3-2021

Fast scene labeling via structural inference

Huaidong ZHANG

Chu HAN

Xiaodan ZHANG

Yong DU

Xuemiao XU

See next page for additional authors

Citation

ZHANG, Huaidong; HAN, Chu; ZHANG, Xiaodan; DU, Yong; XU, Xuemiao; HAN, Guoqiang; QIN, Jing; and Shengfeng HE. Fast scene labeling via structural inference. (2021). *Neurocomputing*. 442, 317-326.
Available at: https://ink.library.smu.edu.sg/sis_research/7838

Author

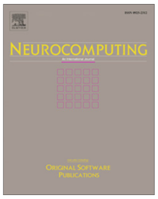
Huaidong ZHANG, Chu HAN, Xiaodan ZHANG, Yong DU, Xuemiao XU, Guoqiang HAN, Jing QIN, and Shengfeng HE

Neurocomputing 442 (2021) 317–326

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Fast scene labeling via structural inference

Huaidong Zhang [a], Chu Han [b], Xiaodan Zhang [c], Yong Du [d], Xuemiao Xu [a,*], Guoqiang Han [a], Jing Qin [e], Shengfeng He [a,*]

[a] School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
[b] Guangdong Provincial People's Hospital, Guangzhou, China
[c] Beijing University of Technology, Beijing, China
[d] Department of Computer Science and Technology, Ocean University of China, Qingdao, China
[e] Hong Kong Polytechnic University, Hongkong, China

ARTICLE INFO

Article history:
Received 24 March 2020
Revised 26 July 2020
Accepted 22 December 2020
Available online 10 March 2021
Communicated by Shijian Lu

Keywords:
LSTM
Structural inference
Scene labeling

ABSTRACT

Scene labeling or parsing aims to assign pixelwise semantic labels for an input image. Existing CNN-based models cannot leverage the label dependencies, while RNN-based models predict labels within the local context. In this paper, we propose a fast LSTM scene labeling network via structural inference. A minimum spanning tree is used to build the image structure for constructing semantic relationships. This structure allows efficient generation of direct parent–child dependencies for arbitrary levels of superpixels, and thus structural relationships can be learned with LSTM. In particular, we propose a bi-directional recurrent network to model the information flow along the parent–child path. In this way, the recurrent units in both coarse and fine levels can mutually transfer the global and local context information in the entire image structure. The proposed network is extremely fast, and it is 2.5× faster than the state-of-the-art RNN-based models. Extensive expseriments demonstrate that the proposed method provides a significant improvement in learning the label dependencies, and it outperforms state-of-the-art methods on different benchmarks.

1. Introduction

Scene labeling or parsing is a fundamental computer vision task which aims to discover pixel-level semantic information of the scene. It demands not only high accuracy labeling result but also restricted time performance in the practical applications such as autonomous driving or surveillance. Most of the existing works [1–11] predict labels by using the feed-forward, end-to-end CNN models. However, the convolutional operation is only capable to capture local context. It cannot model the dependencies and interactions across semantic layouts when there exist interactions among semantic layouts. That will lead to ambiguity when deciding the class of an object. For example in Fig. 1(a), there is a human hand wearing a glove on the table while holding a spoon. To decide whether it is a human hand or a cloth highly depends on the surrounding objects. In addition, this kind of intersection is quite common in the real world scenario. To introduce semantic dependency into the prediction, a solution with clearer structural information is desired.

To inject global context, a straightforward solution is to build a LSTM [12] model on neighboring pixels [13,14] or regular girds [15–18]. These methods can easily model the information flow horizontally and vertically, as shown in Fig. 2(a). However, the recurrent units of these regular LSTM models only perform over a small number of neighboring pixels, which fail to learn the structural information. Liang et al. [19] propose to model information transfer by constructing LSTM over superpixels [20]. They transfer the structural information within a graph-structured LSTM, as shown in Fig. 2(b). Comparing to regular LSTM, Graph-LSTM involves a much larger context into the semantic layouts. However, the number of superpixels must be carefully determined to avoid over-segmentation or under-segmentation problems. To overcome this problem, Peng et al. [21] and Liang et al. [22] propose to gradually evolve superpixel structure according to the semantic layouts. However, the structural relationship among superpixels is still not fully explored, and they cannot leverage the information from different granularities of superpixels.

To perform scene labeling with higher accuracy and much clearer boundaries, we want our model to fully explore both global and local contextual information. Instead of constructing a recurrent network directly on the generated superpixels, we proposed

[*] Corresponding authors.
E-mail addresses: xuemx@scut.edu.cn (X. Xu), hesfe@scut.edu.cn (S. He).

https://doi.org/10.1016/j.neucom.2020.12.134
0925-2312/© 2021 Elsevier B.V. All rights reserved.

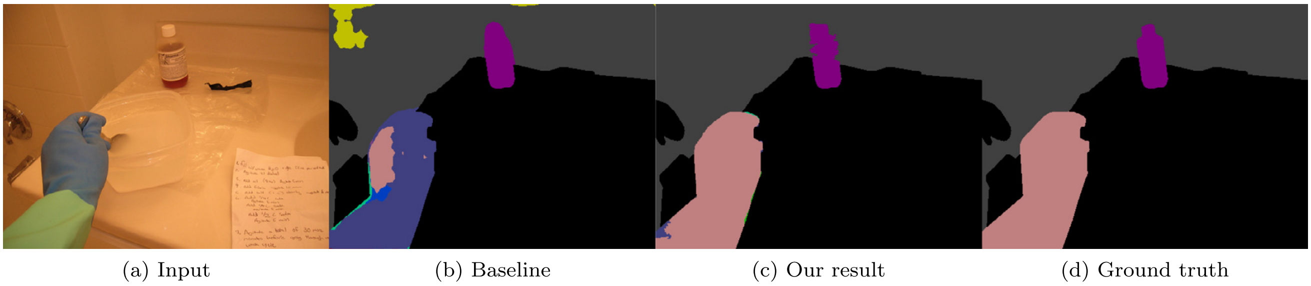

(a) Input (b) Baseline (c) Our result (d) Ground truth

Fig. 1. Ambiguity in real world scenario. It is hard to decide whether this is a human hand or a cloth for existing FCN-based methods since they lack semantic dependency. Our proposed Hier-LSTM can successfully label the human hand correctly with clear boundary by learned structural inference.

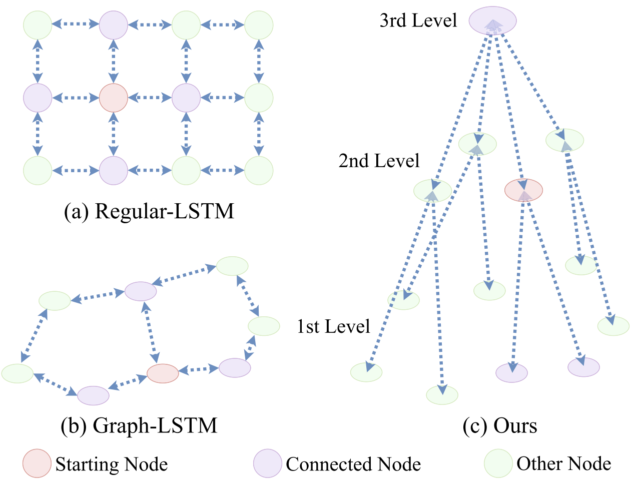


Fig. 2. Comparison of different LSTM architectures. Our method can learn structural dependencies with the proposed hierarchical superpixel architecture. Red nodes: the starting nodes. Purple nodes: nodes connected with the starting nodes that used for inferencing..

a Hierarchical LSTM (Hier-LSTM) to model the structural dependencies of superpixels (Fig. 2(d)). To achieve efficient performance, we tailor a fast superpixel generation method. When generating the superpixels, we record the merging orders of superpixels and form a superpixel tree. This tree contains all the parent–child relationships of the multiscale superpixels. Our proposed Hier-LSTM is designed as a bi-directional recurrent network to learn the structural relationships of different levels in the superpixel tree. Based on the directions of tree traversal, our training process can be divided into the top-down and bottom-up directions. In the bottom-up direction, the recurrent units in the lower levels (child nodes) pass the local information to the recurrent units in the higher levels (parent nodes). On the contrary, in the top-down direction, global context information passes throughout the tree to the lower levels. The bi-directional recurrent network allows us to learn the hierarchical image features and provide interactions between semantic layouts with different scales. Moreover, our proposed fast superpixel generation method boosts our whole system to efficient performance. Extensive experiments demonstrate that the proposed method performs favorably against state-of-the-art methods on different standard datasets. In particular, it is 10 × faster than the other RNN-based methods. Our contributions can be summarized as followed:

- We propose a fast superpixel generation method. This method can record the merging order of superpixels as a superpixel tree which contains all the relationships across different levels of superpixels.
- We propose a bi-directional recurrent network to aggregate both local and global image structures. This network is able to explore the interactions between semantic layouts in different scales, which enables hierarchical features learning and size-aware semantic prediction.
- The proposed scene labeling method outperforms state-of-the-art methods on different benchmarks. To the best of our knowledge, it is the fastest RNN-based network for scene labeling.

2. Related work

2.1. Scene labeling

Fully Convolutional Networks (FCNs) based models [2–6] can be separated to an encoder and a decoder. The encoder is applied for encoding the global information, while the decoder recovers the spatial feature for pixel-level prediction. These methods play a trade-off between abstract image features and local information. Therefore they mainly focus on overcome this problem, i.e., skip connection in [3,4], pyramid pooling module in [5], atrous convolution in [6] and etc. However, these FCN models obtain contextual information using only regular convolution, which makes them difficult to model complex and non-regular structural dependency. That will lead to ambiguity and unclear boundaries. We overcome this issue by modeling the information flow utilizing our proposed Hier-LSTM which is able to learn the dependency through structural inference.

2.2. Long Short-Term Memory Network

Previous LSTM models mainly explore the dependencies between LSTM units within a given superpixel map. Some methods [13–18,23] adopt the LSTM over pixel or grid to model information transfer. The directions of transfer in these methods are intuitive, but the boundaries of observed units lack semantic reasoning. Liang et al. [19] model the information transfer over superpixels, and make a great improvement on object parsing thanks to the semantic boundaries of superpixels. Peng et al. and Liang et al. [21,22] further extend the architecture of [19] to multi-level, and the information transfer on a different level of superpixels can be formulated at the same time. However, the above methods still not fully explore both local and global contextual information due to the LSTM design. Furthermore, the connections of LSTM units in their works are decided by whether two units are neighborhoods or not, which may cause unreasonable connections when the split of superpixels is mess.

Our proposed Hier-LSTM learn structural reasoning along the construction of superpixels. The bi-directional training strategy allows local and global information transfer more effective. Existing methods cannot achieve this by only perform LSTM on the same level of superpixel map.

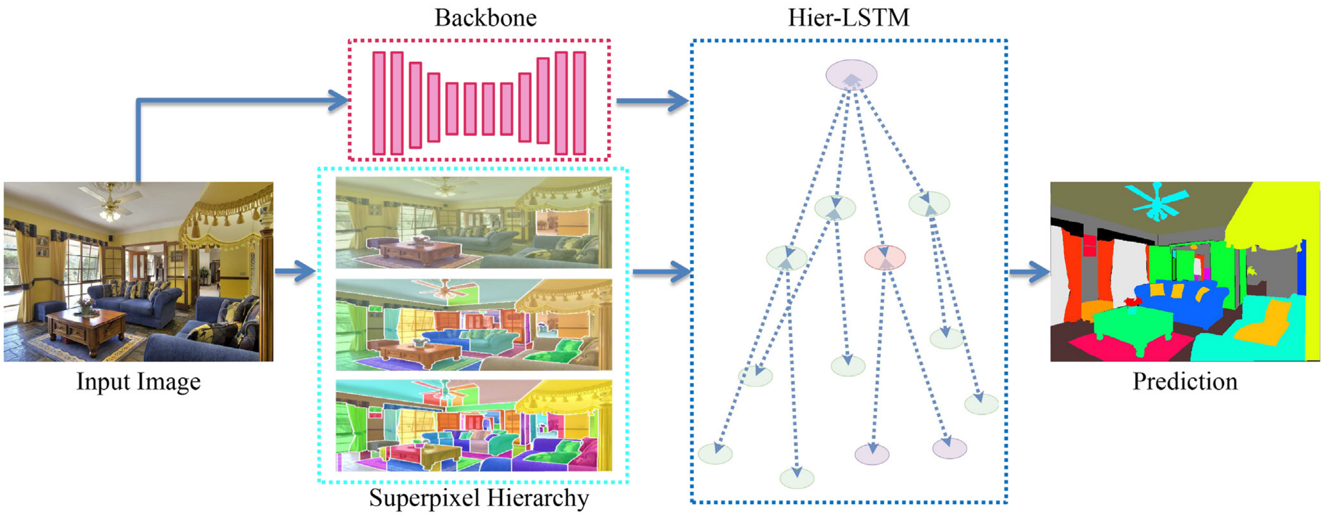


Fig. 3. System overview. Our proposed superpixel generation method not only generates but also records the whole generation process into a tree structure which provides comprehensive local and global information for the proposed Hier-LSTM.

3. Approach

In this paper, we propose a fast scene labeling method by learning the structural inference, as shown in Fig. 3. The input image is passed into a CNN backbone to extract the semantical features. A hierarchical superpixel tree is conducted by our designed fast superpixel generation method in Section 3.1. A Hier-LSTM with bi-directional training strategy is proposed in Section 3.2 to analyze the structural inference inside the superpixel hierarchy. The complete network well balances the tradeoff between accuracy and computational time, and achieves efficient performance.

3.1. Hierarchical superpixel generation

Most of the existing methods only consider the information within one single level superpixel map. But we find that there exists implicit structure information during the superpixel generation. Thus, we proposed to make use of the structure inference conveyed while generating superpixels and utilized a recurrent network to explore it. To this end, in this paper, we design a hierarchical superpixel generation method and conduct a multi-level superpixel maps for scene labeling. Comparing with the single level superpixel map, the multi-level one is able to provide more comprehensive local details and global structure. Besides, we believe that the merging process of superpixels also conveys cross-level structure information. Thus, we record all the merging steps using a tree structure to build up the parent–child relationship of each superpixel as shown in Fig. 4.

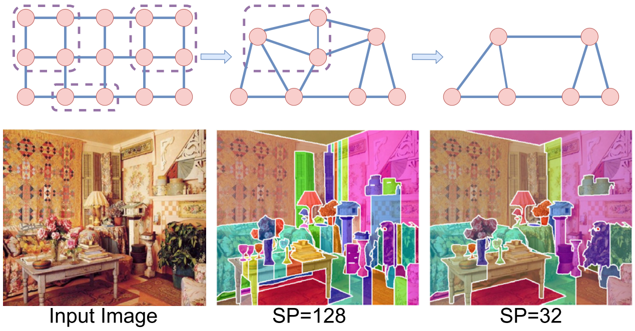


Fig. 4. Multi-level superpixel generation. The upper part of this figure illustrations the multi-level superpixel generation at each level. The lower part is the visualization of superpixel generation in real case. Note that, SP = 128 indicates the number of superpixels in the image is 128, and so forth.

3.1.1. Notation and definition

Formally, we conduct multi-level superpixel maps $\mathscr{G}$ in Eq. (1),

$$\mathscr{G} = \{\mathscr{G}_k | k = 1, 2 \ldots K\} \mathscr{G}_k = \{p_k^n | n = 1, 2 \ldots N_k\} \tag{1}$$

where $\mathscr{G}_k$ denotes the k-th level of superpixel map and K is the number of levels. Each level of superpixel map $\mathscr{G}_k$ contains N_k superpixels of the whole image under the level k. p_k^n is the n-th superpixel in $\mathscr{G}_k$. To build a multi-level observation, we keep increasing the level k which indicates the merging process of superpixels. Thus, the number of superpixels N_k will decrease when k increases. In our paper, we set $K = 8$ in practice.

During the superpixel generation, we conduct a superpixel tree to represent the structural relationship of superpixels between the adjacent two levels. Each superpixel p_k^n in the k-th level can be regarded as one node in the tree. As defined in Eqs. (2),

$$\begin{aligned} \mathscr{P}(p_k^n) &= \{p_{k+1}^m\}, \text{ when } k < K \\ \mathscr{C}(p_k^n) &= \{p_{k-1}^m | m = 1, 2, \ldots, M\}, \text{ when } k > 1 \end{aligned} \tag{2}$$

$\mathscr{P}(\cdot)$ and $\mathscr{C}(\cdot)$ denote the set of parent and child nodes of superpixel p_k^n respectively. Note that, each node can have more than one child node but only one parent node.

3.1.2. Superpixel generation

We propose a multi-level superpixel generation method, which is tailor-made for our Hier-LSTM. To fulfill the demand of efficient performance, we solve this problem by forming the image as a undirected graph shown in Fig. 4. Our method initially considers each pixel as one node, and then assigns edge weights between the node and its 8-connected neighborhoods. Then our algorithm iteratively merges the nodes and updates the edge weights. In each iteration, two nodes with minimum edge weight are merged into a new node and it will inherit the edges of the elder two nodes. We keep iteratively merging the nodes and updating the weights until only one node left. The design of this algorithm is similar to the minimum spanning trees (MST) algorithm [24], which can achieve linear time complexity.

Here, we define the edge weight $\mathscr{E}$ between two nodes as the merging cost in Eq. 3.

$$\mathscr{E} = \omega \mathscr{L}_{gradient} + \frac{1}{\omega}\mathscr{L}_{color} \tag{3}$$

$\mathscr{L}_{gradient}$ denotes the boundary cost, which is presented by the average edge confidence [25] between two nodes. $\mathscr{L}_{color}$ measures the absolute difference of the mean color. When the size of superpixel increases, the content of the superpixel will become busier. Therefore, the mean color will be less informative to present busy content in superpixels. So we introduce a weight ω, which equals to the size of the superpixel, to balance the importance of the color and gradient difference in terms of the superpixel size. For efficient performance, we apply a fast edge detection algorithm [25] to detect edge maps for gradient loss.

With the proposed superpixel generation method, all of the parent–child relationships ($\mathscr{P}(\cdot)$ and $\mathscr{C}(\cdot)$) of one node can be recorded concurrently on the fly. Thanks to the simply design of our superpixel generation method, multi-level superpixel maps are generated efficiently. The time statistics are shown in the experiment. Besides, the iteratively merging process can lead to many-to-one relation across levels, and thus the topologies of the structure are satisfied that each node can have more than one child node but only one parent node. The lower part of Fig. 4 shows some examples of our superpixel results in three different levels.

3.2. Hierarchical LSTM

Given the multi-level superpixel maps, we aims to learn the structural inference inside them. Instead of learning the structure information between two adjacent levels, we want to dig up more implicit structure information hiding in the longer range levels. Thus, we proposed a Hierarchical LSTM to solve this problem.

3.2.1. Basic LSTM formulation

Given the number of layers T and the input sequence $x = \{x_1, \dots, x_T\}$, a Long Short-Term Memory (LSTM) [12] network computes the hidden vector sequence $h = \{h_1, \dots, h_T\}$ as well as the cell vector sequence $c = \{c_1, \dots, c_T\}$ by iterating the same calculation from $t = 1$ to T. Comparing to standard RNN architecture, LSTM leverages memory cells to store the information over the arbitrary length time intervals. It is benefit to exploit long range relationship and structural inference. The basic LSTM [26] is defined as follows:

$$\begin{aligned}
i_t &= \sigma(W_{x_i}x_t + W_{h_i}h_{t-1} + W_{c_i}c_{t-1} + b_i),\\
f_t &= \sigma\left(W_{x_f}x_t + W_{h_f}h_{t-1} + W_{c_f}c_{t-1} + b_f\right),\\
c_t &= i_t \tanh(W_{x_c}x_t + W_{h_c}h_{t-1} + b_c) + f_t c_{t-1},\\
o_t &= \sigma(W_{x_o}x_t + W_{h_o}h_{t-1} + b_o),\\
h_t &= o_t \tanh(c_t).
\end{aligned} \tag{4}$$

where σ is the logistic sigmoid function, and i, f, o and c are the input gate, forget gate, output gate and cell activation vector respectively. The size of them are the same with the hidden vector h. In our framework, we first extract the image feature using CNN backbone. For each superpixel, we take a mean pooling operation and obtain its feature vector. Then the feature vectors of superpixels are the input sequence $x = \{x_1, \dots, x_T\}$ of LSTM.

3.2.2. Hier-LSTM

Since we have already recorded the merging process using a tree structure during the superpixel generation. To leverage the structural inference in multi-level superpixels and obtain higher accuracy and clearer boundaries, we reformulate the basic LSTM to a Hierarchical LSTM which is defined as follows:

$$\begin{aligned}
i_t &= \sigma(W_{x_i}x_t + W_{h_i}h_{t-1} + W_{c_i}c_{t-1} + b_i),\\
f_t &= \sigma\left(W_{x_f}x_t + W_{h_f}h_{t-1} + W_{c_f}c_{t-1} + b_f\right),\\
\overrightarrow{f}_t &= \sigma\left(W_{x_f}x_t + W_{\overrightarrow{h}_f}\overrightarrow{h}_{t-1} + W_{\overrightarrow{c}_f}\overrightarrow{c}_{t-1} + b_f\right),\\
\overleftarrow{f}_t &= \sigma\left(W_{x_f}x_t + W_{\overleftarrow{h}_f}\overleftarrow{h}_{t-1} + W_{\overleftarrow{c}_f}\overleftarrow{c}_{t-1} + b_f\right),\\
\hat{c}_{t-1} &= \tfrac{f_t c_{t-1} + \overrightarrow{f}_t \overrightarrow{c}_{t-1} + \overleftarrow{f}_t \overleftarrow{c}_{t-1}}{\mathscr{N}},\\
c_t &= i_t \tanh(W_{x_c}x_t + W_{h_c}h_{t-1} + b_c) + \hat{c}_{t-1},\\
o_t &= \sigma(W_{x_o}x_t + W_{h_o}h_{t-1} + b_o),\\
h_t &= o_t \tanh(c_t).
\end{aligned} \tag{5}$$

Here, we use a bi-directional learning according to the direction of the superpixel tree traversal, Thus, our network is trained in top-down $\overrightarrow{\cdot}$ and bottom-up $\overleftarrow{\cdot}$ directions alternately and iteratively. Given a superpixel p, the sets of its parent and children nodes are $\mathscr{P}(p)$ and $\mathscr{C}(p)$ respectively. For top down training, we consider the information from the parent node $\mathscr{P}(p)$ of superpixel p. So the hidden vector $\overrightarrow{h}_{t-1}$ and cell vector $\overrightarrow{c}_{t-1}$ in Eq. 5 is calculated as follows:

$$\begin{aligned}
\overrightarrow{h}_{t-1} &= h_{t-1}^{\mathscr{P}(p)}\\
\overrightarrow{c}_{t-1} &= c_{t-1}^{\mathscr{P}(p)}
\end{aligned} \tag{6}$$

For bottom up training, we consider the children nodes $\mathscr{C}(p)$. So the hidden vector $\overleftarrow{h}_{t-1}$ and cell vector $\overleftarrow{c}_{t-1}$ is calculated as follows:

$$\begin{aligned}
\overleftarrow{h}_{t-1} &= \frac{\sum\limits_{m \in \mathscr{C}(p)} h_{t-1}^{m}}{M}\\
\overleftarrow{c}_{t-1} &= \frac{\sum\limits_{m \in \mathscr{C}(p)} c_{t-1}^{m}}{M}
\end{aligned} \tag{7}$$

Since the superpixel p may have more than one children nodes in $\mathscr{C}(p)$. $\overleftarrow{h}_{t-1}$ and $\overleftarrow{c}_{t-1}$ are computed by the mean feature value of all the children nodes in $\mathscr{C}(p)$. Here, M is the number of nodes in set $\mathscr{C}(p)$.

With our proposed Hier-LSTM, the recurrent unit on each superpixel can receive structural information not only from the adjacent levels, but also from the cross levels. Forget gates $\overrightarrow{f}_t$ and $\overleftarrow{f}_t$ can help the network filter relative information in the memory. With this manner, the recurrent unit can selectively store structural information in c_t and h_t for specific task like scene labeling.

Since the information transfer is sensitive to the update order of superpixel, we update each level with dynamic order: For $t = 1, 3..$, we update from bottom-to-top and receive $\left[\overleftarrow{h}_t, \overleftarrow{c}_t\right]$ instead of $\left[\overleftarrow{h}_{t-1}, \overleftarrow{c}_{t-1}\right]$; For $t = 2, 4..$, we update from top-to-bottom and receive $\left[\overrightarrow{h}_t, \overrightarrow{c}_t\right]$ instead of $\left[\overrightarrow{h}_{t-1}, \overrightarrow{c}_{t-1}\right]$.

3.2.3. Multi-level fusion and optimization

In our framework, each superpixel in multi-level superpixel maps can obtain a scene prediction result from its corresponding recurrent unit. However, it is hard to decide which observation level is the best for a specific area in image. Thus, for each superpixel, we concatenate the hidden vectors h_t of itself, its parent as well as its ancestor. With this concatenated feature, we can predict

Table 1
Comparison of scene labeling performance on the Pascal Context validation dataset.

Method	Res101	PA(%)	CA(%)	IoU(%)
FCN-8s [2]		65.9	46.5	35.1
ParseNet [30]		67.5	52.3	39.1
DeepLab [31]		–	–	39.6
ConvPP-8 [32]		–	–	41.0
CAMN [33]		72.1	54.3	41.2
HO-CRF [34]		–	–	41.3
PixelNet [35]		–	51.5	41.4
Piecewise [36]		71.5	53.9	43.3
GCPNet [37]	√	73.8	–	46.5
CRF-RNN [38]		–	–	39.3
Grid [15]	√	71.9	54.8	42.6
Graph [19]	√	75.4	56.6	45.1
MS Graph [19]	√	75.5	56.7	45.3
Ours	√	**76.4**	**59.0**	**47.1**

Table 2
Comparison of scene labeling performance on the Sift Flow validation dataset.

Method	Res101	PA(%)	CA(%)	IoU(%)
RCNN [39]		85.1	51.7	–
ParseNet [30]		86.8	52.0	40.4
FCN-8s [2]		85.9	53.9	41.2
Piecewise [36]		88.1	53.4	44.9
CAMN [33]		86.2	58.7	45.2
Grid-LSTM [15]		70.1	22.6	–
DAG-RNN [40]		81.2	45.5	–
CRNN [17]		86.9	57.7	44.7
Grid [15]	√	87.2	53.6	45.7
Graph [19]	√	87.7	55.7	47.1
MS Graph [19]	√	87.9	58.0	48.5
Ours	√	**88.0**	**60.0**	**51.4**

Table 3
Comparison of scene labeling performance on the ADE20K validation dataset.

Method	Res101	PA(%)	CA(%)	IoU(%)
FCN [2]	√	71.3	–	29.4
SegNet [4]		71.0	–	21.6
DilatedNet [41]	√	73.5	–	32.3
CascadeNet [42]	√	74.5	–	34.9
GCPNet [37]	√	77.8	–	38.4
RefineNet101 [43]	√	–	-	40.2
RefineNet152 [43]		–	–	40.7
DD-RNNs [18]	√	–	–	40.9
Grid [15]	√	77.5	47.7	36.7
Graph [19]	√	78.3	50.0	39.2
MS Graph [19]	√	79.4	50.9	40.6
Ours	√	**80.2**	**51.7**	**41.8**

Table 4
Ablation study on Pascal Context and Sift Flow datasets.

Method	Sift Flow		Pascal Context	
	PA(%)	IoU(%)	PA(%)	IoU(%)
Baseline	87.2	42.5	72.4	42.9
w/o LSTM	87.7	47.0	75.1	44.9
Bottom-up Only	87.9	47.9	76.0	45.8
Top-down Only	87.9	48.5	76.0	46.0
w/o Dilation	87.8	50.8	76.1	46.3
Ours-complete	**88.0**	**51.4**	**76.4**	**47.1**

fine-detail and accurate result for the small superpixel in bottom level with abundent structural information.

During training, we consider the class of largest component in a superpixel as the desired label, and we add classifier over each lstm unit to predict scene label for each superpixel. Softmax entropy loss is applied to optimize our network. Note that we calculate loss over whole the sequence recurrent unit output for deeply supervised. The loss come from fusion results is also adopted during training.

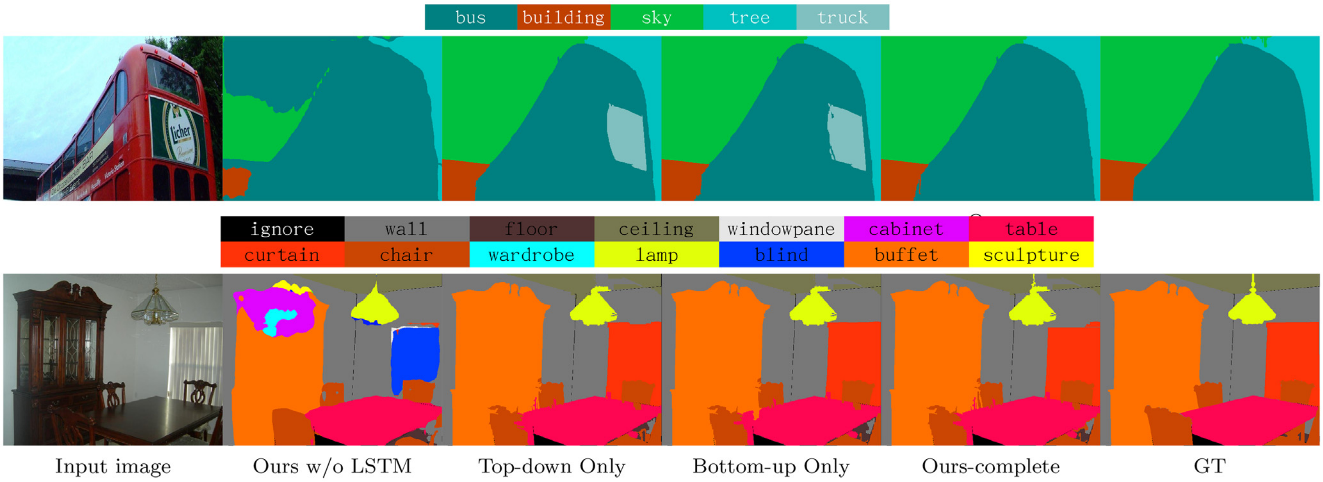


Fig. 5. The ablation visualize results on Pascal-Context and ADE20K datasets. GT indicates ground truths; Ours indicate our structural inference learning model; Top-down Only and Bottom-up Only indicate our system with only top-down modeling or bottom-up modeling; Baseline indicates our Resnet backbone model.

3.3. Implementation details

We implemented the proposed network using Torch library [27]. The network was trained on NVIDIA Geforce GTX1080Ti. For the CNN backbone, we adopt the widely used ResNet101 model [28] pre-trained on ImageNet dataset [29]. For higher accuracy on pixel-level prediction, the global average pooling layer and the final linear classification layer were removed. Dilation convolutions were employed in the last two residual blocks by a factor of 2 and 4 respectively. Thus the resolution of the extracted feature maps can be enlarged from 1/32 to 1/8 of the original size. Then we bilinearly upsample the extracted features maps to the original size and calculated the feature representation for each superpixels.

During the superpixel generation, the number of superpixels in the top level is set to 1 to preserve global structural information from the whole image. In our experiment, the number of superpixels from top to bottom is 1, 4, 16, 32, 64, 128, 256 and 512. How the number of superpixels affect our performance is discussed in Section 4.2.3. The dimension of LSTM hidden state was set to 512, which is same as the size of extracted feature from the CNN backbone. We adopt the module recurrent in 2 times to learn structural relationships, and we cannot observe improvement when more layers are added.

We trained our framework in two steps. First, we added the deconvolutional layers and the softmax layer to the CNN backbone and fine-tune it with evaluation dataset. Then we trained Hier-LSTM with the extracted feature from fine-tuned backbone. We train our model 15 epochs for Pascal-Context and SIFT Flow datasets, and 20 enpochs for ADE20K dataset. We used 4 examples as a batch during training. Adam optimization is used with a fixed learning rate 0.0001.

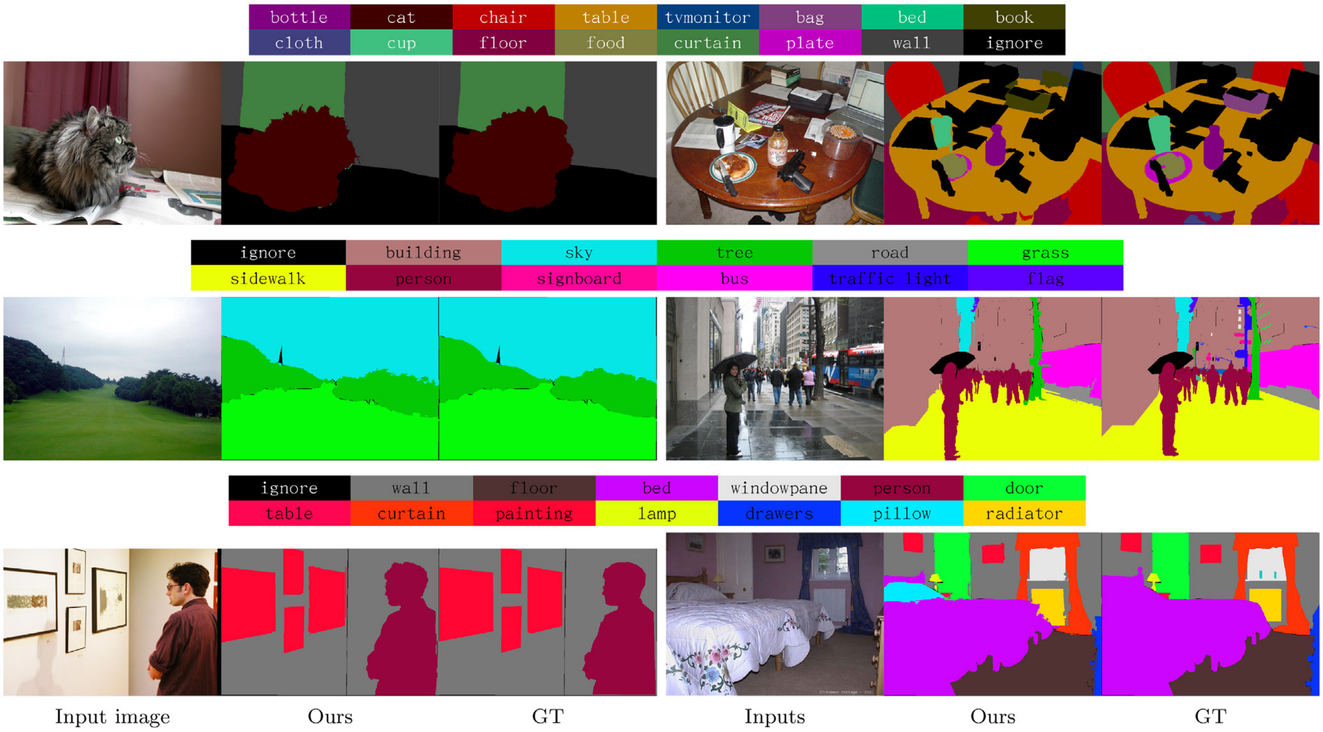


Fig. 6. The scene labeling results of our method from Pascal-Context and ADE20K datasets.

Table 5
Comparison of scene labeling performance on two datasets predicting with different scales.

Scale	Sift Flow			Pascal Context		
	PA(%)	CA(%)	IoU(%)	PA(%)	CA(%)	IoU(%)
1/16	85.1	42.2	36.5	68.8	45.6	35.1
1/64	87.3	53.5	45.6	74.7	55.5	43.9
1/128	87.3	54.5	46.5	75.0	56.5	44.9
1/256	87.8	59.2	48.8	76.2	58.7	46.8
1/512	88.0	60.0	51.4	76.4	59.0	47.1
1/1024	88.0	60.2	50.3	76.4	58.7	46.8

4. Experiments

In this section, we first conduct an experiment to evaluate the performance between our method and state-of-the-art methods. Then an ablation study is introduced to demonstrate the effectiveness of our framework on scene labeling. We also visualize some scene labeling results of our proposed method.

4.1. Datasets and metrics

SIFT Flow [44] consists of 2,688 images. In our experiments, we follow the training/testing split protocol (2,488/200) provided by [44]. The images are captured from 8 typical outdoor scenes with a resolution of 256 × 256 pixels. The task in this dataset is to assign each pixel to one of the 33 semantic classes.

Pascal-Context [45] comprises 4,998 training images and 5105 testing images. Originally, the images are sampled from PASCAL VOC 2010 dataset and re-labeled at pixel-level for the segmentation task. Each image has a resolution of about 375 × 500 pixels.

ADE20K [42] is a challenging scene parsing dataset involves 150 classes. The dataset contains 20,210 (train) and 2,000 (val) pixel-level annotated images. The provided images are labeled with 150 object and stuff classes. The varied resolution of the images and the requirement of distinguishing small stuff bring large challenge to existing methods.

In this paper, we evaluate the performance of scene labeling results by Pixel Accuracy (PA) (the percentage of correctly classified pixels), Per-class Accuracy (CA) and the Intersection-Over-Union (IoU).

4.2. Quantitative evaluation

4.2.1. Comparisons with existing methods

In this experiment, we compare our method with the baseline model, which is the CNN backbone in our paper, three different types of LSTM-based models, i.e. Regular-LSTM [15], Graph-LSTM [19] and Evo-LSTM, and other existing scene labeling methods. Since there is no released code for these three LSTM models, we implemented them according to their papers. For a fair comparison, we implemented these three LSTM-based methods directly on our model by only modifying the core setup while keeping other components fixed.

Tables 1–3 show the evaluations of our method against several scene labeling methods on Pascal Context, Sift Flow and ADE20K respectively. The upper part shows the comparisons with the non-LSTM-based models while the lower part is the comparisons with LSTM-based models. As can be seen, our method outperforms both LSTM-based models and non-LSTM-based models in all datasets. Note that our method shares the same backbone of Res101 to most state-of-the-art methods.

The grid connection (Grid-LSTM) model predicts label on regular grid, which leads to coarse boundaries prediction on small objects. On Pascal Context which has lots of small categories, grid connection even slightly perform lower than the baseline model.

For the graph LSTM model (Graph-LSTM) and the multi scale LSTM model (MS-LSTM), they learn a reliable label inferring on not only the simple scene consisting of sea, sky, road (SIFT Flow dataset), but also the challengeing scene which has lots of foreground objects (cat, flower, cup, etc). Thus their methods both perform better compared with the baseline. Our method further model the structural inferring in multi level bidirectional way, and make our methods more robustly on predicting accurate label for each superpixel based on structural label information.

4.2.2. Ablation study

Table 4 demonstrates the ablation study of our proposed model. We compare our model with baseline model as well as three variations of our model including the model with top down training only (Top-down Only), the model with bottom up training only (Bottom-up Only), and the model without LSTM (ours w/o LSTM).

Comparing the baseline model and the model without LSTM, we can find that the simple multi-level information fusion can also bring an obvious improvement compare to the baseline model thanks to the multi-level scene observation from the superpixels hierarchy. Comparing Bottom-up Only, Top-down Only and Ours-complete, it can be observed that all these three frameworks outperform the models without LSTM. It proves that the hierarchical LSTM mechanism capture the complex structural information from the multi-level superpixel maps. Moreover, the bi-directional training strategy (Ours-complete) brings greater improvement than the other one way training models. It demonstrates the effectiveness of our proposed model on scene labeling by introducing the structure inference. We further evaluate the performance of our model if the dilation convolution is not exploited in the CNN

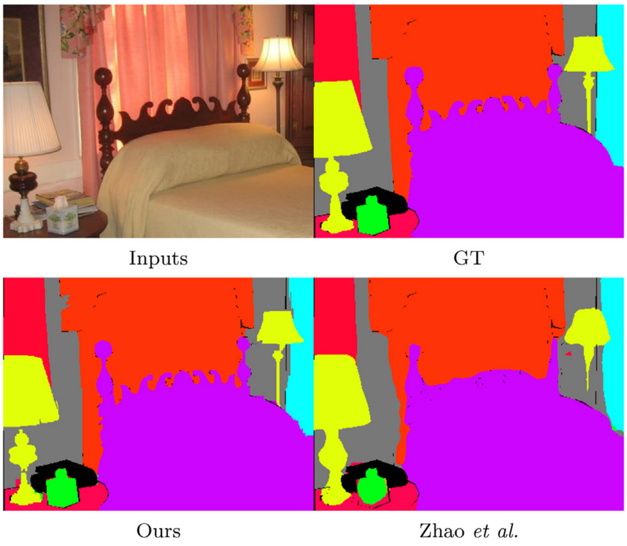

Inputs GT
Ours Zhao *et al.*

Fig. 7. The comparison of visualization result between ours and Zhao et al. [5]. Ths proposed LSTM-based method can predict semantic boundary. more precisely.

Table 6
Time statistics of our system on different resolution.

	Time Cost (ms)			
Resolution	Backbone	SP gen.	SIL Module	Total
256 × 256	14	21	18	53
500 × 375	20	70	18	108

backbone (w/o Dilation). We can find that the model without dilation convolutions performs lower than the equipped one slightly, indicating that gathering more contexts as initialized information for each superpixel can help feature learning.

4.2.3. Superpixel scale

We have decided that the largest scale of the superpixels is the original image itself because we want the network can scan through the whole image to make full use of the structural information. The smallest scale of the superpixels should be carefully designed, as a small scale of superpixels cannot provide sufficient regional information, and it is redundant. While using a too large the scale of superpixels, e.g., 1/16 of the image resolution, it lacks structural information.

To select the appropriate scale of the smallest superpixels, we do ablation analysis about the scale selection of the smallest superpixels in Table 5. The scale in the table indicates the scale factor of superpixel size of the original image resolution. When the scale of superpixels is large (1/16, 1/64, 1/128, etc), one superpixel may contain two or more object categories. It will lead to coarse boundary prediction and greatly harm the precision of the scene labels. When the scale decreases to 1/1024, the our model may trapped into the local information, and can not leverage the global structural context correctly. As the results show, our system performs best when the smallest scale of superpixels equals to 1/512. Thus we use this scale in all the other experiments in this paper.

4.3. Visualization

We show the visualization results of the ablation study in Fig. 5 using Pascal-Context and ADE20k datasets. We can easily observe that without Hier-LSTM, the network get confuse especially on foreground/background or two objects with similar color and textures. The results with LSTM but only one way training shows better scene labeling results than the one without LSTM. However, they may fail when the objects with complex overlapping relationship, like the billboard on the bus. Our complete model with bidirectional training is able to give the most structurally reasonable results and is the most closed one to the groundtruth.

Fig. 6 shows the visualization results on 3 different datasets, SIFT Flow, Pascal Context and ADE20K. The input images on the left hand side in Fig. 6 have clearer boundaries of the objects and between the foreground and background. The input images on the right are from more complex scenarios which contains occlusion, overlapping, objects with different scales. For those easier cases, our method can deliver prediction results almost the same with the groundtruth as a matter of course. When facing the more complex images, our method can still give the structurally reasonable results thanks to the structural information learned by the Hier-LSTM.

Fig. 7 shows more results compared with the non-superpixel based model [21]. In this evaluation, our method can predicts more precisely on the sharp or thin boundaries, e.g., the thin structure the bed and floor lamp. This big visual improvement is caused by the hierarchical superpixels that contain precise semantic boundary. The existing non-superpixel based models, PSPNet [21] for example, use convolutional layer with grid-based kernel for capturing semantic layout information. Therefore the network has similar semantic features on neighbor pixels while ignoring the semantic boundary, making coarse prediction on irregular boundary.

4.4. Time performance

Table 6 shows the time performance of our system. We show the results in millisecond on resolution 256 × 256 (the average resolution of SIFT Flow dataset) and 500 × 375 (the average resolution of Pascal-Context dataset). Here Backbone indicates the running time of backbone model. SP generation indicates the running time of multi-level superpixel generation. SIL module indicates the running time of our structural inference learning module. We can observe that both of these resolution achieve fast prediction. Our proposed method runs at roughly 20 FPS on 256 × 256 (SiftflowVal) and 10 FPS on 500 × 375 (PascalContextVal). We further compare the running time of proposed method with existing LSTM-based methods [17,18,22] in Table 7. They model information flow based on neighbor direction, which may have ten or more superpixels as neighbors at the same time. The slow speed of tedious dependencies learning of their methods leads to heavy computation. We model information flow based on parent–child direction, and we ensure that there is at most one parent for each superpixel, and the sum of the number of children from the superpixels within one level is equal to the superpixel size of next level. Thus we can see that our SIL module runs in stable (Table 6) and lead to low time-consuming of the proposed algorithm.

5. Conclusion

In this paper, we present a fast scene labeling method via structural inference over hierarchical superpixels. Instead of constructing the multi-level superpixels one by one, we generate all the superpixels based on the merging order of minimum spanning tree and therefore achieve fast generation. We further propose a bidirectional recurrent network to learn the structural inference over the hierarchical superpixels. The process can be divided into top-down and bottom-up stages, which aims to aggregate both local and global image structures. The proposed method is able to explore the interactions between semantic layouts in different scales, which enables hierarchical features learning and size-aware semantic prediction. The proposed scene labeling method is evaluated and compared with state-of-the-art algorithms and achieves better results in both numerical and visual evaluations.

As the proposed hierarchical superpixels segmentation is fast and general, we aim to extend this structure to different applications in the future.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table 7
Time statistics of our system compared with other methods.

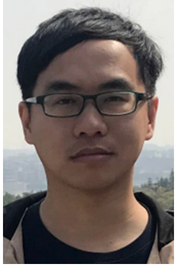
Method	Resolution	Time Cost (ms)
Evo-LSTM [22]	–	1300
CRNN [17]	384 × 384	700
Ours	384 × 384	83
DD-RNNs [18]	512 × 512	360
Ours	512 × 512	142

Acknowledgment

The work is supported by NSFC (Grant Nos. 61772206, U1611461, 61472145, 61702194, 61972162), Guangdong R&D key project of China (Grant No. 2018B010107003, 2020B010165004, 2020B010166003), Guangdong High-level personnel program (Grant No. 2016TQ03X319), Guangdong NSF (Grant No. 2017A030311027), Guangzhou Key Project in Industrial Technology (Grant Nos. 201802010027, 201802010036), and the CCF-Tencent Open Research fund (CCF-Tencent RAGR20190112).

References

[1] C. Farabet, C. Couprie, L. Najman, Y. LeCun, Learning hierarchical features for scene labeling, IEEE TPAMI 35 (8) (2013) 1915–1929.
[2] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proc. CVPR, 2015, pp. 3431–3440.
[3] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Proc. MICAI, Springer, 2015, pp. 234–241.
[4] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE TPAMI 39 (12) (2017) 2481–2495.
[5] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, in: Proc. CVPR, 2017, pp. 2881–2890.
[6] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, arXiv preprint arXiv:1802.02611..
[7] X. Guo, Z. Wang, Q. Yang, W. Lv, X. Liu, Q. Wu, J. Huang, Gan-based virtual-to-real image translation for urban scene semantic segmentation, Neurocomputing. 394 (2020) 127–135.
[8] F. Wu, F. Chen, X.-Y. Jing, C.-H. Hu, Q. Ge, Y. Ji, Dynamic attention network for semantic segmentation, Neurocomputing 384 (2020) 182–191.
[9] F. Zhou, Y. Hu, X. Shen, Scale-aware spatial pyramid pooling with both encoder-mask and scale-attention for semantic segmentation, Neurocomputing 383 (2020) 174–182.
[10] B. Zhao, X. Zhang, Z. Li, X. Hu, A multi-scale strategy for deep semantic segmentation with convolutional neural networks, Neurocomputing 365 (2019) 273–284.
[11] F. Lateef, Y. Ruichek, Survey on semantic segmentation using deep learning techniques, Neurocomputing 338 (2019) 321–348.
[12] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780.
[13] M.F. Stollenga, W. Byeon, M. Liwicki, J. Schmidhuber, Parallel multi-dimensional lstm, with application to fast biomedical volumetric image segmentation, in: Proc. NIPS, 2015, pp. 2998–3006.
[14] Z. Li, Y. Gan, X. Liang, Y. Yu, H. Cheng, L. Lin, Lstm-cf: Unifying context modeling and fusion with lstms for rgb-d scene labeling, in: Proc. ECCV, Springer, 2016, pp. 541–557.
[15] W. Byeon, T.M. Breuel, F. Raue, M. Liwicki, Scene labeling with lstm recurrent neural networks, in: Proc. CVPR, 2015, pp. 3547–3555.
[16] X. Liang, X. Shen, D. Xiang, J. Feng, L. Lin, S. Yan, Semantic object parsing with local-global long short-term memory, in: Proc. CVPR, 2016, pp. 3185–3193.
[17] H. Fan, X. Mei, D. Prokhorov, H. Ling, Multi-level contextual rnns with attention model for scene labeling, IEEE TITS 19 (11) (2018) 3475–3485.
[18] H. Fan, P. Chu, L.J. Latecki, H. Ling, Scene parsing via dense recurrent neural networks with attentional selection, in: Proc. WACV, IEEE, 2019, pp. 1816–1825.
[19] X. Liang, X. Shen, J. Feng, L. Lin, S. Yan, Semantic object parsing with graph lstm, in: Proc. ECCV, Springer, 2016, pp. 125–143.
[20] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk, Slic superpixels compared to state-of-the-art superpixel methods, IEEE TPAMI 34 (11) (2012) 2274–2282.
[21] Z. Peng, R. Zhang, X. Liang, X. Liu, L. Lin, Geometric scene parsing with hierarchical LSTM, in: Proc. IJCAI, 2016, pp. 3439–3445.
[22] X. Liang, L. Lin, X. Shen, J. Feng, S. Yan, E.P. Xing, Interpretable structure-evolving lstm, in: Proc. CVPR, 2017, pp. 2175–2184.
[23] Z. Zuo, B. Shuai, G. Wang, X. Liu, X. Wang, B. Wang, Y. Chen, Learning contextual dependence with convolutional hierarchical recurrent neural networks, IEEE TIP 25 (7) (2016) 2983–2996.
[24] D.B. West, et al., Introduction to Graph Theory, vol. 2, Prentice hall Upper Saddle River, 2001..
[25] P. Dollár, C.L. Zitnick, Fast edge detection using structured forests, IEEE TPAMI 37 (8) (2015) 1558–1570.
[26] F.A. Gers, N.N. Schraudolph, J. Schmidhuber, Learning precise timing with lstm recurrent networks, Journal of Machine Learning Research 3 (Aug) (2002) 115–143.
[27] R. Collobert, K. Kavukcuoglu, C. Farabet, Torch7: A matlab-like environment for machine learning, in: Proc. NIPS, no. EPFL-CONF-192376, 2011.
[28] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. CVPR, 2016, pp. 770–778.
[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Proc. CVPR, IEEE, 2009, pp. 248–255.
[30] W. Liu, A. Rabinovich, A.C. Berg, Parsenet: Looking wider to see better, arXiv preprint arXiv:1506.04579..
[31] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, arXiv preprint arXiv:1606.00915..
[32] S. Xie, X. Huang, Z. Tu, Top-down learning for structured labeling with convolutional pseudoprior, in: Proc. ECCV, Springer, 2016, pp. 302–317.
[33] A.H. Abdulnabi, B. Shuai, S. Winkler, G. Wang, Episodic camn: Contextual attention-based memory networks with iterative feedback for scene labeling, in: Proc. CVPR, IEEE, 2017, pp. 6278–6287.
[34] A. Arnab, S. Jayasumana, S. Zheng, P.H. Torr, Higher order conditional random fields in deep neural networks, in: Proc. ECCV, Springer, 2016, pp. 524–540.
[35] A. Bansal, X. Chen, B. Russell, A. Gupta, D. Ramanan, Pixelnet: Towards a general pixel-level architecture, arXiv preprint arXiv:1609.06694..
[36] G. Lin, C. Shen, A. Van Den Hengel, I. Reid, Efficient piecewise training of deep structured models for semantic segmentation, in: Proc. CVPR, 2016, pp. 3194–3203.
[37] W.-C. Hung, Y.-H. Tsai, X. Shen, Z.L. Lin, K. Sunkavalli, X. Lu, M.-H. Yang, Scene parsing with global context embedding., in: Proc. ICCV, 2017, pp. 2650–2658..
[38] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, P. H. Torr, Conditional random fields as recurrent neural networks, in: Proc. ICCV, 2015, pp. 1529–1537..
[39] M. Liang, X. Hu, B. Zhang, Convolutional neural networks with intra-layer recurrent connections for scene labeling, in: Proc. NIPS, 2015, pp. 937–945.
[40] B. Shuai, Z. Zuo, B. Wang, G. Wang, Dag-recurrent neural networks for scene labeling, in: Proc. CVPR, 2016, pp. 3620–3629.
[41] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, arXiv preprint arXiv:1511.07122..
[42] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, A. Torralba, Scene parsing through ade20k dataset, in: Proc. CVPR, vol. 1, IEEE, 2017, p. 4..
[43] G. Lin, A. Milan, C. Shen, I. Reid, Refinenet: Multi-path refinement networks for high-resolution semantic segmentation, in: Proc. CVPR, vol. 1, 2017, p. 3..
[44] C. Liu, J. Yuen, A. Torralba, Nonparametric scene parsing: Label transfer via dense scene alignment, in: Proc. CVPR, IEEE, 2009, pp. 1972–1979.
[45] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, A. Yuille, The role of context for object detection and semantic segmentation in the wild, in: Proc. CVPR, 2014.



Huaidong Zhang received the B.Eng. degree in the Computer Science and Technology from South China University of Technology, China, in 2015. He is currently working toward the Ph.D. degree with the School of Computer Science and Engineering, South China University of Technology. His research interests include computer vision, image processing, computer graphics and deep learning.

Chu Han is now a postdoctoral fellow at the Guangdong Provincial People's Hospital, Guangdong Academy of Medical Sciences, under the supervision of Prof. Zaiyi Liu and Prof. Changhong Liang. He received his Ph.D. degree from the Chinese University of Hong Kong, under the supervision of Prof. Tien-Tsin Wong. He received the M.Sc. degree in computer science from South China University of Technology, and the B.Sc. degree from South China Agricultural University. His current research interests include medical image analysis, computer graphics, image processing, computer vision and deep learning.

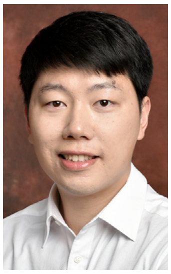
Xiaodan Zhang is currently a post-doctoral fellow of Beijing University of Technology, Bejing, China. She received the B.S. degree from Zhengzhou University in 2010. She obtained her M.S. degree and Ph.D degree from University of Chinese Academy of Sciences, in 2014 and 2018 respectively. Her current research interests include computer vision and natural language processing, specifically for image captioning and image recognition.

Yong Du is an assistant professor in the Department of Computer Science and Technology, Ocean University of China. He obtained B.Sc. and M.Sc. degrees from Jiangnan University and a Ph.D. degree from South China University of Technology. His research interests include computer vision and image processing.

Xuemiao Xu received her B.S. and M.S. degrees in Computer Science and Engineering from South China University of Technology in 2002 and 2005 respectively, and Ph.D. degree in Computer Science and Engineering from The Chinese University of Hong Kong in 2009. She is currently a professor in the School of Computer Science and Engineering, South China University of Technology. Her research interests include object detection, tracking, recognition, and image, video understanding and synthesis, particularly their applications in the intelligent transportation.

Guoqiang Han received his Ph.D. in Sun Yat-sen University in 1988, and he was assigned to teach at South China University of Technology in the same year. He was promoted to professor in 1993. From October 1997 to September 1999, he was a postdoctoral researcher at the University of Tokyo. He is currently a professor in the School of Computer Science and Engineering, South China University of Technology. His research interests include image processing, computer vision and computer graphics.

Jing Qin received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong in 2009. He is currently an Assistant Professor with the School of Nursing, The Hong Kong Polytechnic University, Hong Kong. He is also a Key Member with the Centre for Smart Health, SN, The Hong Kong Polytechnic University. His research interests include innovations for healthcare and medicine applications, medical image processing, deep learning, visualization and human–computer interaction, and health informatics.



Shengfeng He is an Associate Professor in the School of Computer Science and Engineering, South China University of Technology. He was a Research Fellow at City University of Hong Kong. He obtained his B.Sc. degree and M.Sc. degree from Macau University of Science and Technology, and the Ph.D degree from City University of Hong Kong. His research interests include computer vision, image processing, computer graphics, and deep learning.