Deep Context Modeling for Semantic Segmentation

Kien Nguyen Clinton Fookes Sridha Sridharan
Image and Video lab, SAIVT Research Program
Queensland University of Technology, Brisbane, Australia
{k.nguyenthanh,c.fookes,s.sridharan}@qut.edu.au

Abstract

Deep convolutional neural networks (DCNNs) have been employed in many computer vision tasks with great success due to their robustness in feature learning. One of the advantages of DCNNs is their representation robustness to object locations, which is useful for object recognition tasks. However, this also discards spatial information, which is useful when dealing with topological information of the image (e.g. scene parsing, face recognition). Adopting graphical models (GMs) to incorporate spatial and contextual information into the DCNNs is expected to improve the performance of DCNN-based computer vision tasks. Recent research has shown that combining DCNNs and Conditional Random Fields (CRFs) can significantly improve scene parsing accuracy. This is achieved either through the combination of their independent outputs or through their application as a cascade. In this work, we propose a novel strategy to incorporate CRFs deeper inside DCNNs by modeling a CRF as a DCNN layer which is pluggable into any layer of a DCNN. This implants spatial and contextual information into the DCNN, allowing end-to-end training, better controlling the spatial constraints and improving segmentation accuracy. The new strategy for coupling graphical models with the state-of-the-art fully convolutional neural network has shown promising results on the PASCAL-Context dataset.

Keywords: *semantic segmentation, scene understanding, scene parsing, context modeling*

1. Introduction

Semantic segmentation, also known as scene parsing/labeling, is a task to label every pixel in the image with the corresponding object class which it belongs to. After a perfect scene parsing, every region and every object is delineated and tagged [10]. Semantic segmentation is an important task for scene understanding. However, it is a challenging problem as it combines three traditional problems: object detection, segmentation and multi-labels recognition in a single process. From a feature representation point of view, Farabet *et al.* raised two fundamental questions in the context of efficient semantic segmentation [10]:

- Feature representation: How to produce good internal representations of the visual information (local features)?
- Contextual representation: How to employ contextual information to ensure the self-consistency of the interpretation (global features/relationship)?

Finding a good feature representation is critical to the segmentation task. Most traditional approaches [11, 42, 36] rely on hand-crafted features, e.g., color histogram, SIFT [35], HOG [7]. Recently, deep learning has gained great popularity in learning to represent features for computer vision tasks. Since layer-wise learning algorithms were revised in 2006 [18], Deep Learning in general and large-scale Deep Convolutional Neural Networks (DCNNs) in particular have significantly advanced the performance of computer vision systems, including object detection, object segmentation, object recognition and natural language processing systems [38, 28]. With their built-in hierarchical representation learned directly from the data rather than human assumption, which is robust to translation, rotation, scale and deformation variation, DCNNs provide an effective response to the first question raised by Farabet *et al.*. In recent years, we have witnessed great success of DCNNs in semantic segmentation, outperforming all other traditional approaches (i.e. all top 10 approaches on the segmentation challenge in the PASCAL VOC 2012 dataset are CNN-based [1]).

Incorporating contextual information into the parsing task not only provides self-consistency of the interpretation, but also improves the meaningful layout of the scene. To address this problem and deal with the second question mentioned above, the research community has been focusing on modifying DCNNs to incorporate context/global information. Farabet *et al.* [10] represented the raw input image in a 3-scale Laplacian pyramid before feeding them to three 3-stage convolutional networks. The outputs of three

convolutional networks are concatenated with the coarser-scale feature maps being upsampled to match the size of the finest-scale map. This allows integrating large context (as large as the whole scene) into local decisions, yet still remaining manageable in terms of parameters/dimensionality. Long *et al.* [34] casted the network into fully convolutional by replacing the last fully connected neural layers, which have fixed dimensions and throw away spatial coordinates, by 1×1 convolutional layers. Rather than using multiple scales of the input image to feed into multiple networks like [10], they upsampled then concatenated intermediate feature maps outputted from intermediate neural network layers. Concatenating features from multiple intermediate layers has also been used to learn hypercolumn features to expand the contextual relationship modeling [15] or to emphasize the boundary cues [3]. Yu *et al.* seek another approach to aggregat multi-scale contextual information by introducing dilated convolutions for the segmentation modules [43]. Differently, the authors in [16, 12, 13] extracted two types of Region-CNN features: region features extracted from proposal bounding boxes and segment features extracted from the raw image content masked by the segments. However, Dai *et al.* [6] showed that using the masks in the image content as in [16, 12, 13] may lead to artificial boundaries.

Concatenating hierarchical features from multiple layers within a network [34, 15, 3] or multiple regions within a network [16, 12, 13] or multiple shared networks [10, 2] is on the one hand capable of incorporating spatial/contextual relationships of surrounding pixels at different ranges, but on the other hand, it is also detrimental to the fineness of boundary segmentation. Chen *et al.* [5] proved that while the built-in spatial invariance property of DCNNs is effective in high-level vision tasks, it can hamper low-level vision tasks such as semantic segmentation when dealing with fine details. Similarly, Farabet *et al.* also showed that the parsing result of DCNNs, although fairly accurate, is not satisfying visually, as it lacks spatial consistency and precise delineation of objects [10]. The obvious key to the success of a scene parsing system is the capability to model the spatial relationships among pixels. Intuitively, probabilistic graphical models have been long employed to model these relationships [23]. Among graphical models, Conditional Random Fields (CRFs) [27] have been quite successful in segmentation owing to their ability to directly predict the segmentation given the observed image and the ease with which arbitrary functions of the observed features can be incorporated into the training process [41, 26, 24, 22, 17].

Witnessing the advantages of DCNNs and CRFs in the scene parsing task, researchers have started to loosely couple them to improve the accuracy of the pixel-wise semantic segmentation [10, 2, 5, 33]. It is apparent that the spatial information learned from the graphical models and the discriminative hierarchical feature representation will compliment each other, allowing further improvement of the scene parsing task. In this work, we propose a novel approach to deeply integrate context learned by fully connected CRFs into a DCNN to improve the accuracy of the segmentation task. Our contribution is twofold:

- Firstly, we systematically analyze and categorize the state-of-the-art trends of incorporating contextual/spatial information into deep learning approaches for the semantic segmentation task. Our analysis shows a shift trend from loosely combine CRFs and DCNNs to closely integrate them.

- Secondly, motivated by the observation, we propose a novel approach to integrate CRFs deeper inside DCNNs by modeling a fully connected CRF model as a deepnet layer which can perform forward and back-propagation. These layers are based on the Long Short-Term Memory (LSTM) models to incorporate contextual and spatial information into the deep learning approach. Particularly, these CRF-LSTM layers can be plugged in and combined with the deepnet layers for end-to-end training and testing.

The remainder of this paper is organized as follows: Section 2 presents the background models of Fully Convolutional Neural Networks and Fully Connected CRFs; Section 3 summarizes trends in incorporating CRFs into DCNNs and describes our deeply-integrated approach; Section 4 discusses the experimental results; and the paper is concluded in Section 5.

2. Fully convolutional neural network (FCN) and Fully Connected Conditional Random Fields (CRFs)

In this paper, a fully convolutional neural network and a fully connected conditional random fields model are combined in a novel strategy for tightly integrating deep context into the segmentation task. In this section, we introduce these two background models employed in this work.

2.1. Fully convolutional neural networks (FCN) for Semantic Segmentation

Convolutional neural networks are powerful at visual modeling hierarchies of features. Networks such as LeNet [29], AlexNet [25] and its deeper successors [40, 39] have shown great success in various computer vision tasks [38]. These networks consist of multiple convolutional layers (convolution + nonlinear activation + pooling) followed by multiple fully connected layers [29, 25, 40, 39]. These fully connected layers have fixed dimensions and discard spatial coordinates. This, on the one hand, is useful for high level tasks such as recognition due to the robustness to locations of the objects, but on the other hand, is detrimental

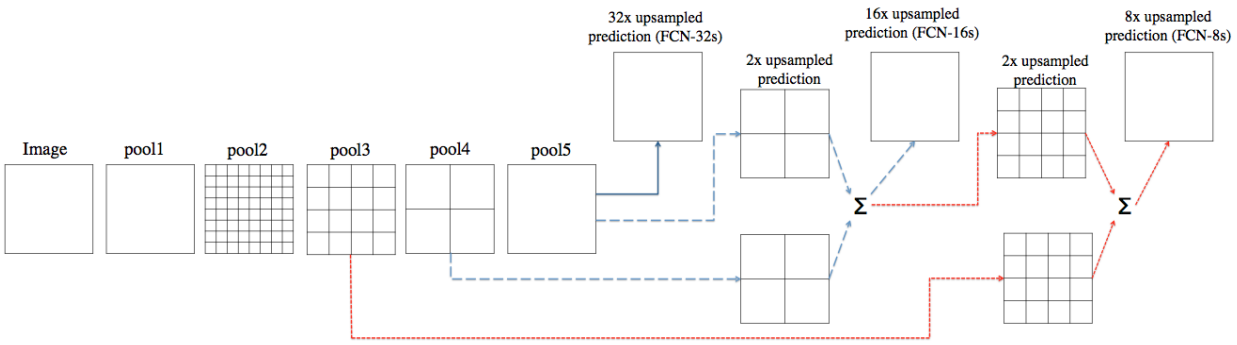

Figure 1. Fully convolutional neural network (FCN) combine coarse, high layer information with fine, low layer information to model the spatial information in the model [34].

to lower level tasks such as segmentation. Observing that these fully connected layers can be viewed as convolutions with kernels that cover their entire input regions, Long *et al.* proposed to replace those fully connected layers with equivalent convolutions to cast the network into a fully convolutional network for the semantic segmentation task [34].

Following the approach in [34], we adapted the pre-trained models available including AlexNet [25], GoogLeNet [40] and VGG 16-layer net [39] to get a fully convolutional network as shown in Figure 1. The network consists of five convolutional layers with each followed by a pooling layer. The outputs are achieved by upsampling and combining predictions at different levels. Three outputs are employed here: FCN-32s by $32x$ upsampling the prediction from the pool 5 layer, FCN-16s by $16x$ upsampling the summation of $2x$ upsampled prediction from the pool 5 layer and prediction from the pool 4 layer, and FCN-8s by $8x$ upsampling the summation of the summation in FCN-16s with prediction from the pool 3 layer.

2.2. Fully connected CRFs for Semantic Segmentation

Fully connected CRFs are a type of discriminative undirected probabilistic graphical model which models the relationships of every pixel pair in the image. These models are effective in modeling the spatial information of the image, which is useful for the semantic segmentation task [24].

Assume that we have a set of input images, $I = \{I_1, ..., I_N\}$, and its set of corresponding pixel-level image labelings, $X = \{X_1, ..., X_N\}$. Both sets I and X are random fields. There are k label classes for labeling each pixel, $L = \{l_1, ..., l_k\}$. By the fundamental theorem of random fields [14], a conditional random field (I, X) is characterized by a Gibbs distribution,

$$P(X|I) = \frac{1}{Z(I)} exp(-\Sigma_{c \in C_G} \Phi_c(X_c|I)), \qquad (1)$$

where $G = (V, E)$ is a graph on X and each clique, c, in a set of cliques, C_G, in G induces a potential, Φ_c [27]. The Gibbs energy of a labeling $x \in L^N$ is $E(x|I) = \Sigma_{c \in C_G} \Phi_c(X_c|I)$. In a fully connected pairwise CRF model, G is the complete graph on X and C_G is the set of all unary and pairwise cliques. The corresponding Gibbs energy is,

$$E(x) = \Sigma_i \psi_u(x_i) + \Sigma_{i<j} \psi_p(x_i, x_j), \qquad (2)$$

where i and j ranges from 1 to N. There are two factors affecting the energy of a labeling: unary potential, $\psi_u(x_i)$, and pairwise potential, $\psi_p(x_i, x_j)$. While the unary potential presents how likely a node takes on a label, the edge pairwise potential presents how likely the labels of two pixels agree. The unary potential normally incorporates shape, texture, location and color descriptor [24] or hand-crafted features such as SIFT [35] and HOG [7]. The pairwise edge potentials can be modeled as linear combinations of Gaussians,

$$\Psi_p(x_i, x_j) = \mu(x_i, x_j) \Sigma_{m=1}^{K} w^{(m)} k^{(m)}(f_i, f_j), \qquad (3)$$

where each $k^{(m)}$ is a Gaussian kernel,

$$k^{(m)}(f_x, f_j) = exp(-\frac{1}{2}(f_i - f_j)^T \Lambda^{(m)} (f_i - f_j)), \qquad (4)$$

and the vectors, f_i and f_j, are feature vectors for pixel i and j in an arbitrary feature space; $w^{(m)}$, are linear combination weights; and μ is a label compatibility function. For scene parsing, a contrast-sensitive two-kernel potential, combining an appearance kernel and a smoothness kernel has been proposed [24],

$$k = w^{(1)} \underbrace{exp(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2})}_{\text{appearance kernel}} + w^{(2)} \underbrace{exp(\frac{|p_i - p_j|^2}{2\theta_\gamma^2})}_{\text{smoothness kernel}}, \qquad (5)$$

where p_i, p_j and I_i, I_j represent the pixel positions and pixel color intensities of pixel i, j. The *appearance kernel* is motivated by the observation that nearby pixels with similar color are likely to be in the same class. The degree of

nearness and color similarity are controlled by parameters θ_α andθ_β. The *smoothness kernel* removes small isolated regions.

While performing inference, the best labeling is found by a Maximum A Posterior (MAP) approach,

$$\hat{x} = argmax_x P(x|I). \quad (6)$$

Since a fully connected CRF is capable of modeling the relationships between all pairs of pixels in the image, it is expected to better represent long-range, or even scene-level spatial relationships between pixels in the image. However, it comes at the cost of computation, where tens of thousands of nodes and billions of edges even on a low-resolution hundreds-by-hundreds image make traditional inference impractical. To deal with computational expense, Krahenbuhl and Koltun [24] employed a mean field approximation to the CRF distribution. This approximation is iteratively optimized through a series of message passing steps, each of which updates a single variable by aggregating information from all other variables. They showed that a mean field update of all variables in a fully connected CRF can be performed using Gaussian filtering in feature space. This reduces the complexity of message passing from quadratic to linear, resulting in an approximate inference algorithm for fully connected CRFs that is linear in the number of variables N and sublinear in the number of edges in the model.

3. Coupling strategy

Other than self-modifying DCNNs to incorporate contextual/global information as discussed in Section 1, a plethora of works have tried to incorporate CRFs with DCNNs for enhancing the segmentation accuracy. The state-of-the-art approaches can be summarized as in the following categories:

- Trend 1 - Parallel: Applying DCNNs and CRFs in parallel as two separate approaches, then combining outputs. Farabet *et al.* employs a 2-layer neural network to combine deep learning features extracted by DCNNs and graphical model features extracted by CRFs [9, 10]. In the same vein, Kekec *et al.* use one DCNN to learn the CRF-type contextual information and another DCNN to learn visual features, then combine them for scene labeling [21].

- Trend 2 - Cascading: Applying DCNNs and CRFs in cascading order, which means using the output of the other as input. The initial work in cascading has used CRFs as a post-processing/second step after DCNNs [10]. Liu *et al.* [33] oversegmented the input image into superpixels, calculated deep convolutional features be a DCNN model pre-trained on ImageNet, before feeding these features into a CRF. A Structured Support Vector Machine (SSVM) is employed to learn the parameters of the CRF model. Both Farabet *et al.* and Alvarez *et al.* simultaneously classified each pixel of the image densely by a multi-scale DCNN and oversegmented the image with superpixels [10, 2]. Similarly, Chen et al. also employed the dense segmentation map computed by a DCNN to model the unary potential, but using a fully connected pairwise CRF model [5]. A fully connected CRF better represents long-range dependencies between the pixels in the image, enhancing the semantic segmentation. Recently, Lin *et al.* investigated learning the unary and pairwise potentials directly from the training images by multi-scale CNNs [32].

- Trend 3 - Jointly: Applying DCNNs and CRFs jointly: rather than applying DCNNs and CRFs separately in parallel or in cascading order, some works have jointly learned them in a unified framework. Zheng *et al.* showed that a CRF is equivalent to one Recurrent Neural Network (RNN) layer and this CRF-RNN layer can be appended to the deepnet for end-to-end training to improve the segmentation accuracy [44].

These trends shows a shift towards drawing CRFs closer to the DCNNs, from parallel to cascading, then to jointly. This observation motivates us to integrate CRFs deeper inside the architecture of DCNNs to better model and embed the spatial and contextual information.

While appending the CRF-RNN layer helps, it is limited in the modeling capability since spatial/contextual information is learned only at the end of the network and is heavily biased by the output of the deepnet. In addition, the CRF-RNN layer is not able to plug into other earlier stages of the deepnet due to the gradient vanishing and exploding effect when propagating the gradients down through many layers [19]. In this paper, we propose a novel CRF-LSTM layer to model the graphical information which can be easily plugged in any location in a deepnet as illustrated in Figure 2. Moving it to earlier stage (i.e. inserting it in earlier portion of the network) allows it to learn intermediate spatial/contextual relationship of intermediate layers, rather than just the final output layer.

In comparison with vanilla-LSTM and its variants (e.g. [31, 30, 4]) which have been shown to be capable of learning spatial correlation, our proposed approach is fundamentally different by incorporating mean-field iterations of a CRF inside each vanilla-LSTM. By incorporating a mean-field iteration of a CRF inside each vanilla-LSTM, our proposed CRF-LSTM capitalizes on the learning capability of both LSTM and CRF to learn strong spatial/contextual clues for the segmentation task. We also show that combining multiple CRF-LSTMs in a specific configuration will better

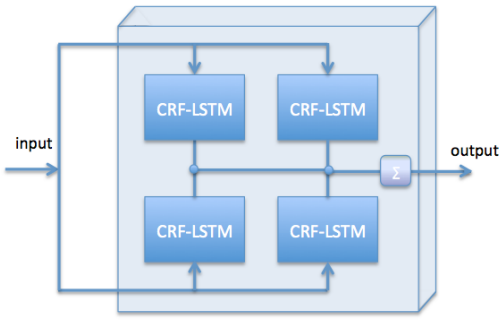

Figure 2. A CRF-LSTM layer consists four CRF-LSTM units, which are connected to their neighbors in one of four quarters: left-top, right-top, left-bottom and right-bottom. These four units when operating simultaneously model the spatial relationship of the current location to its surrounding contexts.

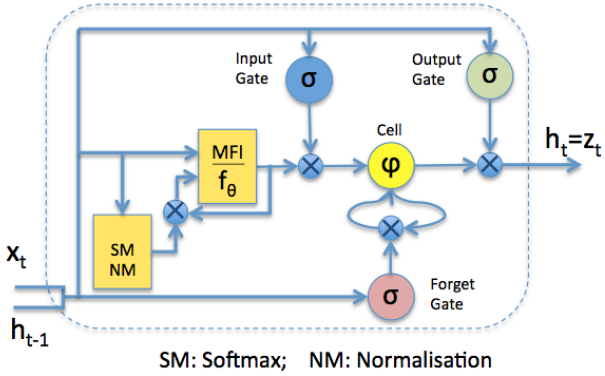

Figure 3. A CRF-LSTM unit is constructed by replacing the input modulation section with the gated combination of a softmax and a meanfield iteration (MFI) function f_θ.

learn multi-directional surrounding spatial/contextual information. Our experiments on the PASCAL-Context sets have shown the effectiveness of the proposed approaches.

3.1. CRF-LSTM layer

In this section, we describe a novel CRF-LSTM layer to model the graphical information which can be easily plugged into a deepnet as illustrated in Figure 2. Four CRF-LSTM units, which are assembled connectedly, are used to represent the surrounding context in different directions. These CRF-LSTM units allow the system to model and memorize the spatial/contextual information. Once modeled by the CRF-LSTM layers, the graphical information can be spread through the whole deepnet. The input travels through all four CRF-LSTM units. Each unit is connected to its neighbors in one of four quarters: left-top, right-top, left-bottom and right-bottom. These four units when operating simultaneously model the spatial relationship of the current location to its surrounding contexts.

Authors in [44] have shown that a mean-field iteration in the iterative algorithm for label estimation using fully connected CRFs can be approximated by the gated combination of a softmax and a mean-field iteration function. Hence, they model a CRF layer as a RNN layer. To avoid the gradient vanishing and exploding effect, we propose to formulate the iterative mean-field algorithm as LSTM.

Traditionally, a LSTM unit consists of three types of gates: (i) input gate, (ii) forget gate, and (iii) output gate. At the core of the LSTM model is a memory cell, which encodes the knowledge of the inputs that have been observed up to that step [8]. This cell is modulated by gates. The cell coupled with the forget gate function as a memory unit that allows the network to learn when to forget the previous hidden states and when to update the hidden states given new information [8]. We modify the conventional LSTM unit by replacing the input modulation section with the gated combination of a softmax and a meanfield iteration (MFI) function f_θ as shown in Figure 3. The gated combination models a fully connected CRF [44].

4. Experimental results

PASCAL-Context dataset augments PASCAL VOC 2010 dataset with annotations for 500+ additional categories, allowing diverse tasks towards comprehensively parsing the images [37]. PASCAL-Context is chosen, other than PASCAL VOC 2012, for the experiments since PASCAL-Context is a subset of PASCAL VOC with richer annotations, which requires stronger contextual/spatial clues for a good segmentation task. This dataset is also widely-used for evaluating semantic segmentation such as [34, 37, 44]. The dataset contains pixel-wise annotations for 10,103 images in the Training and Validation subsets of PASCAL VOC 2010 dataset. Due to the unbalance of the classes appearing in the dataset, similar to [37], we choose 59 most frequent classes for our experiments. The remaining classes are classified as background in this research. The scene parsing task has been performed on training subsets (4,998 images) and testing subsets (5,105 images) of the PASCAL-context dataset.

In this research, we employed a public framework called CAFFE [20]. Caffe is a clean and modifiable C++ framework with state-of-the-art deep learning algorithms for training and deploying general-purpose convolutional neural networks and other deep models efficiently on commodity architectures. We first implemented the baseline as a fully convolutional neural network with three different outputs: FCN-32s, FCN-16s and FCN-8s as described in Section 2.1. The segmentation accuracy metric used here is the pixel accuracy of the images. The segmentation accuracies achieved for three baseline outputs FCN-32s, FCN-16s and FCN-8s are 66.7%, 66.9% and 67.2% respectively.

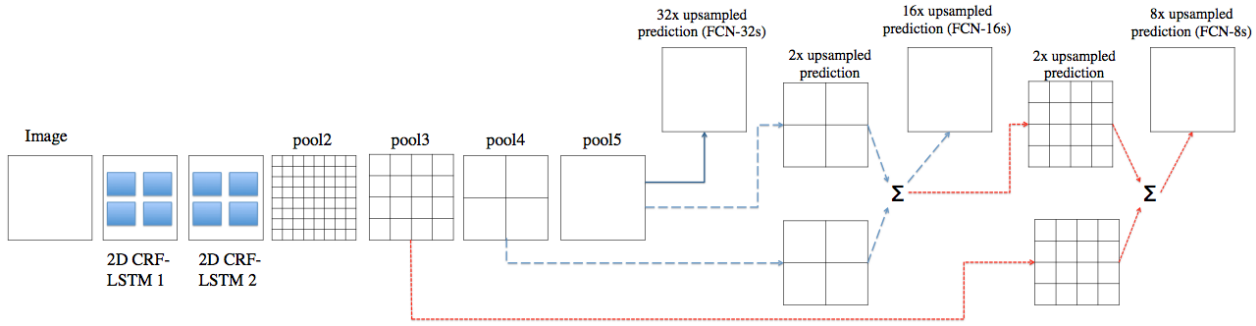

Figure 4. Two CRF-LSTM layers located right after the input image, followed by 4 traditional convolutional-pooling layers of a FCN.

Table 1. Performance of appending one proposed CRF-LSTM layer in comparison with baseline and one CRF-RNN layer to the FCN for the semantic segmentation task on the PASCAL-context dataset.

Approaches	Accuracy (%)
FCN-32s	66.7%
FCN-16s	66.9%
FCN-8s	67.2%
CRF-RNN FCN-32s	69.2%
CRF-RNN FCN-16s	69.8%
CRF-RNN FCN-8s	70.2%
CRF-LSTM FCN-32s	69.7%
CRF-LSTM FCN-16s	69.9%
CRF-LSTM FCN-8s	70.5%

4.1. Comparison with CRF-RNN

We first compare performance of our proposed CRF-LSTM layer with the most-closely-related state-of-the-art CRF-RNN layer [44]. The authors in [44] appended one CRF-RNN layer to the end of a DCNN for the segmentation task. Following the same vein, we append our proposed CRF-LSTM layer to the end of the baseline FCN. The segmentation accuracies achieved for the FCN with one CRF-RNN layer appended are 69.2%, 69.8% and 70.2% for CRF-RNN FCN-32s, CRF-RNN FCN-16s and CRF-RNN FCN-8s respectively. The segmentation accuracies achieved for the FCN with one CRF-LSTM appended are 69.7%, 69.9% and 70.5% for CRF-LSTM FCN-32s, CRF-LSTM FCN-16s and CRF-LSTM FCN-8s respectively. The experimental results show that appending these layers helps to increase the segmentation accuracy. Both CRF-RNN and CRF-LSTM are comparable in terms of the segmentation accuracy boost amounts. The results are presented in Table 1.

4.2. Impact of early spatial/contextual modeling

The major advantage of our proposed CRF-LSTM layer is that it can handle the "gradient vanishing and exploding" effect. This effect limits the CRF-RNN layer to be pluggable only to the end of the deepnet, whereas the proposed CRF-LSTM layer can be pluggable into any layer of the deepnet. We experiment impact of location of the CRF-LSTM layer in the deepnet by shifting this layer to earlier stages. The CRF-LSTM layer is re-located to right after the input image and after each pool layer in the deepnet for testing. The results show shifting the CRF-LSTM layer to earlier stages achieves more accurate segmentation performance. This is because shifting the spatial/contextual modeling to early stages allows this information is spread throughout the whole network, which is more effective in modeling and embedding the spatial/contextual information. The results are presented in Table 2.

Table 2. Impact of the location of graphical modeling on the segmentation accuracy on the PASCAL-context dataset. The CRF-LSTM layer is re-located to various locations (after the input image and after each pool layer) in the deepnet for experiments. The results are in the form of segmentation accuracies of CRF-LSTM FCN-8s.

Image	Pool 1	Pool 2	Pool 3	Pool 4	Pool 5
72.3%	71.7%	71.3%	70.9%	70.6%	70.5%

4.3. Effects of deep CRF-LSTM

We are interested to see whether adding more CRF-LSTM layers will improve the performance. Section 4.2 has shown that when plugging one CRF-LSTM layer, the location right after the input image yields the best segmentation result (i.e. 72.3% for CRF-LSTM FCN-8s) in comparison with other locations. Hence we fix the first added CRF-LSTM layer at this location, then keep adding more CRF-LSTM layers to the right of the first CRF-LSTM layer. The experiment shows that adding the second CRF-LSTM layer increase the accuracy to 74.2% for CRF-LSTM FCN-8s. However, adding more than two CRF-LSTM layers drops the accuracy to 72.9% and 67.3% for CRF-LSTM FCN-8s for three and four layers respectively. Other outputs (CRF-LSTM FCN-32s and CRF-LSTM FCN-16s) observe the same effects. The results are presented in Table 3. The best configuration is achieved by adding two CRF-LSTM

Table 3. Effects of adding multiple CRF-LSTM layers. Adding two CRF-LSTM layers yields the best segmentation result. X-32s, X-16s and X-8s stand for CRF-LSTM FCN-32s, CRF-LSTM FCN-16s and CRF-LSTM FCN-8s respectively.

Approaches	Two layers	Three layers	Four layers
X-32s	72.1%	71.7%	66.4%
X-16s	73.8%	72.2%	67.1%
X-8s	74.2%	72.9%	67.3%

layers successively right after the input image as shown in Figure 4.

5. Conclusion

Incorporating graphical models into deep convolutional neural networks helps to model and embed spatial and contextual information into the network for the scene parsing task. In this paper, we propose a systematic approach to categorize the state-of-the-art trends. This categorization reveals the overall trend of incorporating towards moving CRFs and DCNNs more integrated. From this observation, we propose a novel approach to deeply integrate CRFs inside DCNN by modeling CRFs as a deepnet layer, which can be plugged in any layer of a DCNN. These layers are based on the LSTM models, which effectively represent and memorize the surrounding context. Plugging these CRF-LSTM layers in the early stage of the DCNN allows the spatial and contextual information to spread throughout the whole network, effectively embedding them and improving the segmentation accuracy.

Acknowledgment

This research was supported by an Australian Research Council (ARC) Linkage grant LP140100221

References

[1] Segmentation results: Pascal voc 2012. Accessed: 2016-07-01.

[2] J. Alvarez, Y. LeCun, T. Gevers, and A. Lopez. Semantic road segmentation via multi-scale ensembles of learned features. In *ECCV*, 2012.

[3] G. Bertasius, J. Shi, and L. Torresani. Semantic segmentation with boundary neural fields. In *CVPR*, 2016.

[4] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki. Scene labeling with lstm recurrent neural networks. In *CVPR*, pages 3547–3555, Jun 2015.

[5] 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.

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

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

[8] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[9] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Scene parsing with multiscale feature learning, purity trees, and optimal covers. In *ICML*, 2012.

[10] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *PAMI*, 2013.

[11] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *ICCV*, 2009.

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

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. *PAMI*, 2015.

[14] J. M. Hammersley and P. E. Clifford. Markov random fields on finite graphs and lattices. Unpublished manuscript, 1971.

[15] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015.

[16] B. Hariharan, P. Arbelez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*, 2014.

[17] X. He and S. Gould. An exemplar-based crf for multi-instance object segmentation. In *CVPR*, 2014.

[18] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computing*, 2006.

[19] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

[20] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[21] T. Kekec, R. Emonet, E. Fromont, A. Trmeau, and C. Wolf. Contextually constrained deep networks for scene labeling. In *BMVC*, 2014.

[22] A. Kolesnikov, M. Guillaumin, V. Ferrari, and C. Lampert. Closed-form approximate crf training for scalable image segmentation. In *ECCV*, 2014.

[23] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT Press, Cambridge, MA, 2009.

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

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

[26] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Associative hierarchical crfs for object class image segmentation. In *ICCV*, 2009.

[27] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[28] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[29] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[30] X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan. Semantic object parsing with graph lstm. In *ECCV*, pages 125–143, 2016.

[31] X. Liang, X. Shen, D. Xiang, J. Feng, L. Lin, and S. Yan. Semantic object parsing with local-global long short-term memory. *CoRR*, abs/1511.04510, 2015.

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

[33] F. Liu, G. Lin, and C. Shen. Crf learning with cnn features for image segmentation. *Pattern Recognition*, 2015.

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

[35] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.

[36] A. Lucchi, Y. Li, and P. Fua. Learning for structured prediction using approximate subgradient descent with working sets. In *CVPR*, 2013.

[37] 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.

[38] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 2015.

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

[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[41] J. Verbeek and B. Triggs. Scene segmentation with conditional random fields learned from partially labeled images. In *NIPS*, 2009.

[42] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *CVPR*, 2012.

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

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