] D. Eigen and R. Fergus. (2014). "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture." [Online]. Available: http://arxiv.org/abs/1411.4734

[7] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.

[8] C. Galleguillos, A. Rabinovich, and S. Belongie, "Object categorization using co-occurrence, location and appearance," in *Proc. IEEE Conf. CVPR*, Jun. 2008, pp. 1–8.

[9] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions," in *Proc. IEEE 12th ICCV*, Sep./Oct. 2009, pp. 1–8.

[10] D. Grangier, L. Bottou, and R. Collobert, "Deep convolutional networks for scene parsing," in *Proc. ICML Deep Learn. Workshop*, 2009, pp. 1–2.

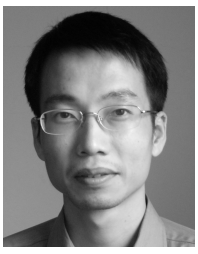
[11] S. Gupta, P. Arbelaez, and J. Malik, "Perceptual organization and recognition of indoor scenes from RGB-D images," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 564–571.
[12] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from RGB-D images for object detection and segmentation," in *Proc. 13th ECCV*, 2014, pp. 345–360.
[13] X. He, R. S. Zemel, and M. Á. Carreira-Perpiñán, "Multiscale conditional random fields for image labeling," in *Proc. IEEE Comput. Soc. Conf. CVPR*, Jun/Jul. 2004, pp. II-695–II-702.
[14] A. Hermans, G. Floros, and B. Leibe, "Dense 3D semantic mapping of indoor scenes from RGB-D images," in *Proc. IEEE ICRA*, May/Jun. 2014, pp. 2631–2638.
[15] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.
[16] I.-H. Jhuo, S. Gao, L. Zhuang, D. Lee, and Y. Ma, "Unsupervised feature learning for RGB-D image classification," in *Proc. 12th ACCV*, Nov. 2014, pp. 276–289.
[17] S. H. Khan, M. Bennamoun, F. Sohel, and R. Togneri, "Geometry driven semantic labeling of indoor scenes," in *Proc. 13th ECCV*, 2014, pp. 679–694.
[18] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic labeling of 3D point clouds for indoor scenes," in *Proc. Adv. NIPS*, 2011, pp. 244–252.
[19] A. Kumar, P. Rai, and H. Daumé, III, "Co-regularized multi-view spectral clustering," in *Proc. Adv. NIPS*, 2011, pp. 1413–1421.
[20] K. Lai, L. Bo, and D. Fox, "Unsupervised feature learning for 3D scene labeling," in *Proc. IEEE ICRA*, May/Jun. 2014, pp. 3050–3057.
[21] Q. V. Le, A. Karpenko, J. Ngiam, and A. Y. Ng, "ICA with reconstruction cost for efficient overcomplete feature learning," in *Proc. Adv. NIPS*, 2011, pp. 1017–1025.
[22] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Proc. IEEE Conf. CVPR*, Jun. 2011, pp. 3361–3368.
[23] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Proc. Adv. NIPS*, 2006, pp. 801–808.
[24] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. 26th Annu. ICML*, 2009, pp. 609–616.
[25] V. Lempitsky, A. Vedaldi, and A. Zisserman, "Pylon model for semantic segmentation," in *Proc. Adv. NIPS*, 2011, pp. 1485–1493.
[26] C. Liu, J. Yuen, and A. Torralba, "SIFT flow: Dense correspondence across scenes and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, May 2011.
[27] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE ICCV*, Sep. 1999, pp. 1150–1157.
[28] B. Micusik and J. Kosecka, "Semantic segmentation of street scenes by superpixel co-occurrence and 3D geometry," in *Proc. IEEE 12th ICCV Workshops*, Sep./Oct. 2009, pp. 625–632.
[29] A. C. Müller and S. Behnke, "Learning depth-sensitive conditional random fields for semantic segmentation of RGB-D images," in *Proc. IEEE ICRA*, May/Jun. 2014, pp. 6232–6237.
[30] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. 28th ICML*, 2011, pp. 689–696.
[31] D. Pei, H. Liu, Y. Liu, and F. Sun, "Unsupervised multimodal feature learning for semantic image segmentation," in *Proc. IJCNN*, Aug. 2013, pp. 1–6.
[32] G. Potamianos, C. Neti, J. Luettin, and I. Matthews, "Audio-visual automatic speech recognition: An overview," in *Issues in Visual and Audio-Visual Speech Processing*, vol. 22. Cambridge, MA, USA: MIT Press, 2004, p. 23.
[33] X. Ren, L. Bo, and D. Fox, "RGB-(D) scene labeling: Features and algorithms," in *Proc. IEEE Conf. CVPR*, Jun. 2012, pp. 2759–2766.
[34] M. Schimidt. (2005). *minFunc*. [Online]. Available: http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html
[35] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "*TextonBoost*: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," in *Proc. 9th ECCV*, 2006, pp. 1–15.
[36] N. Silberman and R. Fergus, "Indoor scene segmentation using a structured light sensor," in *Proc. IEEE ICCV Workshops*, Nov. 2011, pp. 601–608.
[37] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. 12th ECCV*, 2012, pp. 746–760.
[38] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Y. Ng, "Convolutional-recursive deep learning for 3D object classification," in *Proc. Adv. NIPS*, 2012, pp. 665–673.
[39] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng, "Parsing natural scenes and natural language with recursive neural networks," in *Proc. 28th ICML*, 2011, pp. 129–136.
[40] A. Wang, J. Lu, G. Wang, J. Cai, and T.-J. Cham, "Multi-modal unsupervised feature learning for RGB-D scene labeling," in *Proc. 13th ECCV*, 2014, pp. 453–467.
[41] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proc. BMVC*, 2009, pp. 124.1–124.11.
[42] Y. Xie, J. Ho, and B. Vemuri, "On a nonlinear generalization of sparse coding and dictionary learning," in *Proc. ICML*, 2013, pp. 1480–1488.
[43] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. CVPR*, Jun. 2009, pp. 1794–1801.

Anran Wang received the B.S. degree from Tianjin University, China, in 2012. She is currently pursuing the Ph.D. degree with the School of Computer Engineering, Nanyang Technological University, Singapore. Her research interests are computer vision and machine learning.

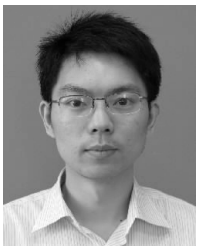
Jiwen Lu received the B.Eng. degree in mechanical engineering and the M.Eng. degree in electrical engineering from the Xi'an University of Technology, Xi'an, China, in 2003 and 2006, respectively, and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore, in 2011.

He is currently an Associate Professor with the Department of Automation, Tsinghua University, China. He has authored or co-authored over 110 scientific papers in these areas, in which more than 40 papers are published in the IEEE TRANSACTIONS journals and top-tier computer vision conferences. His current research interests include computer vision, pattern recognition, and machine learning. He serves as an Associate Editor of *Pattern Recognition Letters* and *Neurocomputing*.

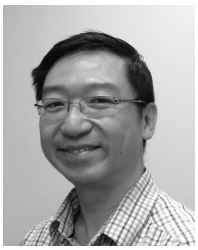
Dr. Lu was a recipient of the First-Prize National Scholarship and the National Outstanding Student Award from the Ministry of Education of China in 2002 and 2003, the Best Student Paper Award from the Pattern Recognition and Machine Intelligence Association of Singapore in 2012, the Top 10% Best Paper Award from the IEEE International Workshop on Multimedia Signal Processing in 2014, and the National 1000 Young Talents Plan Program in 2015.



Jianfei Cai received the Ph.D. degree from the University of Missouri–Columbia. He is currently an Associate Professor and has served as the Head of the Visual and Interactive Computing Division and the Head of the Computer Communication Division with the School of Computer Engineering, Nanyang Technological University, Singapore. He has authored over 150 technical papers in international journals and conferences. His major research interests include computer vision, visual computing, and multimedia networking. He has been actively participating in program committees of various conferences. He served as the leading Technical Program Chair of the IEEE International Conference on Multimedia and Expo in 2012 and the leading General Chair of the Pacific-Rim Conference on Multimedia in 2012. Since 2013, he has served as an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING. He served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from 2006 to 2013.



Gang Wang received the B.S. degree from the Harbin Institute of Technology in Electrical Engineering, in 2005, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign (UIUC), in 2010. He is currently an Assistant Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University. His research interests include computer vision and machine learning. In particular, he is focusing on object recognition, scene analysis, and deep learning. During his Ph.D. study, he was a recipient of the Prestigious Harriett & Robert Perry Fellowship (2009–2010) and the CS/AI Award (2009) at UIUC. He is an Associate Editor of *Neurocomputing*.



Tat-Jen Cham received the B.A. degree in engineering and the Ph.D. degree from the University of Cambridge, in 1993 and 1996, respectively. He received the Jesus College Research Fellowship in Cambridge from 1996 to 1997, and was a Research Scientist with the DEC/Compaq Research Laboratory, Boston, from 1998 to 2001. While with Nanyang Technological University (NTU), he was concurrently a Singapore-MIT Alliance Fellow from 2003 to 2006, the Director of the Centre of Multimedia and Network Technology from 2007 to 2015, and an NTU Senator from 2010 to 2014. He is currently an Associate Professor with the School of Computer Engineering, NTU, and a Principal Investigator with the NRF BeingThere Centre for 3D Telepresence. His research interests are broadly in computer vision and machine learning. He is on the Editorial Board of the *International Journal of Computer Vision*, and was the General Co-Chair of the Asian Conference on Computer Vision in 2014.