CELL PROPOSAL NETWORK FOR MICROSCOPY IMAGE ANALYSIS

Saad Ullah Akram[1,2], *Juho Kannala*[3], *Lauri Eklund*[2,4], *and Janne Heikkilä*[1]

[1] Center for Machine Vision Research, [2] Biocenter Oulu, [4] Oulu Center for Cell-Matrix Research, and [4] Faculty of Biochemistry and Molecular Medicine, University of Oulu, Finland
[3] Department of Computer Science, Aalto University, Finland

ABSTRACT

Robust cell detection plays a key role in the development of reliable methods for automated analysis of microscopy images. It is a challenging problem due to low contrast, variable fluorescence, weak boundaries, conjoined and overlapping cells, causing most cell detection methods to fail in difficult situations. One approach for overcoming these challenges is to use cell proposals, which enable the use of more advanced features from ambiguous regions and/or information from adjacent frames to make better decisions. However, most current methods rely on simple proposal generation and scoring methods, which limits the performance they can reach. In this paper, we propose a convolutional neural network based method which generates cell proposals to facilitate cell detection, segmentation and tracking. We compare our method against commonly used proposal generation and scoring methods and show that our method generates significantly better proposals, and achieves higher final recall and average precision.

***Index Terms*—** cell proposals, cell detection, cell tracking, deep learning, fully convolutional network

1. INTRODUCTION

Light microscopy is the most common method to investigate cells and robust cell detection in microscopy images plays a key role in most cell segmentation and tracking methods, which are necessary to gain insights into cell functions, tissue development and disease progression. Only using human vision and labor based approaches cell detection is impractical or impossible due to very large numbers of cells, and for computer-based solutions it is a very challenging problem due to high cell density, low contrast, variable fluorescence, weak boundaries, strong gradients inside cell bodies, deformable cell shapes, and conjoined and overlapping cells. These factors frequently result in ambiguous regions and cause most cell detection methods to make mistakes. One approach for reducing these errors is to generate a relatively large set of cell proposals such that they have a very high recall and then select the optimal set among them. This approach enables the use of more advanced reasoning and temporal information when finding the optimal set of proposals, leading to improvement in performance.

In recent years, proposal generation methods have become very popular in object detection and all current top ranked methods use object proposals [1]. However, the challenges in bio-medical image analysis are very different from general object detection and most object proposal generation methods do not transfer well when applied directly. Recently, few methods have been presented which utilize cell proposals for cell detection [2, 3, 4] and tracking [5, 6, 7]. These cell proposal methods fall into 3 categories: thresholding [2, 3], shape matching [7, 5] and super-pixel merging [6, 4].

Thresholding based methods (e.g. MSER) [2, 3] assume that cell centers are brighter than their boundaries and there exists some optimal threshold at which individual cells can be segmented as separate proposals. In many challenging sequences, this assumption does not hold true and these methods cannot generate good proposals. One method for overcoming this limitation is to allow each proposal to contain more than one cell [8], while another is to transform the images so that this assumption becomes true [8]. Shape based methods either use multi-scale blob detection [7] or multi-scale ellipse fitting [5] to detect and segment cells. These methods can generate good proposals when cells have round or elliptical shapes but do not work well for general cell shapes. Super-pixel merging methods [6, 4] do not inherently make any of the above assumptions, so they can handle arbitrary shapes, but they still need some criteria for merging superpixels, which can be challenging due to strong gradients within cells and weak gradients between cells.

None of the above mentioned cell proposal generation methods provide a natural way of ranking or scoring the proposals, so a second stage is used to extract some features from each proposal region and these features are used to score it. These features are usually hand crafted and consist of basic appearance and shape statistics, including area, mean intensity [6], histogram of proposal boundary [2], etc. These features are then used by random forest [6], gradient boosted trees [5] or SVM [7, 3, 2] classifier to compute the probability of the proposal being a cell.

After proposal generation, cell detection is posed as the selection of proposals which maximize the combined score

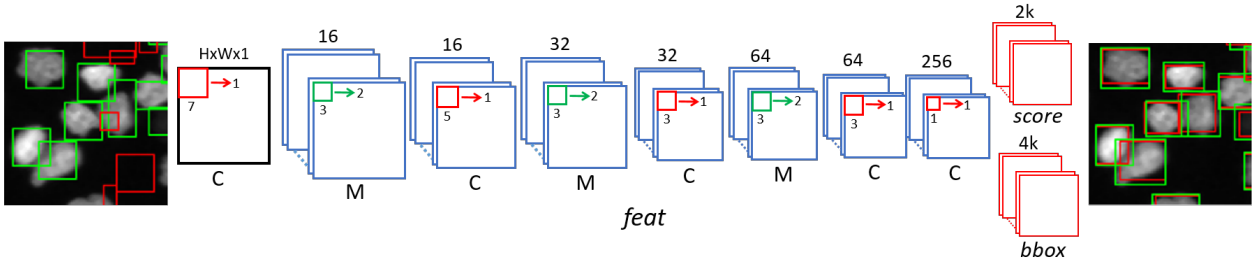


Fig. 1: Cell Proposal Network. A selected area from *Fluo-N2DL-HeLa* dataset is shown. Left: Positive and negative training anchors are marked. Right: Top ranking cell proposals and ground truth bounding boxes are shown.

under the constraint that no two selected proposals conflict (overlap). This optimization is performed either using Integer Linear Programming [3, 4] or dynamic programming [2]. In cell tracking applications, proposals in adjacent frames are linked with each other and then the solution providing trajectories of cells along with selected proposals is obtained either by Linear Integer Programming [5, 6] or by iteratively finding the shortest path [7].

Proposal generation and their scoring is a bottleneck in the performance of these tracking and detection methods. A better proposal generation and/or ranking method can lead to significant improvement in performance of these methods especially on more challenging data-sets. Recently, [9] and [10] have shown that convolutional neural networks can be used to generate very high quality (high recall and precision) object proposals in context of generic object detection.

In this work, we utilize an approach similar to [9] to generate cell proposals for fluorescence microscopy images. That is, we present a deep learning based cell proposal generation network, which provides cell proposals and their scores. Our method bypasses the need for manual selection and design of features for computing cell probabilities. Our novel contributions include: (1) a network for cell proposal generation and (2) a data expansion method for augmenting training data with weakly annotated cell images. We evaluate our method against existing proposal generation and ranking methods and show that it outperforms them. We also show that our method outperforms state of the art cell detection method [2] by greedily selecting non-conflicting proposals.

2. METHOD

Network Structure: Our cell proposal network (CPN) uses a fully convolutional neural network to predict bounding boxes for cell proposals and their scores - probability of them being cells. It consists of two parts and is shown in Fig. 1. First part, *feat*, extracts 256-dimensional feature vectors from 53x53 rectangular regions (with stride of 8) in the input image. This part of our network is based on Zeiler and Fergus model [11] and was selected experimentally. The second part (shown in red) consists of two parallel fully connected layers, *score* and *bbox*. *bbox* layer outputs proposal bounding boxes, while *score* layer outputs their scores. Both fully connected layers (implemented as 1x1 convolution layers) slide over the *feat* output and provide multiple proposals for each pixel in this feature map. CPN uses k (=6) anchors [9] at each pixel in *feat* output to propose cells at multiple scales and aspect ratios. Anchors are bounding boxes placed in the input image at the center of receptive field of each pixel in *feat* output. The outputs of *bbox* layer, b_i, are parameters of predicted bounding boxes, $b = (x, y, w, h)$, relative to anchor bounding boxes, $b_a = (x_a, y_a, w_a, h_a)$ [9].

$$b_i = ((x - x_a)/w_a, (y - y_a)/h_a, log(w/w_a), log(h/h_a)) \quad (1)$$

Training: We use overlap of anchors with cell bounding boxes to generate positive and negative samples. Anchors having the highest intersection over union overlap (IoU) with each ground truth (GT) cell bounding box are used as positive samples. Negative samples are selected randomly from anchors having maximum IoU below 0.4 with all GT cell bounding boxes. Each training batch consists of a single frame, with equal number of positive and negative samples. All other anchors are not used for training and hence do not contribute to the multi-task loss function [9]:

$$L(p_i, b_i) = L_{score}(p_i, p_i^*) + \lambda p_i^* L_{bbox}(b_i, b_i^*) \quad (2)$$

where L_{bbox} is smooth-L_1 loss [12] and penalizes differences between predicted (b_i) and ground truth (b_i^*) bounding box parameters. L_{score} is soft-max classification loss for two classes, cell and background. p_i is the probability of bounding box, b_i, being a cell. Ground truth label, p_i^*, activates the L_{bbox} loss for positive samples and disables it otherwise. λ(=10) balances the bounding box regression loss relative to the classification loss.

Data Expansion: Our network needs bounding boxes for training but the ground truth data that is typically available has very few cells with bounding boxes. This limited training data does not cover cell appearance and shape variation sufficiently for the network to learn the desired invariances. However, in a typical dataset all cells in a sequence have a ground truth marker (few connected pixels identifying each cell uniquely) inside their body. We use these markers to obtain cell bounding boxes and increase training data. We first segment cells from background using graph cuts. This binary

segmentation fails to detect some very dark cells, so we place an average cell sized bounding box centered at their marker location. Then we use ground truth markers and marker-controlled watershed to split cell clusters in initial segmentation. Some watershed regions are very small due to errors in cluster splitting; we remove these regions from training data. Cells in microscopy images, unlike objects in natural images, can be present in any orientation so we use rotations and flips to further increase training data.

Post-processing: CPN evaluates $\sim H \times W \times k/64$ proposals for an image of size $H \times W$ and can generate multiple proposals for some cells. We use non-maxima suppression (IoU=0.5) to remove lower scored duplicate proposals.

Our deep learning network provides a bounding box for cell proposals but in most bio-medical image analysis applications, cell segmentation proposals are needed. We obtain the required segmentation mask by thresholding. The proposal bounding boxes predicted by CPN are not very precisely localized so we first expand the bounding boxes by 3 pixels on each side. Then, we threshold the expanded bounding box regions by using their mean intensities as the threshold. Morphological closing and hole filling are used to refine the proposal segmentations. Some proposed bounding boxes may contain multiple objects after the post-processing steps; only the largest object in a bounding box is retained as the segmentation mask.

3. EXPERIMENTS

Dataset: We evaluate our method on *Fluo-N2DL-HeLa* data-set from ISBI cell tracking challenge [13]. This data-set contains 2 time-lapse sequences (92 frames each) of fluorescent HeLa cells cultured and imaged on two dimensional surface. The ground truth (GT) for this data-set contains markers for all cells in all frames and segmentation masks for all cells in 2 frames from each sequence.Some of the challenges with this data-set are: many cell clusters, frequent cell divisions, low contrast, variation in cell sizes and intensities.

Baseline: We compare our *CPN* with two cell proposal generation methods: *BLOB* and *MSER*. ***BLOB*** **[7]** uses multiple filter banks, covering common cell scales and aspect ratios, to detect and segment cell proposals. ***MSER*** **[2, 14]** finds stable connected components (area does not vary across a range of thresholds) and uses these as cell proposals.

Both proposal generation methods do not provide any ranking so we use the following three feature sets for ranking proposals. ***Set A*** **[2]** contains three intensity and one shape histogram and was proposed as part of a cell detection method. ***Set B*** **[7]** contains few image moments and some basic shape features, e.g. perimeter, solidity, etc. ***Set C*** **[6]** includes few intensity statistics of cell proposal and its dilation. Both feature set *B* and *C* were proposed to compute the probability of a proposal being a cell in joint cell detection and tracking pipelines.

Evaluation Criteria: We use two metrics to evaluate proposals as either true positive (TP) or false positive (FP). First metric evaluates segmentation masks; a proposal is considered TP if its mask has intersection over union overlap (IoU) > 0.5 with any unmatched ground truth (GT) cell segmentation mask, otherwise the proposal is considered FP. GT cells which remain unmatched are false negatives (FN). Since we have only four frames with GT segmentation masks but have access to cell markers for all frames, so we propose a second evaluation metric which considers the number of cell markers inside a proposal's segmentation mask. A proposal is considered TP if it contains only one GT cell marker inside its body and that marker is unmatched, otherwise it is considered FP. Cell markers which do not occur alone inside any evaluated proposal are considered FN.

The number of cells and hence the difficulty of generating proposals in a frame varies a lot [1] so we generate proposals for all frames in the data-set, order them by their score and then evaluate them as either TP or FP, obtaining a pair of recall (R = TP/(TP+FN)) and precision (P = TP/(TP+FP)) values after evaluating each proposal. We report these values using precision-recall curves along with average precision (AP) - area under precision-recall curves.

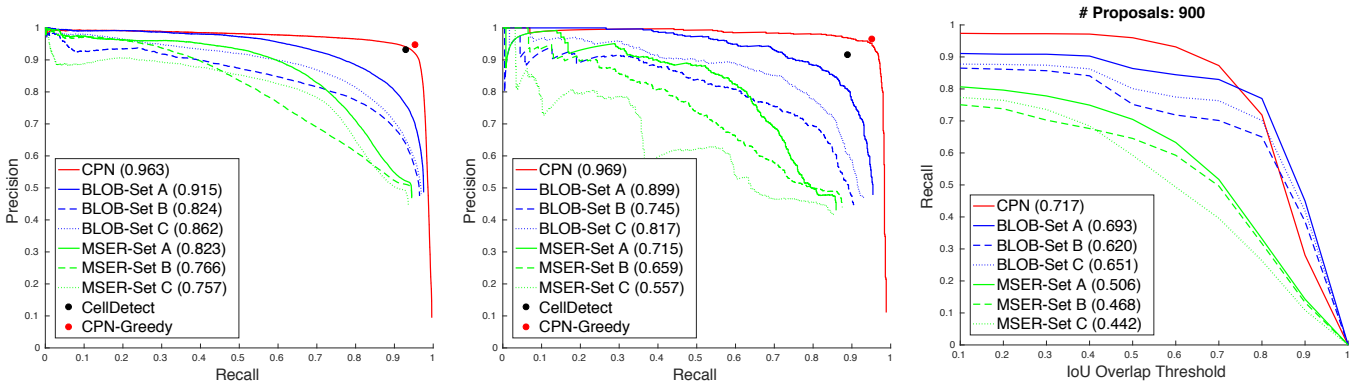
Implementation Details: Same pre-processing (median filtering) is used for all comparison methods. For *BLOB* and *MSER*, we generate the proposals, extract above mentioned feature sets and normalize each feature set to have zero-mean and unit-variance. Proposals with one GT cell marker inside them are labeled as positive samples, while the rest are labeled as negative samples. Then, a 2-class random forest classifier is trained to predict the probability of a proposal being a cell. Once we have the proposal scores, we use non-maxima suppression to get rid of duplicate proposals. We use same non-maxima suppression settings (IoU = 0.5) for all methods.

CPN's weights are initialized randomly from a Gaussian distribution with zero-mean and 0.01 standard deviation. We use learning rate of 0.001 for first 25k iterations, then it is reduced to 0.0001 for next 15k iterations.

For all methods, one sequence is used for training and the other one for testing; this is repeated for both sequences. Then proposals from both sequences are combined, sorted by their score and evaluated as either TP or FP.

Results: Fig. 2a shows the precision-recall curves when GT cell markers are used for evaluation. Average precision (AP) is shown in the legend along with the combination of proposal generation method and feature set used for ranking it. *CPN* has highest AP (0.963), highest final recall (0.996) and higher precision for all recall values. Combination of *BLOB* proposals and *Set A* features has almost the same precision as *CPN* for low recall values, however the difference between them increases with recall. *CPN* maintains precision above 0.95 for recall up to 0.9 indicating that it is quite accurate at picking out easy and moderately difficult cells. Its

[1] In our data-set, the number of cells in a frame varies from 43 to 363.



(a) Precision vs Recall (markers) (b) Precision vs Recall (at IoU = 0.5) (c) Recall vs IoU threshold

Fig. 2: Results for *Fluo-N2DL-HeLa* data-set. Average precision (AP) is shown in legend of (a) and (b). Average recall (AR) is shown in legend of (c). Results in a) were computed using ground truth (GT) cell markers from all frames, while the results in b) and c) were computed using GT segmentation masks from four frames.

precision drops for the last $\sim$10% of cells, some of which can be very challenging.

Fig. 2b shows the precision-recall curves when GT cell masks are used to evaluate methods. The difference in precision and recall of *CPN* and other methods is larger in this case. One reason for this larger difference is the fact that two of the four frames have high cell density and contain many conjoined and overlapping cells: challenging situations which *CPN* can handle better. This trend is also observed in evaluation of individual frames, where *CPN* has greater lead over other methods for dense frames compared with sparse frames. Another reason for this difference is the lower recall of *BLOB* and *MSER* due to removal of some good proposals in non-maxima suppression stage. Higher recall (but still lower than *CPN*) can be obtained for *MSER* and *BLOB* by removing non-maxima suppression at the cost of drop in their precision.

Fig. 2c shows how the recall varies with segmentation IoU threshold, average recall (AR) values are shown in the legend. Top 900 proposals were selected from all methods and recall computed at IoU thresholds between 0.1 and 1. *CPN* has higher recall for IoU thresholds up to 0.7 but its recall drops below combination of *BLOB* and set *A* features for higher thresholds. *CPN* has lower recall at high IoUs due to limited precision of predicted bounding boxes and use of thresholding when obtaining segmentation masks. Localization of predicted bounding boxes can potentially be improved by using feature maps from earlier layers [15] and more accurate segmentation masks can be obtained by replacing thresholding with graph cuts.

We tested 3 very different feature sets for ranking proposals. *Set A*, which contained multiple histograms, consistently produced better ranking for both *BLOB* and *MSER* proposals indicating that using more advanced shape and appearance features has some advantage over more commonly used basic appearance and shape features. The difference between performance of *Set B* and *C*, both of which contained simple intensity and shape statistics, was not as consistent.

We also include the cell detection results for a state of the art method [2, 16] (*CellDetect*) and our method (*CPN-Greedy*) in Fig. 2a and Fig. 2b. *CellDetect* uses MSER to generate cell proposals and structured SVM to learn the probability of each proposal being a cell from GT cell markers. It then uses dynamic programming to select the optimal set of proposals, which are the detected cells. *CPN-Greedy* greedily picks the top ranked proposals under the constraints that selected proposals do not conflict and have a high score. *CPN-Greedy* has higher recall and precision than *CellDetect* (Fig. 2a and Fig. 2b). Both *CPN* and *CellDetect* have almost same performance for low cell density frames but for high cell density frames *CPN-Greedy* has higher recall and precision, indicating that it can better handle challenging situations. Further improvement in recall and precision can be obtained by using Integer Linear Programming for selecting the optimal set of proposals.

4. CONCLUSIONS

In this paper we have proposed a convolutional neural network based cell proposal generation method, which generates a set of segmentation masks along with their probabilities. We have shown that our proposal generation method performs better on a challenging data-set compared with current cell proposal generation methods and a state of the art cell detection method. We have demonstrated that it can generate much better proposal ranking than hand crafted sets of shape and appearance features commonly used for this task. We plan to extend this work by utilizing our proposal generation method in a joint cell detection and tracking method. We also plan to apply this method to images from other microscopy modalities. Some initial tests on phase contrast images of non-labeled cells have provided promising results. Code is available at `https://github.com/SaadUllahAkram/CellProposalNetwork`.

5. REFERENCES

[1] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What makes for effective detection proposals?," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2015. 1

[2] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Learning to Detect Cells Using Non-overlapping Extremal Regions," in *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*. 2012. 1, 2, 3, 4

[3] R. Bise and Y. Sato, "Cell Detection From Redundant Candidate Regions Under Nonoverlapping Constraints," *IEEE Transactions on Medical Imaging*, 2015. 1, 2

[4] J. Funke, F. A. Hamprecht, and C. Zhang, "Learning to Segment: Training Hierarchical Segmentation under a Topological Loss," in *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*. 2015. 1, 2

[5] E. Türetken, X. Wang, C. Becker, C. Haubold, and P. Fua, "Globally Optimal Cell Tracking using Integer Programming," *arXiv:1501.05499*, 2015. 1, 2

[6] M. Schiegg, P. Hanslovsky, C. Haubold, U. Koethe, L. Hufnagel, and F. A. Hamprecht, "Graphical Model for Joint Segmentation and Tracking of Multiple Dividing Cells," *Bioinformatics*, 2015. 1, 2, 3

[7] S. U. Akram, J. Kannala, L. Eklund, and J. Heikkilä, "Joint Cell Segmentation And Tracking Using Cell Proposals," in *IEEE International Symposium on Biomedical Imaging (ISBI)*, 2016. 1, 2, 3

[8] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Detecting Overlapping Instances in Microscopy Images using Extremal Region Trees," *Medical Image Analysis*, 2015. 1

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015. 2

[10] P. O. Pinheiro, R. Collobert, and P. Dollár, "Learning to Segment Object Candidates," in *Advances in Neural Information Processing Systems (NIPS)*, 2015. 2

[11] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *European Conference on Computer Vision (ECCV)*, 2014. 2

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[13] M. Maška, V. Ulman, D. Svoboda, et al., "A Benchmark for Comparison of Cell Tracking Algorithms," *Bioinformatics*, 2014. 3

[14] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust Wide-Baseline Stereo from Maximally Stable Extremal Regions," *Image and Vision Computing*, 2004. 3

[15] A. Ghodrati, A. Diba, M. Pedersoli, T. Tuytelaars, and L. V. Gool, "DeepProposal: Hunting Objects by Cascading Deep Convolutional Layers," in *IEEE International Conference on Computer Vision (ICCV)*, 2015. 4

[16] "Learning to Detect Cells Software," `http://www.robots.ox.ac.uk/~vgg/software/cell_detection/`, (accessed January 22, 2016). 4