RGB-D Scene Labeling with Multimodal Recurrent Neural Networks

Heng Fan[1], Xue Mei[2], Danil Prokhorov[2] and Haibin Ling[1]
[1]Department of Computer and Information Sciences, Temple University, Philadelphia, USA
[2]Toyota Research Institute, North America, Ann Arbor, Michigan, USA
{hengfan, hbling}@temple.edu, xue.mei@toyota.com

Abstract

Recurrent neural networks (RNNs) are able to capture context in an image by modeling long-range semantic dependencies among image units. However, existing methods only utilize RNNs to model dependencies of a single modality (e.g., RGB) for labeling. In this work we extend this single-modal RNNs to multimodal RNNs (MM-RNNs) and apply it to RGB-D scene labeling. Our MM-RNNs are capable of seamlessly modeling dependencies of both RGB and depth modalities, and allow memory sharing across modalities. By sharing memory, each modality possesses multiple properties of itself and other modalities, and becomes more discriminative to distinguish pixels. Moreover, we also analyse two simple extensions of single-modal RNNs and demonstrate that our MM-RNNs perform better than both of them. Integrating with convolutional neural networks (CNNs), we build an end-to-end network for RGB-D scene labeling. Extensive experiments on NYU depth V1 and V2 demonstrate the effectiveness of MM-RNNs.

1. Introduction

As one of the most challenging problems in computer vision, image labeling, which aims to assign a pre-defined semantic label to each pixel in an image, is a key step to understand image. To this end, numerous researches have been done on scene labeling. Roughly speaking, previous approaches can be categorized into two types according to their target scenes: indoor scene labeling and outdoor scene labeling [15].

In contrast to outdoor scene labeling [1, 7, 18, 19, 21, 27, 33], indoor scene labeling is challenged by a large set of semantic labels and large object appearance variation caused by occlusion, deformation, scale changes, etc. [15]. Recently, with the help of low cost depth sensors, a new rich source of information, i.e., depth information, has become easily available to boost the performance of indoor scene labeling by providing structural information to some extend. Based on that, a large body of RGB-D scene labeling methods have been proposed [11, 12, 15, 24, 25, 30, 36, 38]. Nevertheless, there exist issues in two aspects.

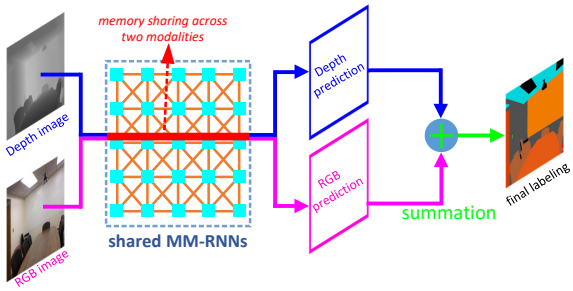

Figure 1. Illustration of the proposed MM-RNNs. Our MM-RNNs take two sources of information as inputs (i.e., RGB and depth), and allow 'memory' sharing between them. As a consequence, the depth information has the memory of RGB modality and vice versa. With memory sharing, each modality becomes more discriminative for pixel classification, and the final labeling is derived by summarizing over the outputs of multiple modalities.

- One problem is how to capture long-range contextual information in an image. Current RGB-D scene labeling approaches only exploit short-range contexts of pixels or superpixels, which easily results in misclassification between visually similar pixels [15, 30]. For instance, it is hard to distinguish 'ceiling' and 'wall' pixels with limited local context (see Figure 1). To address this issue, long-range dependencies among image units are required.
- Another problem is how to effectively take advantages of RGB and depth information. A possible solution is to merge these two sources of information directly, i.e., concatenating the two channels as the input or fusing the outputs of two independent labeling processes for the two channels separately [3, 11, 36]. However, both solutions ignore the strong correlation between RGB and depth channels, which could be beneficial for indoor semantic labeling.

Recently, recurrent neural networks (RNNs) [5], which have shown great success in neural language process (NLP)

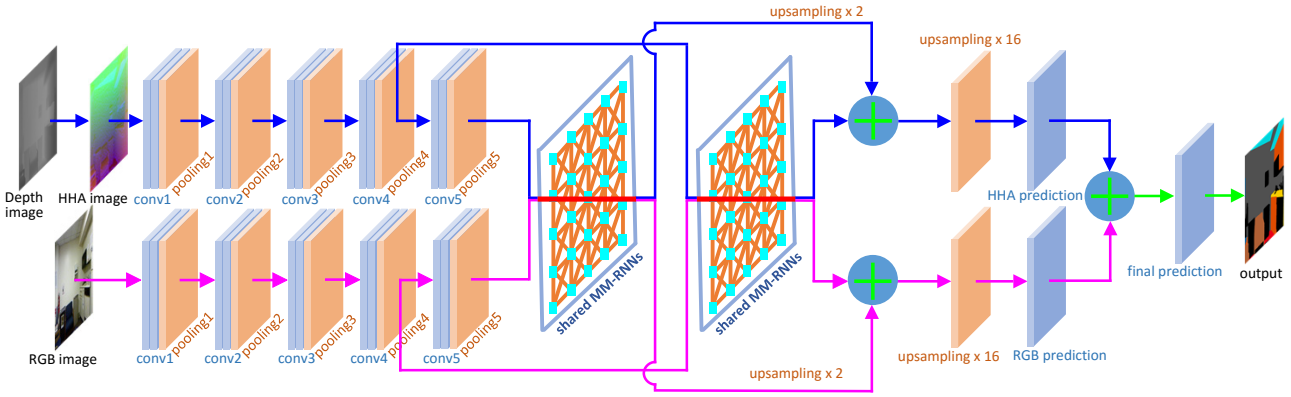

Figure 2. Detailed network architecture of our method. We use two separate CNNs to extract deep features of RGB and depth modalities, respectively. Then our MM-RNNs take as input these two kinds of features and output feature maps of each modality. The obtained feature maps possess property of both its own modality and another modality. After that, we upsample the feature maps to the input size and the final output is derived by summation over the outputs of these two modalities. Note that in this work, we replace a depth image with an HHA image (image with three channels representing **H**orizontal disparity, **H**eight above ground and **A**ngle with gravity respectively) [11] to extract depth features. It is also worth noting that MM-RNNs allow 'memory' sharing between modalities in that the RNNs are shared among all modalities.

[8, 10], have been brought to the computer vision community [40, 6, 1, 9, 34, 28, 37, 41, 17] owing to its capability of modeling long-range dependencies among sequential data. Among them, [28] proposes the graphical-structured RNNs to capture long-range contextual information in images. Nonetheless, this work models dependencies with a single modality, and thus is not suitable for RGB-D scene labeling.

To address the above issues, we propose a multimodal recurrent neural networks (MM-RNNs) for RGB-D scene labeling. For the first problem, we encode the local context of each image unit into RNNs, and the local contexts of all image units are connected in a structural undirected cyclic graph, which results in the long-range context in the entire image. Nevertheless, conventional RNNs are used for sequential and not suitable to be directly applied to structural data. We thus decompose the structural undirected cyclic graph into several directed acyclic graphs as in [28]. Though this method is able to capture long-range context, it only models semantic dependencies of one single modality. To deal with the second problem, we extend this single-modal RNNs to a multimodal one, which takes correlation of multiple modalities into account. Specifically, our MM-RNNs allow 'memory' sharing across multiple modalities. By sharing 'memory', each modality not only possesses its own property but also has the attributes of other modalities, and thus becomes more discriminative to distinguish pixels. Integrating MM-RNNs with convolutional neural networks (CNNs), we propose an end-to-end network for RGB-D scene labeling. To further explore dependencies in different levels in CNNs for pixel classification, we utilize multiple MM-RNNs to learn the dependencies in different levels respectively. Taking into consideration time consumption, we in this paper only use two MM-RNNs to model dependencies of two layers in CNNs. The proposed network architecture is illustrated in Figure 2.

Extensive experimental results on two benchmarks, NYU depth V1 and V2, demonstrate the effectiveness of our proposed method in comparison with state-of-the-arts and baselines.

2. Related Work

Scene labeling: Scene labeling is one of the most challenging problems in computer vision. Several previous non-parametric approaches try to transfer the labels of training data to the query images and perform label inference in a probabilistic graphical model (PGM). [33] proposes a superparsing method for image labeling by comparing superpixels of the query image in a retrieval dataset, and infers their labels via markov random field (MRF). [39] suggests to incorporate context information to improve both image retrieval and superpixel classification, and builds a four-connected pairwise MRF for scene labeling. [27] proposes to integrate parametric and non-parametric models for scene labeling.

Owing to the powerfulness in feature extraction, CNNs [16, 31] have been increasingly utilized in scene labeling. [19] introduces fully convolutional networks (FCNs) for image labeling. [2] proposes to integrate deep convolutional nets with fully connected crfs for scene labeling. [7] proposes to learn hierarchical features with CNNs for scene labeling. In [13], a semi-supervised decoupled deep neural network is proposed for semantic segmentation. [22] combines convolution networks with deconvolution networks for semantic labeling. Different from the above methods,

we integrate CNNs with our MM-RNNs to capture long-range context and thus let it 'see' the whole image to make a more accurate decision for pixel classification.

RGB-D scene labeling: Based on depth information provided by depth sensors such as Microsoft Kinect, many RGB-D scene parsing approaches have been proposed. [25] proposes to use depth descriptors based on traditional multi-channel feature such as gradient, color and surface normal to represent RGB and depth features for labeling. In [15], a rich feature set consisting of LBP [23], texton [26], SPIN [14], SIFT [20] and HOG [4] is extracted to represent object appearance, and scene labeling is achieved by the proposed high-order conditional random field (CRF) in [15]. [3] proposes to utilize CNNs to learn hierarchical RGB-D features for scene labeling. [36] suggests an unsupervised joint feature learning encoding model for RGB-D scene labeling, and [30] proposes a support inference framework for scene labeling. Despite achieving promising results for RGB-D scene labeling, the aforementioned approaches do not take the strong correlation between RGB and depth channels into account, and just concatenate these two kinds of information together as final input, which limits the further improvement of performance. Our model differs from these methods by allowing feature sharing across RGB and depth modalities, and thus makes them more discriminative for pixel classification.

RNNs on image processing: RNNs have been first introduced to deal with sequential prediction tasks [8, 10], and then extended to multi-dimensional image processing tasks [9] such as image completion [34], image classification [41], scene parsing [1, 28], etc. [34] proposes to model discrete probability of raw pixel values with RNNs for image completion. [9] applies multi-dimensional RNNs to handwriting recognition. Inspired by [9], [1] introduces a two-dimensional long-short term memory (LSTM) for outdoor scene labeling. Different from [1], [41] proposes to utilize RNNs to model spatial dependencies among image units from multiple scales, and combine these dependencies for image classification.

The most relevant work related to ours is [28], which uses graphical RNNs to model long-range dependencies among image units. However, this approach is designated for one single modality, and not suitable to be directly applied to RGB-D scene labeling. Considering correlation between multiple modalities, we extend this single-modal RNNs to MM-RNNs and apply it to RGB-D scene labeling. Note that our work is also different from [17]. In [17], long-short term memory (LSTM) is used for modeling the context of single modality, and then simple concatenation is adopted to fuse multiple modalities. The cross-modality correlation is not learned. Different from [17], our MM-RNNs are able to simultaneously learn the contexts of two modalities and their cross-modality correlation, which improves the discrimination of both two modalities for pixel classification. Experiments on two benchmarks demonstrate the effectiveness of our MM-RNNs over the two fusion methods and other RGB-D scene labeling approaches.

3. The Proposed Approach

In this section, we first review the basic RNNs in Sec. 3.1, then introduce two simple extensions of single-modal RNNs in Sec. 3.2, present our MM-RNNs in Sec. 3.3, and elaborate our final RGB-D scene labeling work in Sec. 3.4.

3.1. Review of Recurrent Neural Networks (RNNs)

RNNs [5] are designated for addressing sequential data tasks. Specifically, the hidden layer h_t in RNNs at time step t is represented with a non-linear function over current input x_t and hidden layer at previous time step h_{t-1}. The output layer y_t is connected to hidden layer h_t.

Given an input sequence $\{x_t\}_{t=1,2,\cdots,T}$, the hidden and output layers at time step t can be obtained through

$$\begin{cases} h_t = \phi(Ux_t + Wh_{t-1} + b_h) \\ y_t = \sigma(Vh_t + b_y) \end{cases} \tag{1}$$

where U, W and V denote shared transformation matrices; b_h and b_y are bias terms; and $\phi(\cdot)$ and $\sigma(\cdot)$ are non-linear functions. Figure 3 shows the structure of basic RNNs.

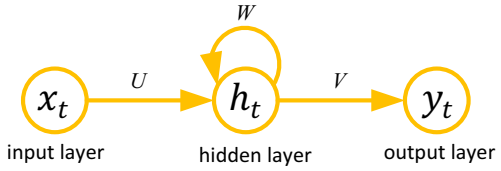

Figure 3. Structure of basic RNNs.

Since the inputs are progressively stored in hidden layers, RNNs are able to keep 'memory' of the whole sequence and thus model long-range dependencies among the sequence. In this process, matrices W and V play crucial roles. The W is responsible for storing 'memory' of whole sequence and V is used to transform this memory to output layer.

3.2. Two simple extensions of single-modal RNNs

Before introducing our MM-RNNs, we first analyse two simple extensions of single-modal RNNs which also take the advantages of multiple modalities.

The first straightforward method is to concatenate the inputs from these modalities into one single input as follows

$$\begin{cases} x_t = cat(x_t^1, x_t^2, \cdots, x_t^M) \\ h_t = \phi(Ux_t + Wh_{t-1} + b_h) \\ y_t = \sigma(Vh_t + b_y) \end{cases} \tag{2}$$

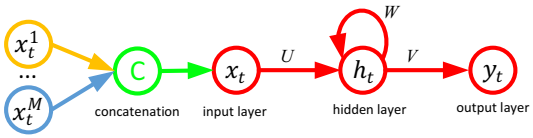

Figure 4. Extension of single-modal RNNs by concatenating all modalities.

where x_t^m denotes input of the m^{th} modality, M the number of modalities (in this paper, $M = 2$), and cat the concatenation operation. Figure 4 illustrates this extension. The approach completely ignores multimodal properties of different inputs and has no explicit mechanism to model the correlation across modalities.

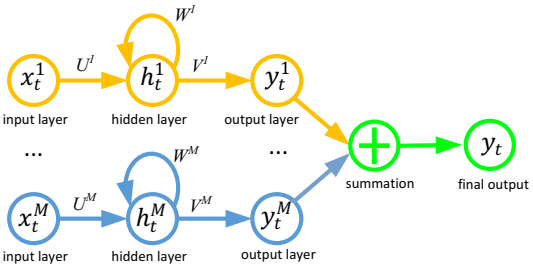

Figure 5. Extension of single-modal RNNs by merging output layers of all modalities.

The second possible solution, shown in Figure 5, is to treat each modality completely independent. Namely, multiple RNNs are utilized in parallel for each modality and the final output is derived by fusing their outputs as follows

$$\begin{cases} h_t^m = \phi(U^m x_t^m + W^m h_{t-1}^m + b_h^m) \\ y_t^m = \sigma(V^m h_t^m + b_y^m) \\ o_t = \sum_{m=1}^{M} w^m y_t^m \end{cases} \tag{3}$$

where x_t^m, h_t^m and y_t^m denote respectively the input, the hidden layer and the output layer of the m^{th} modality. U^m, W^m and V^m represent the shared transformation matrices of the m^{th} modality, b_h^m and b_y^m are bias terms of the m^{th} modality, w^m denotes the weight of the m^{th} modality, and o_t is the final output via weighted summation over output layers of all modalities. Though this approach is able to separately store useful information explicitly for each modality, the across-modality interaction is not taken into account. Therefore, the cross-modality correlation is not incorporated into the learning process.

3.3. Multimodal RNNs (MM-RNNs)

As discussed above, both straightforward extensions are not capable of capturing the strong correlation among modalities, limiting the further improvement of performance. Basically, neither extension encodes the cross-modality correlation by overlooking the relationships of W and V among different modalities. To address this issue, we propose the MM-RNNs to allow cross-modality 'memory' sharing and thus encode the cross-modality correlation into the learning process.

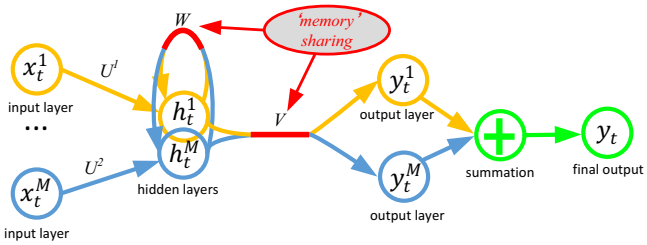

Figure 6. Structure of MM-RNNs.

Noticing that the shared transformation matrices W and V in Eq (1) play key roles in keeping the 'memory' of whole sequence, we develop the new MM-RNNs which are able to explicitly model long-range dependencies both within the same modality and across modalities. Our key idea is to share weights, which are capable of capturing 'memory' across modalities. Specifically, we use multiple parallel RNNs to learn each modality respectively. However, the transformation matrices W and V are shared across all RNNs. In this way, the 'memory' of each modality is shared by all other modalities, and the inter-correlation among modalities is thus encoded into the learning process. This process can be mathematically expressed with

$$\begin{cases} h_t^m = \phi(U^m x_t^m + W h_{t-1}^m + b_h^m) \\ y_t^m = \sigma(V h_t^m + b_y^m) \\ o_t = \sum_{m=1}^{M} w^m y_t^m \end{cases} \tag{4}$$

where W and V are transformation matrices across modalities. For U^m of each modality, it is not responsible for storing any 'memory' and thus not shared across modalities. Note that Eq (3) and Eq (4) are **different**. In Eq (3), each modality has its own W^m and V^m, and they are **NOT shared** across modalities. While in Eq (4), the W and V are **shared** across modalities to learn the correlation among modalities. Figure 6 demonstrates the structure of MM-RNNs.

3.4. Graphical MM-RNNs for RGB-D Labeling

Our goal is to model long-range dependencies among image units. For an image, the interaction among image units are encoded into an undirected cyclic graph (see Figure 7(b)). Nevertheless, due to the loopy structure of undirected cyclic graph, our MM-RNNs are not suitable to be directly applied to images. To address this problem, we approximate the topology of undirected cyclic graph by dividing it into four directed acyclic graphs along southeast, southwest, northeast and northwest directions as in [28] (one of

the four directed acyclic graphs is depicted in Figure 7(c)). Note that different from [28], our graphical MM-RNNs take two input modalities simultaneously.

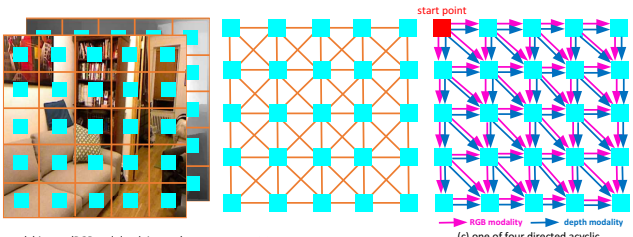

Figure 7. Undirected cyclic graph (a,b) and one of its four directed acyclic graphs (c). Note that each vertex in (c) receives multiple modalities of all its predecessors, and each modality shares 'memory' with other ones.

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ represent the directed acyclic graph, where $\mathcal{V} = \{v_i\}_{i=1,2,\cdots,N}$ is vertex set and $\mathcal{E} = \{e_{ij}\}$ denotes the edge set in which e_{ij} represents directed edge from vertex v_i to v_j. The structure of MM-RNNs follows the same topology as $\mathcal{G}$. A forward pass can be seen as traversing $\mathcal{G}$ from start point, and each vertex depends on its all predecessors. Therefore, for vertex v_i its hidden layer $h^m_{v_i}$ for modality m is expressed as a non-linear function over current input $x^m_{v_i}$ of modality m and summation of hidden layer of all its predecessors of the m^{th} modality. Specifically, the hidden layer $h^m_{v_i}$, output layer $y^m_{v_i}$ and final output o_{v_i} at vertex v_i are calculated by

$$\begin{cases} h^m_{v_i} = \phi(U^m x^m_{v_i} + W \sum\limits_{v_j \in \mathcal{P}_\mathcal{G}(v_i)} h^m_{v_j} + b^m_h) \\ y^m_{v_i} = \sigma(V h^m_{v_i} + b^m_y) \\ o_{v_i} = \sum\limits_{m=1}^{M} w^m y^m_{v_i} \end{cases} \tag{5}$$

where $\mathcal{P}_\mathcal{G}(v_i)$ represents the predecessor set of v_i in $\mathcal{G}$. W stores 'memory' across modalities; V transforms this memory to output, and the final output at v_i is derived via weighted summation over all modalities.

To calculate derivatives in the back propagation, each vertex is processed in the reverse order of forward pass sequence. Specifically, for vertex v_i, we look at the forward passes of its successors. Let $\mathcal{S}_\mathcal{G}(v_i)$ represent the set of direct successors for v_i; then for each vertex $v_k \in \mathcal{S}_\mathcal{G}(v_i)$, its hidden layer at modality m is computed as

$$h^m_{v_k} = \phi(U^m x^m_{v_k} + W h^m_{v_i} + W \sum_{v_l \in \mathcal{P}_\mathcal{G}(v_k) - \{v_i\}} h^m_{v_l} + b^m_h) \tag{6}$$

Combining (5) and (6), we can see that the errors back-propagated to the hidden layer come from two sources: direct errors from v_i (i.e., $\frac{\partial o_{v_i}}{\partial h^m_{v_i}} = \frac{\partial o_{v_i}}{\partial y^m_{v_i}} \frac{\partial y^m_{v_i}}{\partial h^m_{v_i}}$) and summation over indirect errors from all its successors $v_k \in \mathcal{S}_\mathcal{G}(v_i)$ (i.e., $\sum\limits_{v_k \in \mathcal{S}_\mathcal{G}(v_i)} \frac{\partial o_{v_k}}{\partial h^m_{v_i}} = \sum\limits_{v_k \in \mathcal{S}_\mathcal{G}(v_i)} \frac{\partial o_{v_k}}{\partial y^m_{v_k}} \frac{\partial y^m_{v_k}}{\partial h^m_{v_k}} \frac{\partial h^m_{v_k}}{\partial h^m_{v_i}}$). Therefore, we can compute the derivatives at vertex v_i for the m^{th} modality as

$$\begin{cases} \mathrm{d}h^m_{v_i} = V^T \sigma'(y^m_{v_i}) + \sum\limits_{v_k \in \mathcal{S}_\mathcal{G}(v_i)} W^T \mathrm{d}h^m_{v_k} \circ \phi'(h^m_{v_k}) \\ \nabla W^m_{v_i} = \sum\limits_{v_k \in \mathcal{S}_\mathcal{G}(v_i)} \mathrm{d}h^m_{v_k} \circ \phi'(h^m_{v_k})(h^m_{v_i})^T \\ \nabla U^m_{v_i} = \mathrm{d}h^m_{v_i} \circ \phi'(h^m_{v_i})(x^m_{v_i})^T \\ \nabla V^m_{v_i} = \sigma'(y^m_{v_i})(h^m_{v_i})^T \\ \nabla b^m_h = \mathrm{d}h^m_{v_i} \circ \phi'(h^m_{v_i}) \\ \nabla b^m_y = \sigma'(y^m_{v_i}) \end{cases} \tag{7}$$

where $\circ$ is the Hadamard product, $\sigma'(\cdot) = \frac{\partial L}{\partial o(\cdot)} \frac{\partial o(\cdot)}{\partial y(\cdot)} \frac{\partial y(\cdot)}{\partial \sigma}$ denotes the derivative of loss function with respect to function σ, and $\phi'(\cdot) = \frac{\partial h}{\partial \phi}$. We adopt the average cross entropy loss function to compute L. Note that Eq (7) is to compute the derivatives at vertex v_i for the m^{th} modality. For W and V, which are shared across modalities, their derivatives at vertex v_i are calculated as the following

$$\nabla W_{v_i} = \sum_{m=1}^{M} \nabla W^m_{v_i}, \quad \nabla V_{v_i} = \sum_{m=1}^{M} \nabla V^m_{v_i} \tag{8}$$

With Eq (5), (7) and (8), we can perform forward and backward passes on a directed acyclic graph. Following [28], we decompose an undirected cyclic graph into four directed acyclic graphs, denoted by $\mathcal{G}^\mathcal{U} = \{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_4\}$, where $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_4$ are four directed acyclic graphs. For each $\mathcal{G}_d$ ($d = 1, \cdots, 4$), we obtain the corresponding hidden layer for the m^{th} modality in MM-RNNs as follows

$$\begin{cases} h^m_{v_i,d} = \phi(U^m_d x^m_{v_i} + W_d \sum\limits_{v_j \in \mathcal{P}_{\mathcal{G}_d}} h^m_{v_j,d} + b_{h^m_{h,d}}) \\ y^m_{v_i} = \sigma(\sum\limits_{\mathcal{G}_d \in \mathcal{G}^\mathcal{U}} V_d h^m_{v_i,d} + b^m_y) \\ o_{v_i} = \sum\limits_{m=1}^{M} w^m y^m_{v_i} \end{cases} \tag{9}$$

where $h^m_{v_i,d}$ denotes the hidden layer of the m^{th} modality at vertex v_i in directed acyclic graph $\mathcal{G}_d$, U^m_d represents transformation matrix between input layer and hidden layer for modality m in $\mathcal{G}_d$, W_d and V_d are shared transformation matrices between previous hidden layer and current hidden layer, hidden layer and output layer in $\mathcal{G}_d$, $y^m_{v_i}$ is the output layer for modality m, $b_{h^m_{h,d}}$ and b^m_y are bias terms, and o_{v_i} is the final output at vertex v_i.

With Eq (9), we can calculate loss L via

$$L = -\frac{1}{N} \sum_{v_i \in \mathcal{G}^\mathcal{U}} \sum_{c=1}^{C} \log(o^c_{v_i} Y^c_{v_i}) \tag{10}$$

where N is the number of image units, C the number of semantic classes, o_{v_i} the class likelihood vector, and Y_{v_i} the

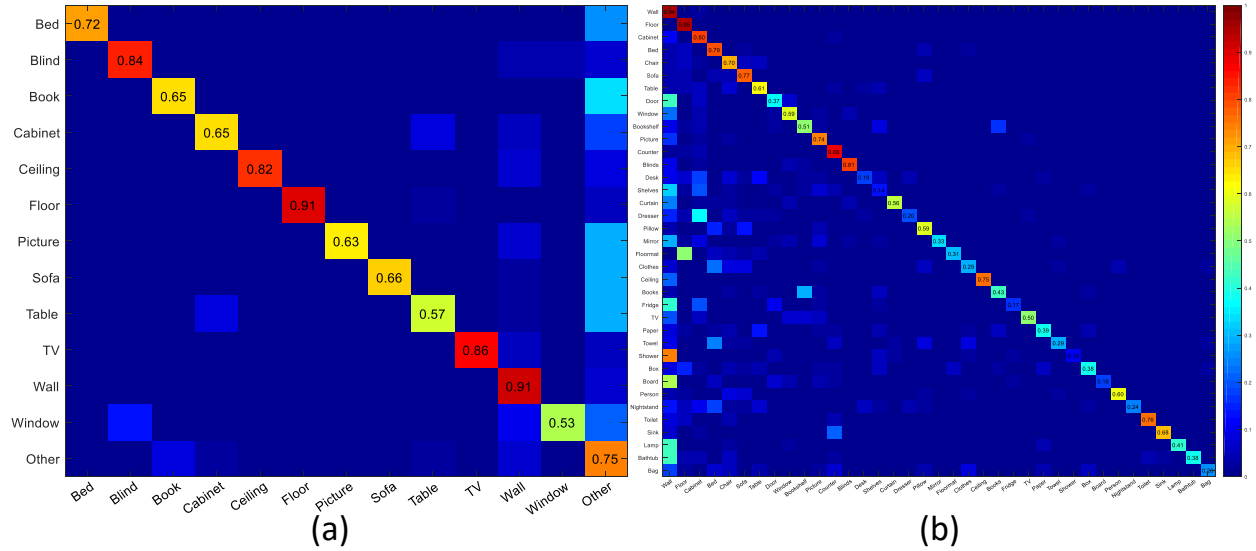

Figure 8. Confusion matrices of our approach on NYU depth V1 (a) of 13 classes and NYU depth V2 (b) of 37 classes.

binary label indicator vector for image unit at v_i. The error back-propagated from MM-RNNs to the convolutional layer at v_i for modality m is computed with

$$\nabla x^m_{v_i} = \sum_{\mathcal{G}_d \in \mathcal{G}^{\mathcal{U}}} (U^m_d)^T \mathrm{d}h^m_{v_i,d} \circ \phi'(h^m_{v_i,d}) \qquad (11)$$

So far, we have introduced our MM-RNNs with forward and backward propagations. By sharing 'memory', MM-RNNs are capable of simultaneously modeling dependencies across multiple modalities. Besides, it can be easily embedded into other networks as an intermediate layer to capture the inter-correlation among modalities.

4. Experimental Results

We test our method on two benchmarks: NYU depth V1 [29] and V2 [30], and adopt two metrics, pixel accuracy and class accuracy, for evaluation.

4.1. Implementation Details

We use the architecture and parameters from the VGG-16 network [31] before the 5^{th} pooling layer. Non-linear activation function $\phi = \max(0, x)$ and σ is *softmax* function. In practice, function σ is applied after upsampling layers (see Figure 2), and Eq (10) is utilized to calculate the loss between prediction and ground truth. The upsampling factor is set to 2. Namely, the ground truth maps subsampled during training stage, while the final label prediction maps are further upsampled to original input size by simple bilinear interpolation [19] in testing phase. The full network is trained by stochastic gradient descent (SGD) with momentum. The learning rate is initialized to be 10^{-3} and decays exponentially with the rate of 0.9 after 10 epochs. The results are reported after 35 epochs. The entire network is implemented in MATLAB using MatConvNet [35] on a single NVIDIA GTX TITAN Z GPU with 6GB memory.

4.2. NYU depth V1 Dataset

Figure 8(a) shows the confusion matrix of 13 classes of our approach on NYU depth V1. The NYU depth V1 dataset [29] consists of 2347 RGB-D images captured in 64 different indoor scenes labeled with 12 categories plus an unknown class (13 classes in total). We follow the usual split protocol [29] (60% for training and 40% for testing) to obtain training and testing images. Table 1 shows the comparisons of pixel and class accuracies between our method and other algorithms[1]. Table 2 demonstrates the comparison of individual class labeling performance on NYU depth V1. Figure 9 shows some qualitative labeling results on NYU depth V1.

Table 1. Quantitative results and comparisons on NYU depth V1.

Method	Pixel Accuracy	Class Accuracy
Wang et al. [36]	-	63.3
Silberman et al. [29]	-	53.0
Pei et al. [24]	-	50.5
Hermans et al. [12]	44.4	59.5
Wolf et al. [38]	67.8	63.6
Khan et al. [15]	70.6	66.5
MM-RNNs*	71.7	69.3
MM-RNNs**	72.1	69.8
MM-RNNs	**78.0**	**73.0**

[1]MM-RNNs* and MM-RNNs** are two simple extensions which merge inputs and outputs respectively, and their structures and corresponding network architectures are demonstrated in the supplementary materia.

Table 2. Individual class accuracy performance of 13-class setting on NYU depth V1 dataset.

Class	Bed	Blind	Book	Cabinet	Ceiling	Floor	Picture	Sofa	Table	TV	Wall	Window	Other
Wang et al. [36]	62.6	60.4	70.0	44.8	75.4	81.4	51.7	54.5	30.4	73.2	71.6	38.7	6.3
Hermans et al. [12]	50.7	57.6	59.8	57.8	**92.8**	89.4	55.8	70.9	48.4	81.7	75.9	18.9	13.5
Wolf et al. [38]	53.5	57.9	**86.4**	34.0	85.7	**95.7**	61.1	64.4	60.1	81.9	79.7	18.1	48.3
Khan et al. [15]	66.8	67.7	47.5	**72.6**	79.2	67.8	53.4	**75.1**	**69.3**	78.6	86.2	**62.0**	38.1
MM-RNNs*	68.8	79.8	58.9	62.9	77.9	86.3	60.9	63.4	53.0	82.1	87.0	48.7	71.4
MM-RNNs**	69.5	79.9	57.3	63.7	78.4	87.1	61.8	66.9	50.2	82.7	86.8	50.9	71.7
MM-RNNs	**71.8**	**84.0**	64.8	65.0	82.2	90.6	**62.7**	65.7	57.3	**86.1**	**91.0**	53.2	**74.6**

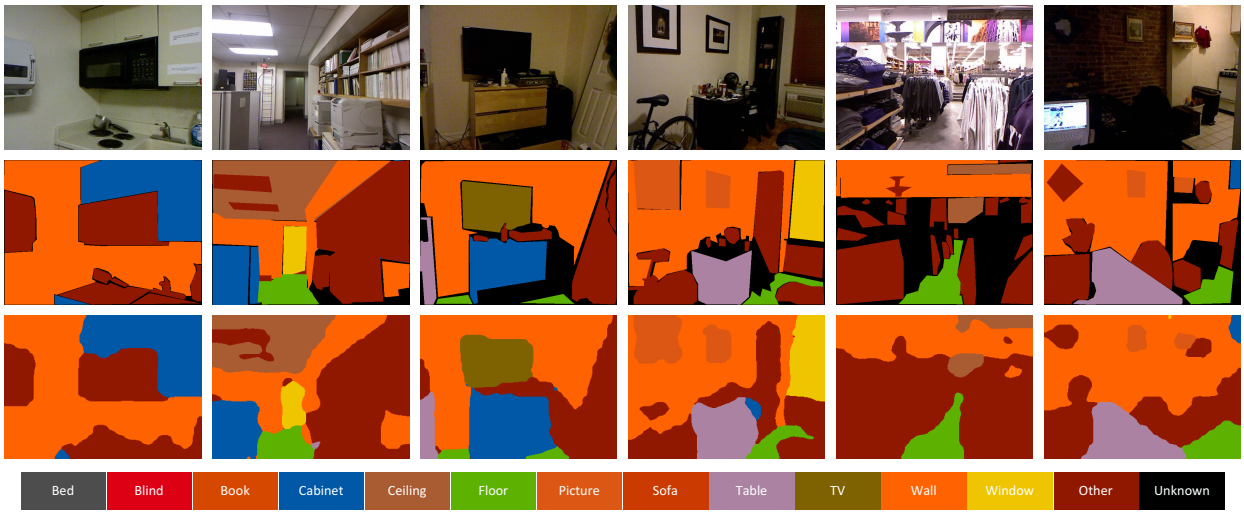

Figure 9. Quantitative labeling results on NYU depth V1. **First row:** input images. **Second row:** ground truth. **Third row:** our results.

From Table 1, our proposed MM-RNNs outperform other methods and the baselines, i.e., the two other extensions of single-modal RNNs, on both pixel and class accuracies. Our MM-RNNs improve the pixel accuracy from 70.6% to 78.0%, and the class accuracy from 66.5% to 73.0%. By sharing 'memory' across modalities, each modality becomes more discriminative for pixel classification, and thus boosts the performance.

Table 3. Quantitative results and comparisons on NYU depth V2.

Method	Pixel Accuracy	Class Accuracy
Silberman et al. [30]	-	17.5
Ren et al. [25]	-	20.2
Gupta et al. [11]	58.3	30.7
Wang et al. [36]	51.6	29.2
Khan et al. [15]	50.7	43.9
Li et al. [17]	-	49.4
MM-RNNs*	67.1	45.9
MM-RNNs**	68.5	46.6
MM-RNNs	**70.9**	**50.2**

4.3. NYU depth V2 Dataset

Figure 8(b) shows the confusion matrix of 37 classes of our approach on NYU depth V2. The NYU depth V2 dataset [30] contains 1449 RGB-D images captured in 464 different indoor scenes. Each image is labeled with 37 semantic classes as in [32]. Following the split protocol [30], 795 image are used for training and the rest for testing. Table 3 shows the comparisons of pixel and class accuracies between our method and other algorithms. Table 4 demonstrates the comparison of individual class labeling performance on NYU depth V1. Figure 10 shows some qualitative labeling results on NYU depth V2.

From Table 3, we can see that our method outperforms other approaches as well as the two other extensions. Our MM-RNNs improve the pixel accuracy from 58.3% to 70.9%, and the class accuracy from 49.4% to 50.2%.

5. Conclusion

This paper proposes a modality fusion method for RGB-D scene labeling by extending single-modal RNNs to MM-RNNs. With MM-RNNs, each modality possesses proper-

Table 4. Individual class accuracy performance of 37-class setting on NYU depth V2 dataset.

Class	Wall	Floor	Cabinet	Bed	Chair	Sofa	Table	Door	Window	Bookshelf	Picture	Counter	Blinds	Desk	Shelves	Curtain	Dresser	Pillow	Mirror
Wang et al. [36]	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
Silberman et al. [30]	60.7	77.8	33.0	40.3	32.4	25.3	21.0	5.9	29.7	22.7	35.7	33.1	40.6	4.7	3.3	27.4	13.3	18.9	4.4
Ren et al. [25]	60.0	74.4	37.1	42.3	32.5	28.2	16.6	12.9	27.7	17.3	32.4	38.6	26.5	10.1	6.1	27.6	7.0	19.7	17.9
Gupta et al. [11]	67.4	80.5	41.4	56.4	40.4	44.8	30.0	12.1	34.1	20.5	38.7	50.7	44.7	10.1	1.6	26.3	21.6	31.3	14.6
Khan et al. [15]	65.7	62.5	40.1	32.1	44.5	50.8	43.5	**51.6**	49.2	36.3	41.4	39.2	55.8	**48.0**	**45.2**	53.1	55.3	50.5	46.1
Li et al. [17]	79.6	83.5	69.3	77.0	58.3	64.9	42.6	47.0	43.6	**59.5**	**74.5**	68.2	74.6	33.6	13.1	53.2	**56.5**	48.0	**47.7**
MM-RNNs*	89.9	87.2	74.3	72.9	67.9	74.8	55.6	33.1	53.7	47.9	72.9	81.6	82.7	15.7	8.3	48.3	21.8	45.1	29.3
MM-RNNs**	91.2	89.7	76.8	75.3	69.8	73.6	54.7	35.8	55.9	50.8	70.6	83.4	86.1	16.2	9.9	48.7	20.1	46.9	31.3
MM-RNNs	**93.8**	**94.7**	**80.2**	**78.9**	**70.1**	**76.5**	**61.3**	37.2	**59.1**	51.0	73.8	**88.0**	**80.6**	18.0	13.8	**55.8**	20.4	**59.4**	32.9

Class	Floormat	Clothes	Ceiling	Books	Fridge	TV	Paper	Towel	Shower	Box	Whiteboard	Person	Nightstand	Toilet	Sink	Lamp	Bathtub	Bag
Wang et al. [36]	13.8	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
Silberman et al. [30]	7.1	6.5	73.2	5.5	1.4	5.7	12.7	0.1	3.6	0.1	0.0	6.6	6.3	26.7	25.1	15.9	0.0	0.0
Ren et al. [25]	20.1	9.5	53.9	14.8	1.9	18.6	11.7	12.6	5.4	3.3	0.2	13.6	9.2	35.2	28.9	14.2	7.8	1.2
Gupta et al. [11]	28.2	8.0	61.8	5.8	14.5	14.4	14.1	19.8	6.0	1.1	12.9	1.5	15.7	52.5	47.9	31.2	29.4	0.2
Khan et al. [15]	**54.1**	**35.4**	50.6	39.1	**53.6**	**50.1**	35.4	39.9	**41.8**	36.3	60.6	35.6	**32.5**	31.8	22.5	26.3	38.5	**37.3**
Li et al. [17]	0.0	22.7	70.2	**49.7**	0.0	0.0	**52.1**	**60.6**	0	17.6	**93.9**	**77.0**	0	**81.8**	58.4	**67.6**	**72.6**	7.5
MM-RNNs*	26.4	25.3	74.8	39.1	15.8	47.1	37.1	23.6	8.1	33.9	17.1	50.9	19.9	61.3	66.1	28.4	34.7	20.7
MM-RNNs**	27.1	23.2	75.1	40.5	15.1	45.6	33.3	24.5	7.8	32.8	18.0	52.1	21.3	62.7	64.5	35.1	35.1	23.8
MM-RNNs	31.2	29.0	**75.3**	42.5	17.2	**50.1**	39.2	28.8	10.3	**37.6**	18.1	59.7	23.6	75.8	**67.6**	41.4	37.5	25.8

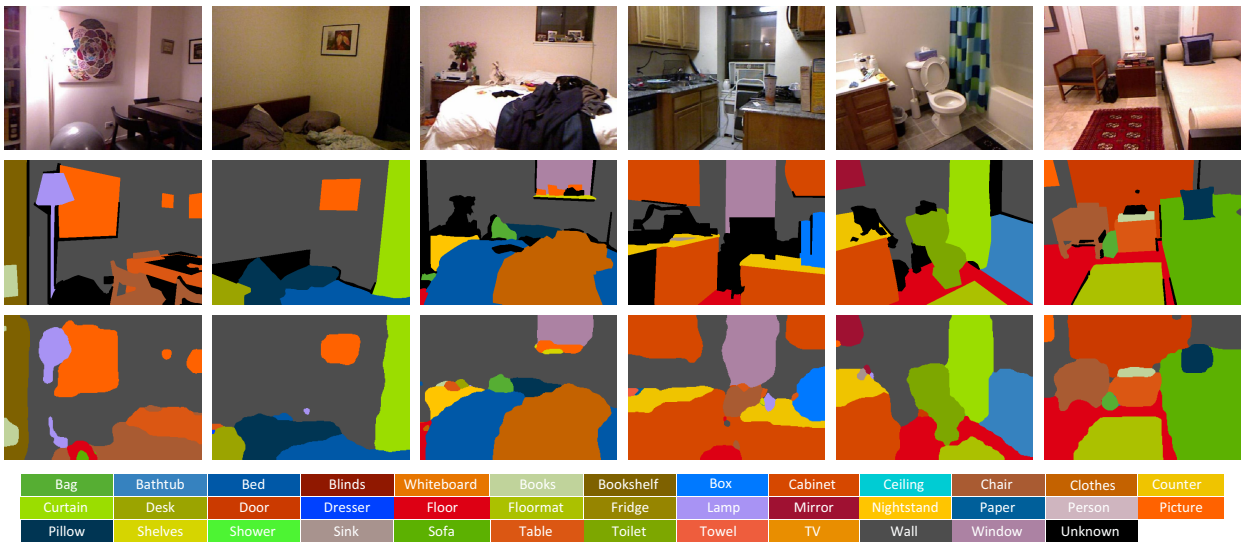

Figure 10. Quantitative labeling results on NYU depth V2. **First row:** input images. **Second row:** ground truth. **Third row:** our results.

ties of its own and other modalities, and becomes more discriminative. Besides, we also introduce two simple extensions of single-modal RNNs and demonstrate that our proposed method outperforms both. Integrating with CNNs, we build an end-to-end network for RGB-D scene labeling. Extensive experiments on two large-scale benchmarks evidence the effectiveness of our approach.

References

[1] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki. Scene labeling with lstm recurrent neural networks. In *CVPR*, 2015.

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.

[3] C. Couprie, C. Farabet, L. Najman, and Y. LeCun. Indoor semantic segmentation using depth information. In *ICLR*, 2013.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[5] J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[6] H. Fan, X. Mei, D. Prokhorov, and H. Ling. Multi-level contextual rnns with attention model for scene labeling. 2016.

[7] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *TPAMI*, 35(8):1915–1929, 2013.

[8] A. Graves. Sequence transduction with recurrent neural networks. In *ICML*, 2014.

[9] A. Graves, S. Fernandez, and J. Schmidhuber. Multi-dimensional recurrent neural networks. In *ICANN*, 2007.

[10] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *ICASSP*, 2013.

[11] S. Gupta, P. Arbeláez, R. Girshick, and J. Malik. Indoor scene understanding with rgb-d images: Bottom-up segmentation, object detection and semantic segmentation. *IJCV*, 112(2):133–149, 2015.

[12] A. Hermans, G. Floros, and B. Leibe. Dense 3d semantic mapping of indoor scenes from rgb-d images. In *ICRA*, 2014.

[13] S. Hong, H. Noh, and B. Han. Decoupled deep neural network for semi-supervised semantic segmentation. In *NIPS*, 2015.

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

[15] S. H. Khan, M. Bennamoun, F. Sohel, R. Togneri, and I. Naseem. Integrating geometrical context for semantic labeling of indoor scenes using rgbd images. *IJCV*, 117(1):1–20, 2016.

[16] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[17] Z. Li, Y. Gan, X. Liang, Y. Yu, H. Cheng, and L. Lin. Lstm-cf: Unifying context modeling and fusion with lstms for rgb-d scene labeling. In *ECCV*, 2016.

[18] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *CVPR*, 2009.

[19] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

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

[21] M. Najafi, S. T. Namin, M. Salzmann, and L. Petersson. Sample and filter: Nonparametric scene parsing via efficient filtering. In *CVPR*, 2016.

[22] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015.

[23] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *TPAMI*, 24(7):971–987, 2002.

[24] D. Pei, H. Liu, Y. Liu, and F. Sun. Unsupervised multimodal feature learning for semantic image segmentation. In *IJCNN*, 2013.

[25] X. Ren, L. Bo, and D. Fox. Rgb-(d) scene labeling: Features and algorithms. In *CVPR*, 2012.

[26] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 81(1):2–23, 2009.

[27] B. Shuai, G. Wang, Z. Zuo, B. Wang, and L. Zhao. Integrating parametric and non-parametric models for scene labeling. In *CVPR*, 2015.

[28] B. Shuai, Z. Zuo, G. Wang, and B. Wang. Dag-recurrent neural networks for scene labeling. In *CVPR*, 2016.

[29] N. Silberman and R. Fergus. Indoor scene segmentation using a structured light sensor. In *ICCV Workshop*, 2011.

[30] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[32] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, 2015.

[33] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *ECCV*, 2010.

[34] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *ICML*, 2016.

[35] A. Vedaldi and K. Lenc. Matconvnet: Convolutional neural networks for matlab. In *ACM MM*, 2015.

[36] A. Wang, J. Lu, J. Cai, G. Wang, and T.-J. Cham. Unsupervised joint feature learning and encoding for rgb-d scene labeling. *TIP*, 24(11):4459–4473, 2015.

[37] W. Wang, Z. Cui, Y. Yan, J. Feng, S. Yan, X. Shu, and N. Sebe. Recurrent face aging. In *CVPR*, 2016.

[38] D. Wolf, J. Prankl, and M. Vincze. Fast semantic segmentation of 3d point clouds using a dense crf with learned parameters. In *ICRA*, 2015.

[39] J. Yang, B. Price, S. Cohen, and M.-H. Yang. Context driven scene parsing with attention to rare classes. In *CVPR*, 2014.

[40] H. Zuo, H. Fan, E. Blasch, and H. Ling. Combining convolutional and recurrent neural networks for human skin detection. *IEEE SPL*, 24(3):289–293, 2017.

[41] Z. Zuo, B. Shuai, G. Wang, X. Liu, X. Wang, B. Wang, and Y. Chen. Learning contextual dependence with convolutional hierarchical recurrent neural networks. *TIP*, 25(7):2983–2996, 2016.