Conditional Random Fields as Recurrent Neural Networks

Shuai Zheng[1], Sadeep Jayasumana[1] Bernardino Romera-Paredes[1] Vibhav Vineet[1,2] Zhizhong Su[3] Dalong Du[3] Chang Huang[3] Philip H. S. Torr[1]
[1]Department of Engineering Science, Oxford University. [2]Stanford University. [3]Baidu

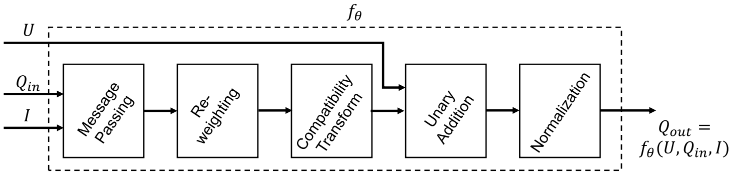

Figure 1: **A mean-field iteration as a CNN.** A single iteration of the mean-field algorithm can be modelled as a stack of common CNN layers.

1 Introduction

Pixel-level labelling tasks, such as semantic segmentation, play a central role in image understanding. Recent approaches [4, 8] have attempted to harness the capabilities of deep learning techniques for image recognition to tackle pixel-level labelling tasks. One central issue in this methodology is the limited capacity of deep learning techniques to delineate visual objects. To solve this problem, we introduce a new approach that combines the strengths of Convolutional Neural Networks (CNNs) and Conditional Random Fields (CRFs)-based probabilistic graphical modelling. To this end, we formulate mean-field approximate inference for the fully-connected CRFs with Gaussian pairwise potentials as Recurrent Neural Networks (RNNs). This network, called CRF-RNN, is then plugged in as a part of a CNN to obtain a deep network that has advantages of both CNNs and CRFs. Importantly, our system fully integrates CRF modelling with CNNs, making it possible to train the whole deep network end-to-end with the usual propagation algorithm, avoiding offline post-processing methods for object delineation.

We apply the proposed method to the problem of semantic image segmentation, obtaining top results on the challenging Pascal VOC benchmark and Pascal Context dataset.

2 A Mean-field Iteration as a Stack of CNN Layers

A key contribution of this paper is to show that the mean-field CRF inference can be reformulated as a RNN. To this end, we first consider individual steps of the mean-field algorithm summarized in Algorithm 1, and describe them as CNN layers as shown in Figure 1. Our contribution is based on the observation that filtering-based mean-field approximate inference for fully-connected CRFs relies on applying Gaussian spatial and bilateral filters on the mean-field approximates in each iteration. Unlike the standard convolutional layer in a CNN, in which filters are fixed after the training stage, we use edge-preserving Gaussian filters [12, 16], coefficients of which depend on the original spatial and appearance information of the image. These filters have the additional advantages of requiring a smaller set of parameters, despite the filter size being potentially as big as the image.

We reformulate the steps of the inference algorithm as CNN layers, it is essential to be able to calculate error differentials in each layer w.r.t. its inputs in order to be able to backpropagate the error differentials to previous layers during training. In our formulation, CRF parameters such as the weights of the Gaussian kernels and the label compatibility function can be optimized automatically during the training of the full network.

Once the individual steps of the algorithm are broken down as CNN layers, the full algorithm can then be formulated as an RNN. We illustrate this in Algorithm 1, where we use $U_i(l)$ to denote the negative of the unary energy, i.e., $U_i(l) = -\psi_u(X_i = l)$. In the conventional CRF setting, this input $U_i(l)$ to the mean-field algorithm is obtained from an independent classifier.

Algorithm 1 Mean-field in fully-connected CRFs [6], broken down to common CNN operations.

```
$Q_i(l) \leftarrow \frac{1}{Z_i} \exp(U_i(l))$ for all $i$                         ▷ Initialization
while not converged do
    $\tilde{Q}_i^{(m)}(l) \leftarrow \sum_{j \neq i} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l)$ for all $m$
                                                                              ▷ Message Passing
    $\check{Q}_i(l) \leftarrow \sum_m w^{(m)} \tilde{Q}_i^{(m)}(l)$
                                                                    ▷ Weighting Filter Outputs
    $\hat{Q}_i(l) \leftarrow \sum_{l' \in \mathcal{L}} \mu(l, l') \check{Q}_i(l')$
                                                                   ▷ Compatibility Transform
    $\breve{Q}_i(l) \leftarrow U_i(l) - \hat{Q}_i(l)$
                                                                ▷ Adding Unary Potentials
    $Q_i \leftarrow \frac{1}{Z_i} \exp\left(\breve{Q}_i(l)\right)$
                                                                              ▷ Normalizing
end while
```

3 The End-to-end Trainable Network

We present an end-to-end deep learning system for semantic image segmentation. We first explain how repeated mean-field iterations can be organized as an RNN. We then draw the completing picture for our system.

3.1 CRF as RNN

As shown in Fig. 1, one iteration of the mean-field algorithm can be formulated as a stack of common CNN layers. We use the function f_θ to denote the transformation done by one mean-field iteration: given an image I, pixel-wise unary potential values U and an estimation of marginal probabilities Q_{in} from the previous iteration, the next estimation of marginal distributions after one mean-field iteration is given by $f_\theta(U, Q_{\text{in}}, I)$. The vector $\theta = \{w^{(m)}, \mu(l, l^{'})\}, m \in \{1, ..., M\}, l, l^{'} \in \{l_1, ..., l_L\}$ represents the CRF parameters.

Multiple mean-field iterations can be implemented by repeating the above stack of layers in such a way that each iteration takes Q value estimates from the previous iteration and the unary values in their original form. This is equivalent to treating the iterative mean-field inference as a RNN. Using the notation in the figure, the behaviour of the network is given by the following equations where T is the number of mean-field iterations:

$$H_1(t) = \begin{cases} \text{softmax}(U), & t = 0 \\ H_2(t-1), & 0 < t \leq T, \end{cases} \tag{1}$$

$$H_2(t) = f_\theta(U, H_1(t), I), \quad 0 \leq t \leq T, \tag{2}$$

$$Y(t) = \begin{cases} 0, & 0 \leq t < T \\ H_2(t), & t = T. \end{cases} \tag{3}$$

We name this RNN structure CRF-RNN. Parameters of the CRF-RNN are the same as the mean-field parameters and denoted by θ here. Since the calculation of error differentials w.r.t. these parameters in a single iteration can be learnt in the RNN setting using the standard backpropagation through time algorithm [10, 14]. It was shown in [6] that the mean-field iterative algorithm for fully-connected CRF converges in less than 10 iterations. Furthermore, in practice, after about 5 iterations, increasing the number of iterations usually does not significantly improve results [6]. Therefore, it does not suffer from the vanishing and exploding gradient problem inherent to deep RNNs [1, 13].

3.2 Completing the Picture

Our approach comprises a fully convolutional network stage, which predicts pixel-level labels without considering structure, followed by a CRF-

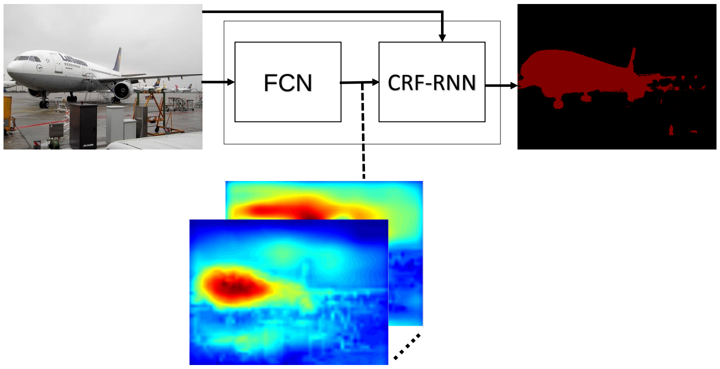

Figure 2: **The End-to-end Trainable Network.** Schematic visualization of our full network which consists of a CNN and the CNN-CRF network. Best viewed in colour.

RNN stage, which performs CRF-based probabilistic graphical modelling for structured prediction. The complete system, therefore, unifies strengths of both CNNs and CRFs and is trainable end-to-end using the backpropagation algorithm [7] and the Stochastic Gradient Descent (SGD) procedure. During training, a whole image (or many of them) can be used as the mini-batch and the error at each pixel output of the network can be computed using an appropriate loss function such as the softmax loss with respect to the ground truth segmentation of the image. We used the FCN-8s architecture of [8] as the first part of our network, which provides unary potentials to the CRF. This network is based on the VGG-16 network [15] but has been restructured to perform pixel-wise prediction instead of image classification. The complete architecture of our network, including the FCN-8s part can be found in the our publicly available source code and models.

In the forward pass through the network, once the computation enters the CRF-RNN after passing through the CNN stage, it takes T iterations for the data to leave the loop created by the RNN. Once the output Y leaves the loop, next stages of the deep network after the CRF-RNN can continue the forward pass. In our setup, a softmax loss layer directly follows the CRF-RNN and terminates the network.

During the backward pass, once the error differentials reach the CRF-RNN's output Y, they similarly spend T iterations within the loop before reaching the RNN input U in order to propagate to the CNN which provides the unary input. In each iteration inside the loop, error differentials are computed inside each component of the mean-field iteration as described in Section 2. We note that unnecessarily increasing the number of mean-field iterations T could potentially result in the vanishing and exploding gradient problems in the CRF-RNN. We, however, did not experience this problem during our experiments.

4 Experiments

In order to evaluate our approach with existing methods under the same circumstances, we conducted two main experiments with the Pascal VOC 2012 dataset, followed by a qualitative experiment.

In the first experiment, following [8, 9, 11], we used a training set consisted of VOC 2012 training data (1464 images), and training and validation data of [5], which amounts to a total of 11,685 images. After removing the overlapping images between VOC 2012 validation data and this training dataset, we were left with 346 images from the original VOC 2012 validation set to validate our models on. We call this set the reduced validation set in the sequel. Annotations of the VOC 2012 test set, which consists of 1456 images, are not publicly available and hence the final results on the test set were obtained by submitting the results to the Pascal VOC challenge evaluation server [3]. Regardless of the smaller number of images, we found that the relative improvements of the accuracy on our validation set were in good agreement with the test set.

As a first step we directly compared the potential advantage of learning the model end-to-end with respect to alternative learning strategies. These are plain FCN-8s without applying CRF, and with CRF as a postprocessing method disconnected from the training of FCN, which is comparable to the approach described in [2] and [11]. In all cases, the resolution of the input and the output of FCN-8s is 500 × 500, the crop layer in the FCN-8s aligns the input image and the output feature map. This feature map is used as the input of the CRF-RNN layers. The results are reported in Table 1 and show a clear advantage of the end-to-end strategy over the offline application of CRF as a post-processing method. This can be attributed to the fact that during the SGD training of the CRF-RNN, the CNN component and the CRF component learn how to co-operate with each other to produce the optimum output of the whole network.

Method	Without COCO	With COCO
Plain FCN-8s	61.3	68.3
FCN-8s and CRF disconnected	63.7	69.5
End-to-end training of CRF-RNN	69.6	72.9

Table 1: Mean IU accuracy of our approach, CRF-RNN, compared with similar methods, evaluated on the reduced VOC 2012 validation set.

Our approach achieves 74.7% mean IOU score over 20 classes on the test set of Pascal VOC 2012, and 39.28% mean IOU score over 59 classes on Pascal Context dataset.

5 Conclusions

The paper formulates a fully-connected CRFs as RNNs. In semantic image segmentation application, we show this lead to further improvement over the fully-convolutional neural networks.

Acknowledgement This work was supported by grants EP/I001107/2 and ERC 321162-HELIOS. We thank Caffe team, Baidu IDL, and Oxford ARC for their support. We gratefully acknowledge GPU donations from NVIDIA.

[1] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 1994.

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.

[3] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015.

[4] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE TPAMI*, 2013.

[5] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *ECCV*, 2014.

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

[7] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86 (11):2278–2324, 1998.

[8] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE CVPR*, 2015.

[9] Mohammadreza Mostajabi, Payman Yadollahpour, and Gregory Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *IEEE CVPR*, 2015.

[10] Michael C. Mozer. A Focused Backpropagation Algorithm for Temporal Pattern Recognition. *Complex Systems*, 3(1):349–381, 1989.

[11] George Papandreou, Liang-Chieh Chen, Kevin Murphy, and Alan L. Yuille. Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. In *arXiv:1502.02734*, 2015.

[12] Sylvain Paris and Fredo Durand. A fast approximation of the bilateral filter using a signal processing approach. *IJCV*, 81(1):24–52, 2013.

[13] Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *ICML*, 2013.

[14] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning Internal Representations by Error Propagation. *Parallel distributed processing: explorations in the microstructure of cognition*, 1:318–362, 1986.

[15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *arXiv:1409.1556*, 2014.

[16] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *IEEE CVPR*, 1998.