Microscopy Cell Segmentation via Convolutional LSTM Networks

Assaf Arbelle, Tammy Riklin Raviv

Departent of Electrical and Computer Engineering, Ben Gurion University

1 Methods

We address individual cells' segmentation from microscopy sequences using C-LSTM as thouroghly described in our article [2]. The main challenge in this type of problems is not only foreground-background classification but also the separation of adjacent cells. As in [1] we suggest to enhance individual cells' delineation by a partitioning of the image domain $\Omega \in \mathbb{R}^2$ into three classes: foreground, background, and edges. We set $C = \{0, 1, 2\}$ to denote these classes, respectively. Let $\{I_t\}_{t=1}^{\tau}$ be the input image sequence, where $I_t : \Omega \to \mathbb{R}$ is a grayscale image. We define a network f_Θ with parameters Θ as follows:

$$(o_t, h_t) = f_\Theta(I_t, o_{t-1}, h_{t-1}), \quad c \in \mathcal{C} \tag{1}$$

where, the output $o_t : \Omega \to \mathbb{R}^3$ is a three-dimensional feature vector corresponding to each input pixel $\mathbf{x} \in \Omega$ and h_t are the hidden variables of the network. We define the pixel label probabilities using the softmax equation:

$$p(c|o_t(\mathbf{x})) = \frac{\exp\{[o_t(\mathbf{x})]_c\}}{\sum_{c' \in \mathcal{C}} \exp\{[o_t(\mathbf{x})]_{c'}\}} \tag{2}$$

Network Architecture The proposed network f_Θ incporporates C-LSTM blocks into the U-Net architecture. While the U-Net [5] and C-LSTM [6] are widely used, their composition is suggested here for the first time and is shown to be powerful. The U-Net architecture, built as an encoder-decoder with skip connections, is able to extract meaningful descriptors at multiple image scales. However, this alone does not account for the cell specific history that can significantly support the segmentation. The introduction of C-LSTM blocks into the network's decoder allows considering past cell appearances by holding their compact representations in the memory units. The network is fully convolutional and, therefore, can be used with any image size[1] during both training and testing. Figure 1 illustrates the full network architecture detailed in Section 2.

[1] In order to avoid artefacts it is preferable to use image sizes which are multiples of eight due to the three max-pooling layers.

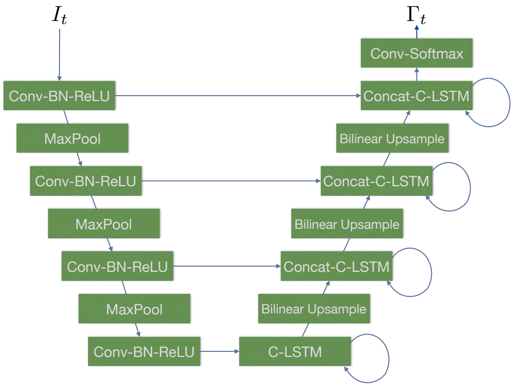

Fig. 1: The U-LSTM network architecture. The down-sampleing path (left) consists of convolutional layers, followed by batch normalization and ReLU activations, the output is then down-sampled and passed to the next layer. The up-sampleing path (right) consists of a concatination of the input from the lower layer with the parallel layer from the down-sampeling path followed by a C-LSTM layer and a bilinear up-sampeling interpolation.

Training During the training phase the network is presented with a full sequence and manual annotations $\{I_t, \mathit{\Gamma}_t\}_{t=1}^{T}$, where $\mathit{\Gamma}_t : \mathit{\Omega} \rightarrow [0, 1, 2]$ are the ground truth (GT) labels for the pixels in image I_t. The network is trained using Truncated Back Propagation Through Time (TBPTT). At each back propogation step the network is unrolled to τ time-steps. The loss is defined using the cross-entropy loss:

$$L = \sum_{t'=t}^{t+\tau} \sum_{\mathbf{x} \in \mathit{\Omega}} p(\mathit{\Gamma}_{t'}(\mathbf{x}) | o_{t'}(\mathbf{x})) \tag{3}$$

and the weights are updated using gradient descent. Note that this loss differs from the U-Net loss, where boundary pixels are labelled as background weighted by their proximity to the two nearest cells [5]. Here, since a separate class for boundary pixels is defined, weighting is not required. Figure 2 show a visual example of the annotations and the network output. Comparisons of the two losses using the proposed network architecture is reported in the supplementary material.

2 Implementation Details

Architecture The network comprises four down-sampling blocks and four up-sampling blocks. Each block in the down-sampleing branch is composed of a convolutional layer, batch normalization [4], and leaky ReLU. The up-sampling blocks consist of a bilinear interpolation, a concatenation with the parallel down-sample block and a C-LSTM. All convolutional layers use kernel size 3×3 with

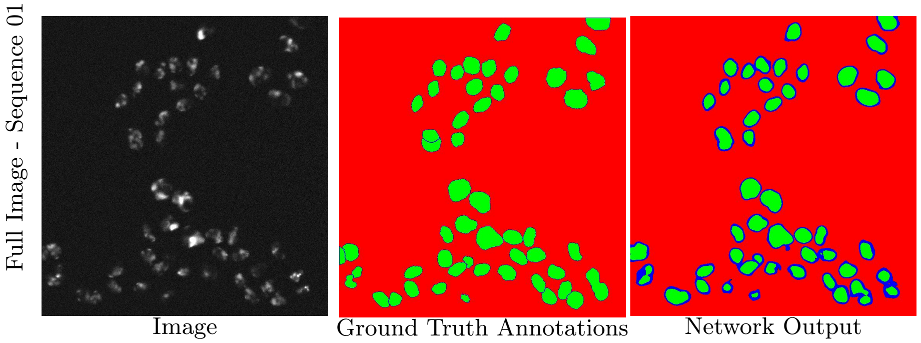

Fig. 2: Annotation Example: The visualization of the annotations as a three-class segmentations. On the left an image from the Fluo-N2DH-SIM+ data set, the center image is the GT annotation and the right image is the network output. The colors red, green and blue represent the three classes, background, foreground and cell contour, respectively. It is evident that the network learned to classify the contour of the cells as class edge allowing the separation of individual cells.

layer depths $(32, 64, 128, 256)$. All maxpool layers use kernel size 2×2 without overlap. All C-LSTM kernels are of size 7×7 with layer depths $(32, 64, 128, 256)$. The last convolutional layer uses kernel size 1×1 with depth 3 followed by a softmax layer to produce the final probabilities (see Figure 1).

Training Regime We trained the networks for approximately $100K$ iterations with an RMS-Prop optimizer [3] as implemented in Tensorflow (version 1.4.0) with learning rate of 0.0001. The unroll length parameter τ was set to five (Section 1) and the batch size was set to three sequences.

References

1. Assaf Arbelle and Tammy Riklin Raviv. Microscopy cell segmentation via adversarial neural networks. *arXiv preprint arXiv:1709.05860*, 2017.
2. Assaf Arbelle and Tammy Riklin Raviv. Microscopy cell segmentation via convolutional lstm networks. *arXiv preprint arXiv:1805.11247*, 2018.
3. Geoffry Hinton. Rms prop: Coursera lectures slides, lecture 6.
4. Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456, 2015.
5. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *arXiv preprint arXiv:1505.04597*, 2015.
6. Shi Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.