2019 IEEE Winter Conference on Applications of Computer Vision

Scene Parsing via Dense Recurrent Neural Networks with Attentional Selection

Heng Fan Peng Chu Longin Jan Latecki Haibin Ling
Department of Computer and Information Sciences, Temple University, USA
{hengfan,tug29183,latecki,hbling}@temple.edu

Abstract

Recurrent neural networks (RNNs) have shown the ability to improve scene parsing through capturing long-range dependencies among image units. In this paper, we propose dense RNNs for scene labeling by exploring various long-range semantic dependencies among image units. Different from existing RNN based approaches, our dense RNNs are able to capture richer contextual dependencies for each image unit by enabling immediate connections between each pair of image units, which significantly enhances their discriminative power. Besides, to select relevant dependencies and meanwhile to restrain irrelevant ones for each unit from dense connections, we introduce an attention model into dense RNNs. The attention model allows automatically assigning more importance to helpful dependencies while less weight to unconcerned dependencies. Integrating with convolutional neural networks (CNNs), we develop an end-to-end scene labeling system. Extensive experiments on three large-scale benchmarks demonstrate that the proposed approach can improve the baselines by large margins and outperform other state-of-the-art algorithms.

1. Introduction

Scene parsing or scene labeling, aiming to assign one of predefined labels to each pixel in an image, is usually formulated as a pixel-level classification problem. Inspired by the success of convolutional neural networks (CNNs) in image classification [22, 27, 45], CNNs have drawn increasing interests in scene labeling and demonstrated promising performance [3, 17, 19, 36, 39]. A potential issue, however, for CNN based methods is that only limited contextual cues from a local region (i.e., *receptive field*) in CNNs are explored for classification, which is prone to cause ambiguities for visually similar pixels of different categories. For example, the 'sand' pixels can be visually indistinguishable from 'road' pixels even for human with limited context. To alleviate this issue, a natural solution is to use rich context to discriminate locally ambiguous pixels [7, 34, 53, 57]. In these methods, nevertheless, the long-range contextual dependencies among different image regions are still not effectively explored, which are crucial in scene parsing.

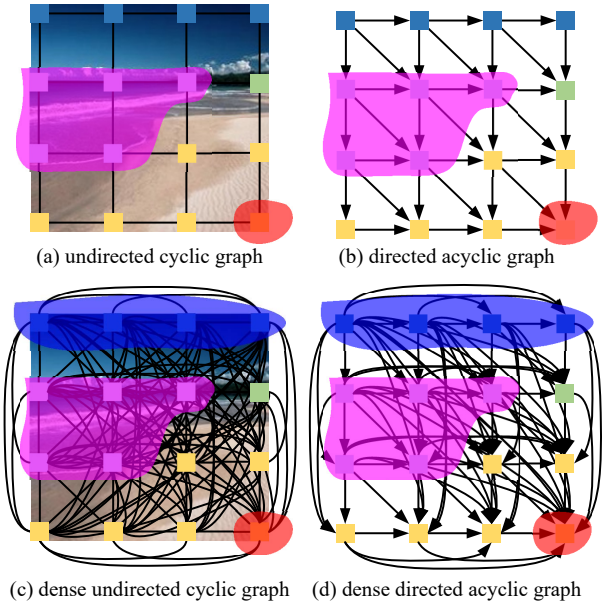


Figure 1. Image (a) shows the image of UCG structure as in [44], and (b) demonstrates one of four DAG decompositions. Unlike [44], we represent an image with dense UCG (D-UCG) as shown in (c), and (d) displays one of four dense DAGs (D-DAGs). Compared to plain UCG and DAG, our D-UCG and D-DAG capture richer dependencies in an image. Best viewed in color.

Motivated by the ability of capturing long-range dependency among sequential data, recurrent neural networks (RNNs) [14] have recently been employed to model semantic dependencies in images for scene labeling [6, 16, 29, 30, 43, 44, 49], allowing us to perform long-range inferences to discriminate ambiguous pixels.

To model the dependencies among image units, an effective way [44, 61] is to represent the image with an undirected cyclic graph (UCG) in which the image units are vertices and their interactions are encoded by undirected edges (see Fig. 1(a)). Due to the loopy structure of UCG, however, it is hard to directly apply RNNs to model dependencies in an image. To handle this problem, a UCG is approximated

978-1-7281-1975-5/19/$31.00 ©2019 IEEE
DOI 10.1109/WACV.2019.00198

with several directed acyclic graphs (DAGs) (see Fig. 1(b)). Then several DAG-structured RNNs are adopted to model the dependencies in these DAGs.

1.1. Motivation

Though these DAG-structured RNNs can model dependencies in images, some useful information may be discarded. For instance in Fig. 1(a), to correctly distinguish a 'sand' unit (marked in red region) from a 'road' one, DAG-structured RNNs can use the dependencies of 'water' units (marked in pink region) from its adjacent neighbors. However, the 'water' information may be decaying because it needs to pass through conductors (i.e., the adjacent neighbors of this 'sand' unit). Instead, a better way is to directly use dependencies from 'water' units to recognize the 'sand' unit. To such end, we propose dense RNNs to fully explore abundant dependencies in images for scene parsing.

Analogous to CNNs, DAG-structured RNNs can be unfolded to a feed-forward network where each vertex is a layer and a directed edge between two layers represents information flow (i.e., dependency relationship between two vertexes). The dependency information in an image flows from the first layer (i.e., the start vertex at top-left corner in Fig. 1(b)) to the last layer (i.e., the end vertex at bottom-right corner in Fig. 1(b)). Inspired by the superior performance of recently proposed DenseNet [23] in image recognition, which introduces dense connections among layers to improve information flow in CNNs, we propose to add more connections into the RNN feed-forward network as well (see Fig. 1(d)), to incorporate richer dependency information among image units.

Despite abundant dependencies from dense connections, we argue that not all dependencies are equally helpful to recognize a specific image region. For example in Fig. 1(d), the 'sky' units in blue region are not useful to distinguish the 'sand' unit in the red region from the 'road' unit. In contrast, the dependencies from 'water' units in the pink region are more crucial to infer its label. Therefore, more importance should be assigned to the dependencies from 'water' units, which motivates us to integrate an attention model into dense RNNs to select more useful dependencies.

1.2. Contribution

The **first contribution** of this work is the dense RNNs, which capture richer dependencies for image units from various abundant connections. Unlike previous approaches representing an image as a UCG, we formulate each image with a dense UCG (D-UCG), which is a complete graph. In D-UCG, each pair of vertexes are connected with an undirected edge (see Fig. 1(c)). By decomposing the D-UCG into several dense DAGs (D-DAGs), we propose the DAG-structured dense RNNs (DD-RNNs) to model dependencies in an image (see Fig. 1(d)). Compared with plain DAG-structured RNNs, our DD-RNNs can gain richer dependencies from various levels. For instance in Fig. 1(c), to correctly recognize the 'sand' unit in the red region, in addition to the dependencies from its neighbors, DD-RNNs enable the *firsthand* use of dependencies from 'water' units in the pink region to improve its discriminability.

Although DD-RNNs are capable of capturing vast dependencies through dense connections, for a specific image unit, certain dependencies are *irrelevant* to help improve discriminative power. To tackle this issue, we make the **second contribution** by introducing a novel attention model into DD-RNNs. The attention model is able to automatically select *relevant* and meanwhile restrain *irrelevant* dependency information for image units, further enhancing their discriminative power.

Last but not least, the **third contribution** is to implement an end-to-end labeling system based on our DD-RNNs. For validation, we test our method on three benchmarks: PASCAL Context [38], MIT ADE20K [59] and Cityscapes [11]. In these experiments the proposed approach significantly improves the baselines and outperforms other state-of-the-art methods.

2. Related Work

Scene parsing. Scene parsing has drawn extensive attentions in recent decades. Early efforts mainly focus on the graphical model with hand-crafted features [20, 33, 47, 52]. Despite great progress, these methods are restricted due to the use of hand-crafted features.

Inspired by the success in image recognition [22, 27, 45], CNNs have been extensively explored for scene parsing. Long *et al.* [36] propose a scene labeling method by transforming standard CNNs for classification into fully convolutional networks (FCN), resulting in significant performance gains. To generate desired full-resolution predictions, various methods are proposed to upsample low-resolution feature maps to high-resolution feature maps for final prediction [3, 31, 39]. In order to remit boundary problem in predictions, graphical models such as Conditional Random Field (CRF) or Markov Random Field (MRF) are introduced into CNNs [7, 35, 58]. As a pixel-level classification problem, contexts are crucial role in scene labeling to distinguish visually similar pixels of different categories. The work of [53] introduces the dilated convolution into CNNs to aggregate multi-scale context. Liu *et al.* [34] suggest an additional branch in CNNs to incorporate global context for scene parsing. In [57], Zhao *et al.* propose a spatial pyramid pooling module to fuse contexts from different levels, showing superior performance in scene parsing. Zhang *et al.* [55] introduce an context encoding module into CNNs to improve parsing performance.

RNNs on computer vision. With the capability of model-

ing spatial dependencies in images, RNNs [14] have been applied to many computer vision tasks such as image completion [40], handwriting recognition [21], image classification [61], visual tracking [15], skin detection [60] and so forth. Considering the importance of spatial dependencies in an image to distinguish ambiguous pixels, there are attempts to applying RNNs for scene labeling.

The work of [6] explores the two-dimensional long-short term memory (LSTM) networks for scene parsing by taking into account the spatial dependencies of pixels in images. Stollenga *et al.* [46] introduce a parallel multi-dimensional LSTM for image segmentation. Liang *et al.* [30] propose a graph based LSTM to model the dependencies among different superpixels. The work of [30] applies a local-global LSTM model on object parsing. Visin *et al.* [49] suggest to utilize multiple linearly structured RNNs to model horizontal and vertical dependencies among image units for scene labeling. Li *et al.* [29] extend this method by substituting RNNs with LSTM and apply it to RGB-D scene labeling. Qi [41] proposes the gated recurrent units (GRUs) to model long-range context. Especially, to exploit more spatial dependencies in images, Shuai *et al.* [44] propose to represent an image with a UCG. By decomposing UCG into several DAGs, they then propose to use DAG-structured RNNs to model dependencies among image units.

Attention model. The attention model, being successfully applied in Natural Language Processing (NLP) such as machine translation [4] and sentence summarization [42], has drawn increasing interest in computer vision. Xu *et al.* [51] propose to leverage an attention model to find out regions of interest in images which are relevant in generating next word. In [9], Chen *et al.* propose a scale attention model for semantic segmentation by adaptively merging outputs from different scales. In [1], the attention model is utilized to assign importance to different regions for context modeling in images. The work of [37] introduces a co-attention model to combine question and image features for question answering. Chu *et al.* [10] utilize attention model to fuse multi-context for human pose estimation.

Our approach. In this paper, we focus on how to effectively exploit abundant dependencies in images and introduce the dense RNNs module. Our approach is related to but different from previous RNN approaches (e.g., DAG-structured RNNs [44] and linearly structured RNNs [49] or LSTM [29]), in which each image unit only receives dependency information from its limited neighbors and considerable useful dependencies are thrown away. In contrast, we propose to add dense paths into RNNs to enable immediate long-range dependencies. Consequently, each image unit can directly 'see' dependencies in the whole image, leading to more discriminative representation. It is worth noting that the idea of dense connections can not only used for graphical RNNs [44] but also easily applied to other linearly structured RNNs [29, 49].

Furthermore, we introduce an attention model into dense RNNs. To the best of our knowledge, this work is the first to use attention mechanism in RNNs for scene parsing. Our attention model automatically selects relevant and restrains irrelevant dependencies for image units from dense connections, further improving their discriminabilities.

3. The Proposed Approach

3.1. Review of DAG-structured RNNs

The linear RNNs in [14] are designated to deal with sequential data related tasks. Specifically, a hidden unit 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}, and the output y_t is connected to the hidden unit h_t. Given an input sequence $\{x_t\}_{t=1,2,\cdots,T}$, the hidden unit and output at time step t can be computed with

$$h_t = \phi(Ux_t + Wh_{t-1} + b) \tag{1}$$

$$y_t = \sigma(Vh_t + c) \tag{2}$$

where U, V and W represent transformation matrices, b and c are bias terms, and $\phi(\cdot)$ and $\sigma(\cdot)$ are non-linear functions, respectively. Since the inputs $\{x_t\}_{t=1,2,\cdots,T}$ are progressively stored in the hidden layers as in Eq. (1), RNNs are able to preserve the memory of entire sequence and thus capture the long-range contextual dependencies.

For an image, the interactions among image units can be formulated as a graph in which the dependencies are forwarded through edges. The solution in [44] utilizes a standard UCG to represent an image (see again Fig. 1(a)). To break the loopy structure of UCG, [44] further proposes to decompose the UCG into four DAGs along different directions (see Fig. 1(b) for a southeast example).

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ denote the DAG as shown in Fig. 1(b), where $\mathcal{V} = \{v_i\}_{i=1}^{N}$ represents the vertex set of N vertexes, $\mathcal{E} = \{e_{ij}\}_{i,j=1}^{N}$ represents the edge set, and e_{ij} indicates a directed edge from v_i to v_j. A DAG-structured RNN resembles the identical topology of $\mathcal{G}$, with a forward pass formulated as traversing $\mathcal{G}$ from start vertex. In such modeling, the hidden layer of each vertex relies the hidden units of its adjacent predecessors (see Fig. 2(b)). For vertex v_i, its hidden layer h_{v_i} and output y_{v_i} are computed with

$$h_{v_i} = \phi(Ux_{v_i} + W \sum_{v_j \in \mathcal{P}_{\mathcal{G}}(v_i)} h_{v_j} + b) \tag{3}$$

$$y_{v_i} = \sigma(Vh_{v_i} + c) \tag{4}$$

where x_{v_i} denotes the local feature at vertex v_i and $\mathcal{P}_{\mathcal{G}}(v_i)$ represents the predecessor set of v_i in $\mathcal{G}$. By storing local inputs into hidden layers and progressive forwarding among them with Eq. (3), the discriminative power of each image unit is improved with dependencies from other units.

3.2. Dense RNNs

In DAG-structured RNNs, each image unit receives the dependencies from other units through recurrently forwarding information between adjacent units. Nevertheless, the useful dependency information may be potentially degraded after going through many conductors, resulting in a *dependency decaying* problem. For instance in Fig. 1(b), the most useful contextual cues from 'water' units have to pass through conductors to arrive at the 'sand' unit covered in the red region. A natural solution to remedy the problem of dependency decaying is to add additional paths between hidden layers of distant units and current image unit.

Inspired by the recently proposed DenseNet [23] that introduces dense connections into CNNs, we propose DAG-structured dense RNNs (DD-RNNs) to model richer dependencies in an image. We first view a DAG-structured RNNs as unfolded to get a feed-forward network, where the dependency information in an image flows from start to end vertexes. Then, to capture richer dependencies in images (e.g., forthright dependencies among *non-adjacent* units in Fig. 1(b)), we introduce more connections in the RNN feed-forward network, resulting in the proposed DD-RNNs.

To achieve dense connections, we represent each image with a dense UCG (D-UCG), which is equivalent to a complete graph (see Fig. 1(c) for illustration). Compared to standard UCG, D-UCG allows each image unit to connect with all of other units. Because of the loopy property of D-UCG, we adopt the strategy as in [44] to decompose the D-UCG to four D-DAGs along four directions. One of the four D-DAGs along the southeast direction is shown in Fig. 1(d).

Let $\mathcal{D}$ represent the D-DAG in Fig. 1(d). The structure of DD-RNNs resembles the identical topology of $\mathcal{D}$ as in Fig. 2(c). In DD-RNNs, the hidden layer of each vertex relies on the hidden units of all its *adjacent* and *non-adjacent* predecessors, which fundamentally differs from [44] in which the hidden unit of each vertex only relies on hidden units of its adjacent predecessors (see Fig. 2(b)). The forward pass at the vertex v_i in DD-RNNs is expressed as

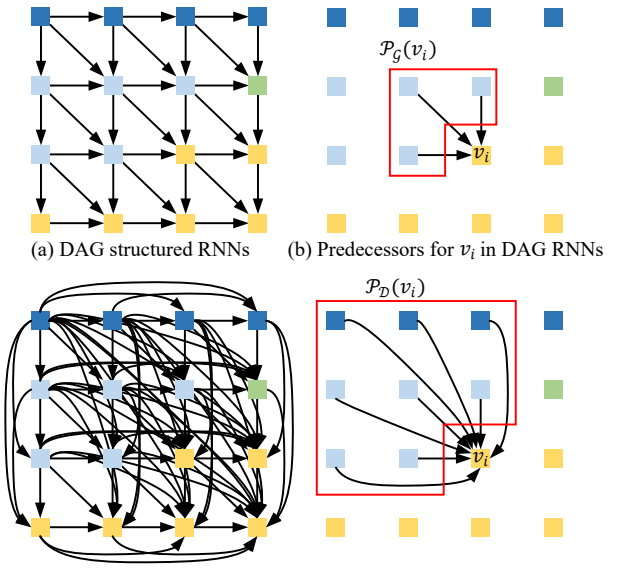
$$\hat{h}_{v_i} = \sum_{v_j \in \mathcal{P}_{\mathcal{D}}(v_i)} h_{v_j} \tag{5}$$

$$h_{v_i} = \phi(U x_{v_i} + W \hat{h}_{v_i} + b) \tag{6}$$

$$y_{v_i} = \sigma(V h_{v_i} + c) \tag{7}$$

where $\mathcal{P}_{\mathcal{D}}(v_i)$ is the dense predecessor set of v_i in D-DAG $\mathcal{D}$, and it contains both adjacent and non-adjacent predecessors (see Fig. 2(d)). Compared to the DAG-structured RNNs in [44], our DD-RNNs are able to capture richer dependencies in an image through various dense connections.

A concern arisen naturally from the dense model is the complexity. In fact, it is unrealistic to directly apply DD-RNN to pixels of an image. Fortunately, neither is it necessary. As described in Section 3.4, we apply DD-RNN to a high layer output of existing CNN models. Such strategy largely reduces the computational burden – as summarized in Table 7, our final system runs faster than many state-of-the-arts while achieving better labeling accuracies.



(a) DAG structured RNNs (b) Predecessors for v_i in DAG RNNs

(c) DAG structured dense RNNs (d) Predecessors for v_i in DD-RNNs

Figure 2. The illustration of difference between DAG-structured RNNs [44] and our DD-RNNs. Image (a) shows the DAG-structured RNNs along southeast direction, and in (b) the hidden layer of vertex v_i relies on its three adjacent predecessors (see the red region in (b)). Image (c) is our DD-RNNs, and in (d) the hidden layer of v_i is dependent on all its adjacent and non-adjacent predecessors (see the red region in (d)). Best viewed in color.

3.3. Attention model in DD-RNNs

For the hidden layer at vertex v_i, it receives various dependency information from predecessors through dense connections. However, the dependencies from different predecessors are not always equally helpful to improve the discriminative representation (see Fig. 2(d)). For example, to distinguish the 'sand' units from visually alike 'road' units in a beach scene, the most important contextual cues are probably the dependencies from 'water' units instead of other units such as 'sky' or 'tree'. In this case, we term the relation from 'water' units as relevant dependencies while the information from 'sky' or 'tree' units as irrelevant ones.

To encourage relevant dependencies and meanwhile restrain irrelevant ones for each image unit, we introduce a soft attention model [4] into DD-RNNs. In [4], the attention model is employed to softly assign importance to input words in a sentence when predicting a target word for machine translation. In this paper, we leverage attention model to select more relevant and useful dependencies for each image unit. To this end, we do not directly use Eq. (5) and (6) to model the relationships between h_{v_i} and its predecessors.

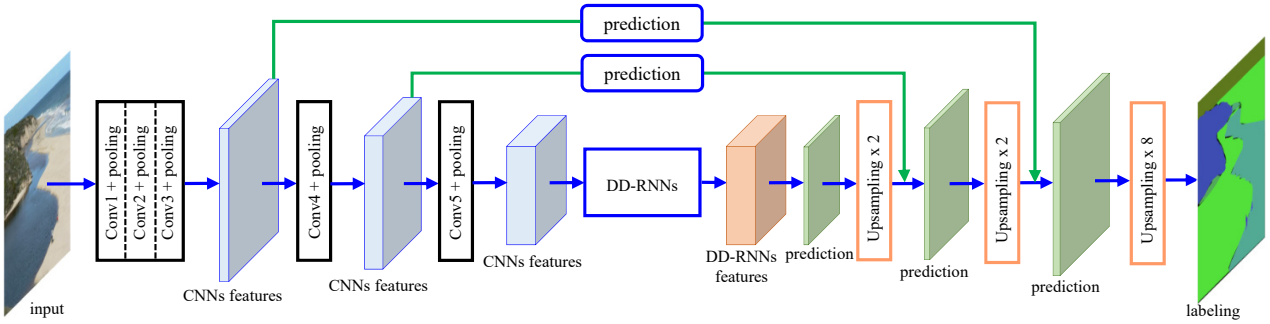


Figure 3. The architecture of our full system. The DD-RNNs are placed on the top of feature maps obtained from the last convolutional block to model long-range dependencies in an image, and the deconvolution is used to upsample the predictions. Low-level and high-level features are combined through skip strategy for final labeling (see the green arrows). Best viewed in color.

Instead, we employ the following expression to model the dependency between h_{v_i} and one of its predecessors h_{v_j}

$$h_{v_i,v_j} = \phi(Ux_{v_i} + Wh_{v_j} + b) \tag{8}$$

where h_{v_j} represents the hidden layer of a predecessor $v_j \in \mathcal{P}_{\mathcal{D}}(v_i)$ of v_i. The h_{v_i,v_j} in Eq. (8) models dependency information from h_{v_j} for h_{v_i}. The final hidden unit h_{v_i} at v_i is obtained by summarizing all h_{v_i,v_j} with attentional weights, as computed by

$$h_{v_i} = \sum_{v_j \in \mathcal{P}_{\mathcal{D}}(v_i)} h_{v_i,v_j} w_{v_i,v_j} \tag{9}$$

where the attention weight w_{v_i,v_j} for h_{v_j} reflects the relevance of the predecessor v_j to v_i, calculated by

$$w_{v_i,v_j} = \frac{\exp(z^{\mathrm{T}} h_{v_i,v_j})}{\sum_{v_k \in \mathcal{P}_{\mathcal{D}}(v_i)} \exp(z^{\mathrm{T}} h_{v_i,v_k})} \tag{10}$$

where z^{T} represents a transformation matrix.

With the above attention model, we replace Eq. (5) and (6) with Eq. (8) and (9) for a forward pass at v_i in DD-RNNs. By using stochastic gradient descent (SGD) method, the attentional DD-RNNs can be trained in an end-to-end manner.

3.4. Full labeling system

Before showing the full labeling system, we first introduce the decomposition of D-UCG. As in [44], we decompose the D-UCG $\mathcal{U}$ into a set of D-DAGs represented with $\{\mathcal{D}^l\}_{l=1}^{L}$, where L is the number of D-DAGs. Since Equation (9) only computes the hidden layer at vertex v_i in one of L D-DAGs, the final output $\hat{y}_{v_i}$ at v_i is derived by aggregating the hidden layers at v_i from all D-DAGs. The mathematical formulation for this process is expressed as

$$h^l_{v_i,v_j} = \phi(U^l x_{v_i} + W^l h^l_{v_j} + b_l) \tag{11}$$

$$h^l_{v_i} = \sum_{v_j \in \mathcal{P}_{\mathcal{D}^l}(v_i)} h^l_{v_i,v_j} w^l_{v_i,v_j} \tag{12}$$

$$\hat{y}_{v_i} = \sigma(\sum\nolimits_{l=1}^{L} V^l h^l_{v_i} + c) \tag{13}$$

With the equations above, the proposed DD-RNNs can be used to capture abundant dependencies among image units.

We develop an end-to-end scene labeling system by integrating our approach with CNNs for scene parsing as shown in Fig. 3. The proposed DD-RNNs are placed on the top of feature maps obtained after the last convolutional block to model long-range dependencies in the input image, and the deconvolution operations are used to upsample the predictions. To produce the desired input size of labeling result, we utilize the deconvolution [54] to upsample predictions. Taking into account both spatial and semantic information for scene labeling, we adopt the skip strategy [36] to combine low-level and high-level features. The whole system is trained end-to-end with the pixel-wise cross-entropy loss.

4. Experimental Results

Implementation details. In order validate the effectiveness of the proposed DD-RNNs, we develop two labeling systems by integrating our DD-RNNs with two different architectures: the VGG-16 [45] and the ResNet-101 [22]. The DD-RNNs are employed to model dependencies among image units in output of the last convolutional block (Fig. 3). The network takes 512×512 images as inputs, and outputs the labeling results with the same resolution. When evaluating, the labeling results are resized to the original input size. The dimension of input, hidden and output units for DD-RNNs is set to 512. The two non-linear activations ϕ and σ are *ReLU* and *softmax* functions, respectively. The full networks are end-to-end trained with standard SGD method. For convolutional blocks, the learning rate is initialized to

Table 1. Baseline comparisons of mIoU (%) with different backbones on PASCAL Context [38], MIT ADE20K (validation set) [59] and Cityscapes (validation set) [11].

	PASCAL Context [38]		MIT ADE20K [59]		Cityscapes [11]	
	VGG-16	ResNet-101	VGG-16	ResNet-101	VGG-16	ResNet-101
Baseline FCN	35.6	40.3	28.7	35.1	64.7	68.9
FCN+CRF	40.1	43.8	30.8	36.2	66.7	69.2
FCN+DAG-RNN	41.3	45.1	32.1	37.4	70.2	75.5
FCN+DD-RNN	**44.9**	**49.3**	**35.7**	**40.9**	**72.3**	**78.2**

be 10^{-4} and decays exponentially with the rate of 0.9 after 10 epochs. For D-RNNs, the learning rate is initialized to be 10^{-2} and decays exponentially with the rate of 0.9 after 10 epochs. The batch sizes for both training and testing phases are set to 1. The results are reported after 50 training epochs. The networks are implemented in Matlab using MatConvNet [48] on a single Nvidia GeForce TITAN GPU with 12GB memory.

Datasets. We test our method on the large-scale PASCAL Context [38], MIT ADE20K [59] and Cityscapes [11].

The PASCAL Context contains 10,103 images annotated into 540 classes, where 4,998 images are used for training and the rest for testing. Similar to other literatures, we only consider the most frequent 59 classes for evaluation.

The recent MIT ADE20K consists of 20,000 images in training set and 2,000 images in validation set. There are total 150 semantics classes in the dataset.

The Cityscapes contains 5000 images of street traffic scene, where 2975 images are used for training, 500 images for validation, and the rest for testing. In total, 19 classes are considered for training and evaluation.

Evaluation metrics. As in [36], we utilize mean Intersection over Union (mIoU%) for evaluation.

4.1. Baseline comparisons

To better analyze our method, we develop several baselines to prove its effectiveness:

Baseline FCN is implemented by removing our attentional DD-RNNs from networks. Note that the baseline FCN differs from FCN-8s [36] because we discard two fully connected layers. Other settings remain the same as in FCN-8s [36].

FCN+CRF is implemented by applying CRF [26] to perform post-processing on the results of baseline FCN.

FCN+DAG-RNN is implemented by substituting the attentional DD-RNNs with *plain* DAG-RNN. Note that FCN+DAG-RNN varies from [44] because we do not use class weighting strategy and larger conventional kernel in our labeling system.

FCN+DD-RNNs represents the proposed scene labeling method.

Table 1 shows the quantitative results between different baselines and our approach with two backbones. All CRF, DAG-RNN and DD-RNNs can improve the performance of baseline FCN. More specific, our method obtains mIoU gains of 9.3%, 7.0% and 7.6% with VGG-16 and of 9.0%, 5.8% and 9.3% with ResNet-101 on three datasets, and outperforms other two baselines using CRF and DAG-RNN.

4.2. Comparison results on PASCAL Context

Table 2. Quantitative comparisons on PASCAL Context [38] (59 classes).

Algorithm	Backbone	mIoU (%)
CAMN [1]	VGG-16	41.2
PixelNet [5]	VGG-16	41.4
FCN-8s [36]	VGG-16	38.2
HO-CRF [2]	VGG-16	41.3
BoxSup [12]	VGG-16	40.5
ParseNet [34]	VGG-16	40.4
ConvPP-8 [50]	VGG-16	41.0
CNN-CRF [32]	VGG-16	43.3
CRF-RNN [58]	VGG-16	39.3
DAG-RNN [44]	VGG-16	42.6
DAG-RNN-CRF [44]	VGG-16	43.7
DeepLab v2-CRF [8]	ResNet-101	44.4
GCE [24]	ResNet-101	46.5
RefineNet [31]	ResNet-101	47.1
DD-RNNs	VGG-16	44.9
DD-RNNs	ResNet-101	**49.3**

The quantitative comparisons to state-of-the-art methods are summarized in Table 2. Benefiting from deep CNNs, FCN-8s [36] achieves promising result with mIoU of 38.2%. In order to alleviate boundary issue in FCN-8s, CRF-RNN [58] and DeepLab v2-CRF [8] use probabilistic graphical model such as CRF in CNNs, and obtain mIoUs of 39.3% and 39.6%, respectively. Other approaches such as CAMN [1], ParseNet [34] and GCE [24] suggest to improve performance by incorporating global contextual information and obtain mIoUs of 41.2%, 40.4% and 46.5%. Despite improvements, these methods ignore long-range dependencies in images, which are crucial for inferring ambiguous pixels. The method in [44] employs RNNs to capture contextual dependencies among image units for scene labeling and shows outstanding performance with mIoU of 42.6% with VGG-16. Moreover, they use CRF to improve

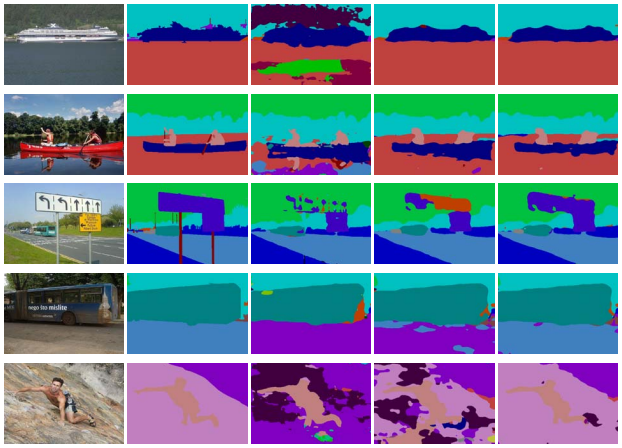


Figure 4. Qualitative labeling results with VGG-16 on the PASCAL Context [38]. Best viewed in color.

the result to 43.7%. Different from [44], we propose DD-RNNs to capture richer dependencies. Without any class weighting strategy and post-processing, our DD-RNNs with VGG-16 obtain the mIoU of 44.9%, which outperforms the method in [44] by 1.2%, showing the advantage of DD-RNNs. With deeper ResNet-101, we achieve the mIoU of 49.3%, outperforming the state-of-the-art RefineNet [31].

Fig. 4 shows qualitative results obtained with VGG-16 on PASCAL Context [38]. Without considering long-range contextual dependencies in images, FCN-8s [36] is prone to cause misclassification (see the third column in Fig. 4). Our baseline can help alleviate this situation using RNNs to capture dependencies in images. For example, in the first two rows in Fig. 4, the 'water' can be correctly recognized with the dependencies from 'boat'. However, the plain RNNs fail in more complex scenes (see the last three rows in Fig. 4). For example, in the fourth row in Fig. 4, most of 'road' pixels are mistakenly classified into 'ground' pixels without full use of dependencies from 'bus'. By contrast, the proposed DD-RNNs are capable of recognizing most of 'road' pixels by taking advantages of richer dependencies from 'bus' in images.

4.3. Comparison results on MIT ADE20K

Table 3 summarizes the quantitative results and comparisons to other algorithms. The FCN-8s [36] achieves the mIoU of 29.4%. To incorporate multi-scale contexts, [53] proposes the dilated convolution and improves the mIoU to 32.3%. To same end, Hung *et al.* [24] embed global context into CNNs to obtain improvements, and improve the performance to 38.4% with ResNet-101. Though the aforementioned methods take the global context of image into account, they ignore long-range contextual dependencies in images. In this work, we employ DD-RNNs to model this dependency information for scene labeling. In specific, we obtain the mIoU of 35.7% with VGG-16, and achieve better performance with mIoU of 40.9% when using ResNet-101 as backbone.

Table 3. Quantitative comparisons on MIT ADE20K (validation set) [59].

Algorithm	Backbone	mIoU (%)
SegNet [3]	VGG-16	21.6
FCN-8s [36]	VGG-16	29.4
DilatedNet [53]	VGG-16	32.3
Cascade-SegNet [59]	VGG-16	27.5
Cascade-Dilated [59]	VGG-16	34.9
GCE [24]	ResNet-101	38.4
RefineNet [31]	ResNet-101	40.2
DD-RNNs	VGG-16	35.7
DD-RNNs	ResNet-101	**40.9**

4.4. Comparison results on Cityscapes

Table 4 summarizes the quantitative comparison results with state-of-the-art approaches on Cityscapes [11]. Since the resolution of image is too large, we divide each image into multiple patches. After obtaining the parsing result of each patch, we combine them to derive the labeling of original image. Among the compared algorithms, FCN-8s [36] achieves the mIoU of 65.3%. Liu *et al.* [35] adopt Markov Random Field (MRF) to model high-order CNNs and obtain a mIoU of 66.8%. The approach of [32] utilizes CRF to capture contextual information for scene parsing and improves the mIoU to 71.6%. DeepLabv2 [8] combines both CRF and atrous convolution to incorporate more contexts and achieves a mIoU of 70.4%. In this work, we propose dense RNNs to capture richer dependencies from the whole image for each image unit. With the ResNet backbone, we achieve the mIoU of 78.2%, outperforming other context aggregation methods.

Table 4. Quantitative comparisons on Cityscapes (test set) [11].

Algorithm	Backbone	mIoU (%)
SegNet [3]	VGG-16	57.0
FCN-8s [36]	VGG-16	65.3
DPN [35]	VGG-16	66.8
LRR-4x [18]	VGG-16	71.8
CNN-CRF [32]	VGG-16	71.6
DilatedNet [53]	VGG-16	67.1
DeepLab v2-CRF [8]	ResNet-101	70.4
LC [28]	ResNet-101	71.1
RefineNet [31]	ResNet-101	73.6
PEARL [25]	ResNet-101	74.9
SAC [56]	ResNet-101	78.1
DD-RNNs	VGG-16	72.3
DD-RNNs	ResNet-101	**78.2**

Table 5. Analysis of mIoU (%) with and without attention model in DD-RNNs using VGG-16.

	DD-RNNs w/o attention model	DD-RNNs w/ attention model
PASCAL Context	44.3	**44.9**
MIT ADE20K	34.5	**35.7**
Cityscapes	72.0	**72.3**

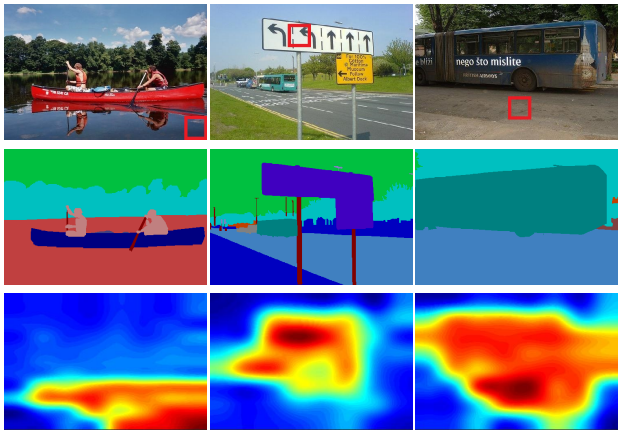


Figure 5. Visualization of the learned attentional weight map for a specific region (marked in red rectangle in the first row). First row: input image. Second row: groundtruth. Third row: attentional weight map. Best viewed in color.

4.5. Ablation study on attention model

In this paper, we propose the DD-RNNs to model richer dependencies in images, which significantly enhances discriminability for each image unit. However, different dependencies are not always equally helpful. To activate relevant and restrain irrelevant dependencies, we introduce an attention model into DD-RNNs. To demonstrate the effectiveness of attention model, we conduct experiment by removing attention model from DD-RNNs. Note that in these two groups of experiments, the only difference is the attention model, while other settings (e.g., parameters for all other layers) are exactly the same. Table 5 summarizes the results on three benchmarks with VGG-16, and shows that the attention model helps to further improve performance.

In order to better understand the attention model, we show the learned attentional weight map for a specific region as shown in Fig. 5. From Fig. 5, we can see that relevant dependencies are enhanced while irrelevant information are restrained. For example in the first column, the most helpful contextual dependencies for the 'water' region come from its surrounding and the 'boat' instead of 'tree' or 'sky', and our attention model learns to pay more importance to the relevant dependencies (i.e., surrounding region and 'boat') in the weight map. In the second column in Fig. 5, to recognize 'sign' region, the useful information comes from surrounding and the 'bus', our attention model highlights these regions. In the third column, we can see that our model pays more attention to the relevant 'bus' dependencies to correctly recognize the 'road' region.

Table 6. Analysis of computation complexity and accuracy of DD-RNNs on PASCAL Context [38] dataset.

Algorithm	Inference	mIoU (%)
CFM [13]	0.57 s	34.4
CAMN [1]	0.27 s	41.2
FCN-8s [36]	0.32 s	38.2
ParseNet [34]	0.25 s	40.4
CRF-RNN [58]	0.70 s	39.3
DeepLab [7]	0.40 s	37.6
Baseline FCN (VGG-16)	0.17 s	35.6
Baseline DAG-RNN (VGG-16)	0.21 s	41.3
DD-RNNs (VGG-16)	0.28 s	44.9
DD-RNNs (ResNet-101)	0.36 s	49.3

4.6. Study on model complexity

As a practical application, both efficiency and accuracy are crucial for scene labeling. To better analyze the proposed approach, we demonstrate the inference time of one forward pass and accuracy on the PASCAL Context [38].

Table 6 reports the efficiency and accuracy of our baseline and other scene labeling algorithms. Compared to its baseline FCN (VGG-16), our algorithm DD-RNNs (VGG-16) obtains mIoU gain of 9.3% while the inference time only increase by 0.11s, showing the advantage of our DD-RNNs module. Moreover, when replacing the VGG-16 with ResNet-101 as our backbone, the mIoU is further improved to 49.3%. In comparison with approaches including CAMN [1], FCN-8s [36], CRF-RNN [58] and DeepLab [7], our method runs efficiently while achieving better accuracy.

5. Conclusion

This paper proposes dense RNNs for scene labeling. Unlike existing methods exploring limited dependencies, our DAG-structured dense RNNs (DD-RNNs) exploit abundant contextual dependencies through dense connections in an image, which better improves the discriminative power of image units. In addition, considering that different dependencies are not always equally helpful to recognize each image unit, we propose an attention model to assign more importance to relevant dependencies. Integrating with CNNs, we develop an end-to-end labeling system. Extensive experiments on PASCAL Context, MIT ADE20K and Cityscapes demonstrate that our DD-RNNs significantly improve the baselines and outperform other state-of-the-art algorithms, evidencing the effectiveness of proposed dense RNNs.

Acknowledgements. This work is supported in part by US NSF Grants 1350521, 1407156, and 1814745.

References

[1] A. H. Abdulnabi, B. Shuai, S. Winkler, and G. Wang. Episodic camn: Contextual attention-based memory networks with iterative feedback for scene labeling. In *CVPR*, 2017. 3, 6, 8

[2] A. Arnab, S. Jayasumana, S. Zheng, and P. H. Torr. Higher order conditional random fields in deep neural networks. In *ECCV*, 2016. 6

[3] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *TPAMI*, 39(12):2481–2495, 2017. 1, 2, 7

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015. 3, 4

[5] A. Bansal, X. Chen, B. Russell, A. G. Ramanan, et al. Pixelnet: Representation of the pixels, by the pixels, and for the pixels. In *CVPR*, 2017. 6

[6] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki. Scene labeling with lstm recurrent neural networks. In *CVPR*, 2015. 1, 3

[7] 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. 1, 2, 8

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2017. 6, 7

[9] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, 2016. 3

[10] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang. Multi-context attention for human pose estimation. In *CVPR*, 2017. 3

[11] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. 2, 6, 7

[12] J. Dai, K. He, and J. Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015. 6

[13] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In *CVPR*, 2015. 8

[14] J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990. 1, 3

[15] H. Fan and H. Ling. SANet: Structure-aware network for visual tracking. In *CVPRW*, 2017. 3

[16] H. Fan, X. Mei, D. Prokhorov, and H. Ling. Rgb-d scene labeling with multimodal recurrent neural networks. In *CVPRW*, 2017. 1

[17] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *TPAMI*, 35(8):1915–1929, 2013. 1

[18] G. Ghiasi and C. C. Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *ECCV*, 2016. 7

[19] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1

[20] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009. 2

[21] A. Graves and J. Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In *NIPS*, 2009. 3

[22] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2, 5

[23] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. In *CVPR*, 2017. 2, 4

[24] W.-C. Hung, Y.-H. Tsai, X. Shen, Z. Lin, K. Sunkavalli, X. Lu, and M.-H. Yang. Scene parsing with global context embedding. *ICCV*, 2017. 6, 7

[25] X. Jin, X. Li, H. Xiao, X. Shen, Z. Lin, J. Yang, Y. Chen, J. Dong, L. Liu, Z. Jie, et al. Video scene parsing with predictive feature learning. In *ICCV*, 2017. 7

[26] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011. 6

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 2

[28] X. Li, Z. Liu, P. Luo, C. C. Loy, and X. Tang. Not all pixels are equal: difficulty-aware semantic segmentation via deep layer cascade. In *CVPR*, 2017. 7

[29] 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. 1, 3

[30] X. Liang, X. Shen, D. Xiang, J. Feng, L. Lin, and S. Yan. Semantic object parsing with local-global long short-term memory. In *CVPR*, 2016. 1, 3

[31] G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks with identity mappings for high-resolution semantic segmentation. In *CVPR*, 2017. 2, 6, 7

[32] G. Lin, C. Shen, A. van den Hengel, and I. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, 2016. 6, 7

[33] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *TPAMI*, 33(5):978–994, 2011. 2

[34] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. In *ICLRW*, 2016. 1, 2, 6, 8

[35] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *ICCV*, 2015. 2, 7

[36] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1, 2, 5, 6, 7, 8

[37] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *NIPS*, 2016. 3

[38] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object

detection and semantic segmentation in the wild. In *CVPR*, 2014. 2, 6, 7, 8

[39] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015. 1, 2

[40] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *ICML*, 2016. 3

[41] G.-J. Qi. Hierarchically gated deep networks for semantic segmentation. In *CVPR*, 2016. 3

[42] A. M. Rush, S. Chopra, and J. Weston. A neural attention model for abstractive sentence summarization. In *EMNLP*, 2015. 3

[43] B. Shuai, Z. Zuo, B. Wang, and G. Wang. Dag-recurrent neural networks for scene labeling. In *CVPR*, 2016. 1

[44] B. Shuai, Z. Zuo, B. Wang, and G. Wang. Scene segmentation with dag-recurrent neural networks. *TPAMI*, 2017. 1, 3, 4, 5, 6, 7

[45] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1, 2, 5

[46] M. F. Stollenga, W. Byeon, M. Liwicki, and J. Schmidhuber. Parallel multi-dimensional lstm, with application to fast biomedical volumetric image segmentation. In *NIPS*, 2015. 3

[47] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *CVPR*, 2013. 2

[48] A. Vedaldi and K. Lenc. Matconvnet: Convolutional neural networks for matlab. In *ACM MM*, 2015. 6

[49] F. Visin, M. Ciccone, A. Romero, K. Kastner, K. Cho, Y. Bengio, M. Matteucci, and A. Courville. Reseg: A recurrent neural network-based model for semantic segmentation. In *CVPRW*, 2016. 1, 3

[50] S. Xie, X. Huang, and Z. Tu. Top-down learning for structured labeling with convolutional pseudoprior. In *ECCV*, 2016. 6

[51] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 3

[52] J. Yang, B. Price, S. Cohen, and M.-H. Yang. Context driven scene parsing with attention to rare classes. In *CVPR*, 2014. 2

[53] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 1, 2, 7

[54] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *ICCV*, 2011. 5

[55] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal. Context encoding for semantic segmentation. In *CVPR*, 2018. 2

[56] R. Zhang, S. Tang, Y. Zhang, J. Li, and S. Yan. Scale-adaptive convolutions for scene parsing. In *ICCV*, 2017. 7

[57] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1, 2

[58] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 2, 6, 8

[59] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 2, 6, 7

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

[61] 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. 1, 3