Scene Labeling with LSTM Recurrent Neural Networks

Wonmin Byeon [1 2], Thomas M. Breuel [1], Federico Raue [1 2], Marcus Liwicki [1 2]
[1] University of Kaiserslautern, Germany. [2] German Research Center for Artificial Intelligence (DFKI), Germany

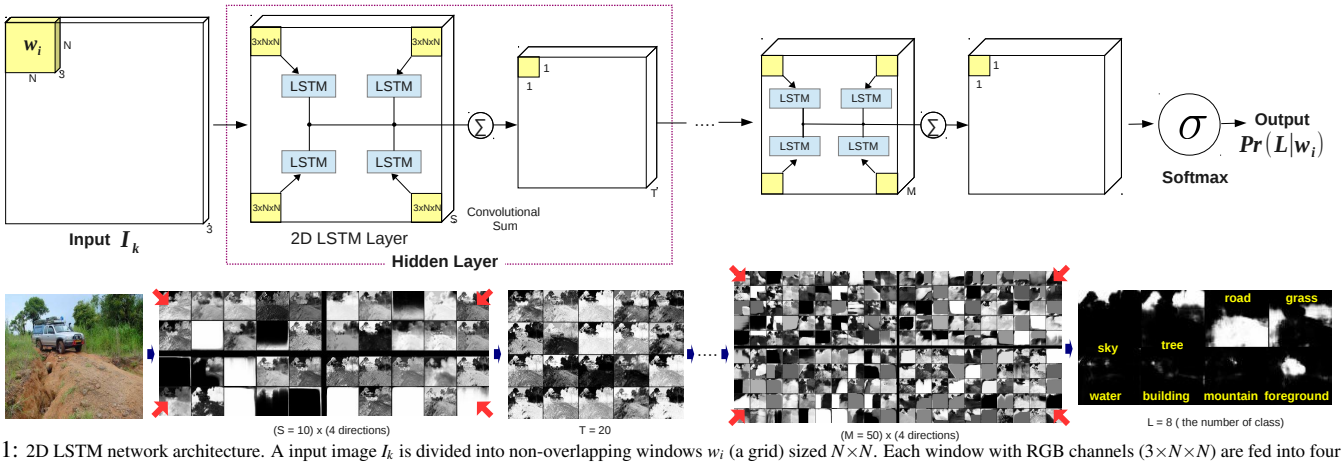

Figure 1: 2D LSTM network architecture. A input image I_k is divided into non-overlapping windows w_i (a grid) sized $N \times N$. Each window with RGB channels ($3 \times N \times N$) are fed into four separate LSTM memory blocks with size S. Each LSTM block is connected to its surrounding directions, i.e., left-top, left-bottom, right-top, and right-bottom. The output of each LSTM block is convoluted separately with size T, then summed and squashed by the Hyperbolic tangent ($tanh$). From this step, all the information from the different directions is accumulated and passed to the next layer. At the last layer, the outputs of the final LSTM blocks are summed up and sent to the *softmax* layer. Finally, the networks output the class probabilities for each input window. The bottom images are corresponding outputs for each layer.

Introduction

The scene labeling task consists of partitioning the meaningful regions of an image and labeling pixels with their regions. This paper addresses the problem of pixel-level segmentation and classification of scene images with a entirely learning-based approach using Long Short Term Memory (LSTM) recurrent neural networks, which are commonly used for sequence classification. We investigate two-dimensional (2D) LSTM networks for natural scene images taking into account the complex spatial dependencies of labels. Many prior methods generally have required separate classification and image segmentation stages and/or pre- and post-processing. In our approach, classification, segmentation, and context integration are all carried out by 2D LSTM networks, allowing texture and spatial model parameters to be learned within a single model. The networks efficiently capture local and global contextual information over raw RGB values and adapt well for complex scene images.

The networks are divided into the three main layers: Input layer, hidden layer, and output layer. The hidden layer consists of 2D LSTM layer and feed-forward layer, and is stacked as deep networks. The architecture of 2D LSTM networks is illustrated in Figure 1. The input images was split into grids, then pass to the LSTM subnets which allow to easily memorize the context information. Each LSTM memory block scans on all directions (left-top, left-bottom, right-top, and right-bottom), then the outputs are combined together in feed-forward layer. Finally, the outputs from the last hidden layer is normalized with the *softmax* function. As an objective function, we apply the negative log probability (i.e., cross entropy error function) with *Probabilistic target coding scheme*.

Experiments

Table 1 compares the performance of LSTM networks with current state-of-the-art methods on the Stanford Background dataset and the SIFT Flow dataset, and selected examples of labeling results from the Stanford dataset are shown in Figure 2.

Our approach, which has a much lower computational complexity than prior methods, achieves state-of-the-art performance over the Stanford Background and the SIFT Flow datasets. In fact, if no pre- or post-processing is applied, LSTM networks outperform other state-of-the-art approaches. Hence, only with a single-core Central Processing Unit (CPU), the running time of our approach is equivalent or better than the compared state-of-the-art approaches which use a Graphics Processing Unit (GPU).

Figure 2: The results of scene labeling on the Stanford Background dataset. First row: input image; Second row: ground-truth; Third row: predicted image. Colors on images indicate labels — identical colors on ground-truth and predicted images indicate a correct labeling

Table 1: Pixel and averaged per class accuracy comparison on the Stanford Background dataset and the SIFT Flow dataset (in %). B and UB indicate balancing and unbalancing of class frequency of input images, respectively. Balancing the class frequencies would improve the class-average accuracy, but is not realistic for scene labeling in general. The performance of recurrent CNNs (RCNNs) reported here is from two instances. CT indicates the averaged computing time per image.

Method	Pixel Acc.	Class Acc.	CT (sec.)	# params
Stanford				
Single-scale ConvNet [1]	66	56.5 (B)	0.35 (GPU)	-
ACNNs [3]	71.97	66.16 (B)	-	701K
RCNNs [4]	76.2	67.2 (UB)	1.1 (GPU)	-
LSTM networks (window 5×5)	77.78	**69.60** (UB)	1.4 (CPU)	173K
LSTM networks (window 3×3)	**78.56**	68.79 (UB)	3.7 (CPU)	155K
SIFT Flow				
Multi-scale ConvNet [1]	67.9	**45.9** (B)	-	-
ACNNs [3]	49.39	44.54 (UB)	-	1225K
RCNNs [4]	65.5	20.8 (UB)	-	-
LSTM networks (window 5×5)	68.74	**22.59** (UB)	1.2 (CPU)	178K
LSTM networks (window 3×3)	**70.11**	20.90 (UB)	3.1 (CPU)	168K

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

[2] S Hochreiter and J Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

[3] Taygun Kekeç, Rémi Emonet, Elisa Fromont, Alain Trémeau, Christian Wolf, and France Saint-Etienne. Contextually constrained deep networks for scene labeling. In *In BMVC*, 2014.

[4] Pedro Pinheiro and Ronan Collobert. Recurrent convolutional neural networks for scene labeling. In *In ICML*, 2014. URL http://jmlr.org/proceedings/papers/v32/pinheiro14.pdf.

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.