Cell Detection with Star-convex Polygons

Uwe Schmidt[1,⋆], Martin Weigert[1,⋆], Coleman Broaddus[1], and Gene Myers[1,2]

[1] Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, Germany
Center for Systems Biology Dresden, Germany
[2] Faculty of Computer Science, Technical University Dresden, Germany

Abstract. Automatic detection and segmentation of cells and nuclei in microscopy images is important for many biological applications. Recent successful learning-based approaches include per-pixel cell segmentation with subsequent pixel grouping, or localization of bounding boxes with subsequent shape refinement. In situations of crowded cells, these can be prone to segmentation errors, such as falsely merging bordering cells or suppressing valid cell instances due to the poor approximation with bounding boxes. To overcome these issues, we propose to localize cell nuclei via *star-convex polygons*, which are a much better shape representation as compared to bounding boxes and thus do not need shape refinement. To that end, we train a convolutional neural network that predicts for every pixel a polygon for the cell instance at that position. We demonstrate the merits of our approach on two synthetic datasets and one challenging dataset of diverse fluorescence microscopy images.

1 Introduction

Many biological tasks rely on the accurate detection and segmentation of cells and nuclei from microscopy images [11]. Examples include high-content screens of variations in cell phenotypes [2], or the identification of developmental lineages of dividing cells [1,17]. In many cases, the goal is to obtain an *instance segmentation*, which is the assignment of a cell instance identity to every pixel of the image. To that end, a prevalent *bottom-up* approach is to first classify every pixel into semantic classes (such as *cell* or *background*) and then group pixels of the same class into individual instances. The first step is typically done with learned classifiers, such as random forests [16] or neural networks [15,4,5]. Pixel grouping can for example be done by finding connected components [4]. While this approach often gives good results, it is problematic for images of very crowded cell nuclei, since only a few mis-classified pixels can cause bordering but distinct cell instances to be fused [3,19].

An alternative *top-down* approach is to first localize individual cell instances with a rough shape representation and then *refine* the shape in an additional step. To that end, state-of-the-art object detection methods [9,12,14] predominately predict axis-aligned bounding boxes, which can be refined to obtain an

⋆ Equal contribution.

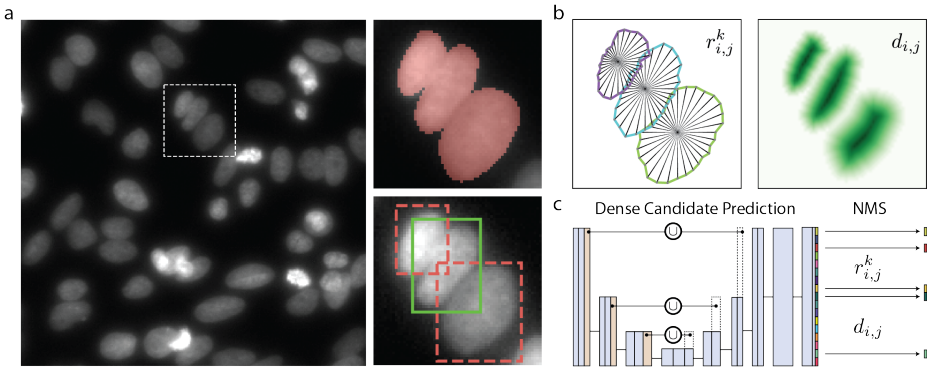

Fig. 1: *(a)* Potential segmentation errors for images with crowded nuclei: Merging of touching cells (upper right) or suppression of valid cell instances due to large overlap of bounding box localization (lower right). *(b)* The proposed StarDist method predicts object probabilities $d_{i,j}$ and star-convex polygons parameterized by the radial distances $r^k_{i,j}$. *(c)* We densely predict $r^k_{i,j}$ and $d_{i,j}$ using a simple U-Net architecture [15] and then select the final instances via non-maximum suppression (NMS).

instance segmentation by classifying the pixels within each box (*e.g.*, *Mask R-CNN* [6]). Most of these methods have in common that they avoid detecting the same object multiple times by performing a *non-maximum suppression* (NMS) step where boxes with lower confidence are suppressed by boxes with higher confidence if they substantially overlap. NMS can be problematic if the objects of interest are poorly represented by their axis-aligned bounding boxes, which can be the case for cell nuclei (Fig. 1a). While this can be mitigated by using *rotated* bounding boxes [10], it is still necessary to refine the box shape to accurately describe objects such as cell nuclei.

To alleviate the aforementioned problems, we propose StarDist, a cell detection method that predicts a shape representation which is flexible enough such that – without refinement – the accuracy of the localization can compete with that of instance segmentation methods. To that end, we use *star-convex polygons* that we find well-suited to approximate the typically roundish shapes of cell nuclei in microscopy images. While Jetley et al. [7] already investigated star-convex polygons for object detection in natural images, they found them to be inferior to more suitable shape representations for typical object classes in natural images, like people or bicycles.

In our experimental evaluation, we first show that methods based on axis-aligned bounding boxes (we choose Mask R-CNN as a popular example) cannot cope with certain shapes. Secondly, we demonstrate that our method performs well on images with very crowded nuclei and does not suffer from merging bordering cell instances. Finally, we show that our method exceeds the performance of strong competing methods on a challenging dataset of fluorescence microscopy images. StarDist uses a light-weight neural network based on *U-Net* [15] and is easy to train and use, yet is competitive with state-of-art methods.

2 Method

Our approach is similar to object detection methods [12,9,7] that directly predict shapes for each object of interest. Unlike most of them, we do not use axis-aligned bounding boxes as the shape representation ([7,10] being notable exceptions). Instead, our model predicts a *star-convex polygon* for every pixel[3]. Specifically, for each pixel with index i, j we regress the distances $\{r_{i,j}^k\}_{k=1}^n$ to the boundary of the object to which the pixel belongs, along a set of n predefined radial directions with equidistant angles (Fig. 1b). Obviously, this is only well-defined for (non-background) pixels that are contained within an object. Hence, our model also separately predicts for every pixel whether it is part of an object, so that we only consider polygon proposals from pixels with sufficiently high object probability $d_{i,j}$. Given such polygon candidates with their associated object probabilities, we perform non-maximum suppression (NMS) to arrive at the final set of polygons, each representing an individual object instance.

Object probabilities. While we could simply classify each pixel as either object or background based on binary masks, we instead define its object probability $d_{i,j}$ as the (normalized) Euclidean distance to the nearest background pixel (Fig. 1b). By doing this, NMS will favor polygons associated to pixels near the cell center (*cf.* Fig. 5b), which typically represent objects more accurately.

Star-convex polygon distances. For every pixel belonging to an object, the Euclidean distances $r_{i,j}^k$ to the object boundary can be computed by simply following each radial direction k until a pixel with a different object identity is encountered. We use a simple GPU implementation that is fast enough that we can compute the required distances on demand during model training.

2.1 Implementation

Although our general approach is not tied to a particular regression or classification approach, we choose the popular U-Net [15] network as the basis of our model. After the final U-Net feature layer, we cautiously add an additional 3×3 convolutional layer with 128 channels (and *relu* activations) to avoid that the subsequent two output layers have to "fight over features". Specifically, we use a single-channel convolutional layer with *sigmoid* activation for the object probability output. The polygon distance output layer has as many channels as there are radial directions n and does not use an additional activation function.

Training. We minimize a standard *binary cross-entropy* loss for the predicted object probabilities. For the polygon distances, we use a *mean absolute error* loss weighted by the ground truth object probabilities, *i.e.* the pixel-wise errors are multiplied by the object probabilities before averaging. Consequently, background pixels will not contribute to the loss, since their object probability is zero. Furthermore, predictions for pixels closer to the center of each object

[3] Although we only consider the single object class *cell nuclei* in our experiments, note that we are not limited to that and thus use the generic term *object* in the following.

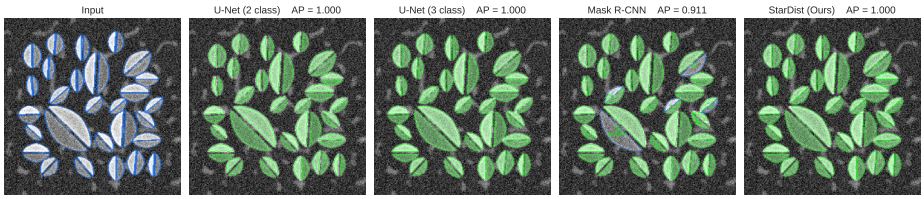

Fig. 2: Segmentation result ($\tau = 0.5$) for TOY image. Predicted cell instances are depicted in green if correctly matched (TP), otherwise highlighted in red (FP). Ground truth cells are always shown by their blue outlines in the input image (left), and in all other images only when they are not matched by any predicted cell instance (FN).

are weighted more, which is appropriate since these will be favored during non-maximum suppression. The code is publicly available[4].

Non-maximum suppression. We perform common, greedy non-maximum suppression (NMS, *cf.* [14,9,12]) to only retain those polygons in a certain region with the highest object probabilities. We only consider polygons associated with pixels above an object probability threshold as candidates, and compute their intersections with a standard polygon clipping method.

3 Experiments

3.1 Datasets

We use three datasets that pose different challenges for cell detection:

Dataset TOY*:* Synthetically created images that contain pairs of touching half-ellipses with blur and background noise (*cf.* Fig. 2). Each pair is oriented in such a way that the overlap of both enclosing bounding boxes is either very small (along an axis-aligned direction) or very large (when the ellipses touch at an oblique angle). This dataset contains 1000 images of size 256×256 with associated ground truth labels. We specifically created this dataset to highlight the limitations of methods that predict axis-aligned bounding boxes.

Dataset TRAGEN*:* Synthetically generated images of an evolving cell population from [18] (*cf.* Fig. 3). The generative model includes cell divisions, shape deformations, camera noise and microscope blur and is able to simulate realistic images of extremely crowded cell configurations. This dataset contains 200 images of size 792×792 along with their ground truth labels.

Dataset DSB2018*:* Manually annotated real microscopy images of cell nuclei from the 2018 Data Science Bowl[5]. From the original dataset (670 images from diverse modalities) we selected a subset of fluorescence microscopy images and removed images with labeling errors, yielding a total of 497 images (*cf.* Fig. 4).

For each dataset, we use 90% of the images for training and 10% for testing. We train all methods (Section 3.3) with the same random crops of size 256×256 from the training images (augmented via axis-aligned rotations and flips).

[4] `https://github.com/mpicbg-csbd/stardist`
[5] `https://www.kaggle.com/c/data-science-bowl-2018`

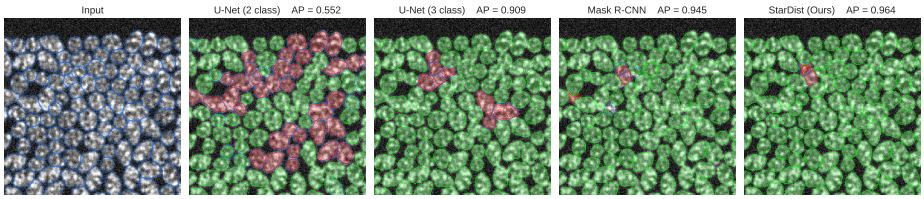

Fig. 3: Segmentation result ($\tau = 0.5$) for TRAGEN image. See Fig. 2 caption for legend.

3.2 Evaluation Metric

We adopt a typical metric for object detection: A detected object I_{pred} is considered a match (*true positive* TP_τ) if a ground truth object I_{gt} exists whose *intersection over union* $IoU = \frac{I_{\text{pred}} \cap I_{\text{gt}}}{I_{\text{pred}} \cup I_{\text{gt}}}$ is greater than a given threshold $\tau \in [0, 1]$. Unmatched predicted objects are counted as *false positives* (FP_τ), unmatched ground truth objects as *false negatives* (FN_τ). We use the *average precision* $AP_\tau = \frac{TP_\tau}{TP_\tau + FN_\tau + FP_\tau}$ evaluated across all images as the final score.

3.3 Compared Methods

U-Net (2 class): We use the popular U-Net architecture [15] as a baseline to predict 2 output classes (cell, background). We use 3 down/up-sampling blocks, each consisting of 2 convolutional layers with $32 \cdot 2^k (k = 0, 1, 2)$ filters of size 3×3 (approx. 1.4 million parameters in total). We apply a threshold σ on the cell probability map and retain the connected components as final result (σ is optimized on the validation set for every dataset).

U-Net (3 class): Like U-Net (2 class), but we additionally predict the *boundary pixels* of cells as an extra class. The purpose of this is to differentiate crowded cells with touching borders (similar to [4,5]). We again use the connected components of the thresholded cell class as final result.

Mask R-CNN: A state-of-the-art instance segmentation method combining a bounding-box based region proposal network, non-maximum-suppression (NMS), and a final mask segmentation (approx. 45 million parameters in total). We use a popular open-source implementation[6]. For each dataset, we perform a grid-search over common hyper-parameters, such as detection NMS threshold, region proposal NMS threshold, and number of anchors.

STARDIST*:* Our proposed method as described in Section 2. We always use $n = 32$ radial directions (*cf.* Fig. 1b) and employ the same U-Net backbone as for the first two baselines described above.

3.4 Results

We first test our approach on dataset TOY, which was intentionally designed to contain objects with many overlapping bounding boxes. The results in Table 1

[6] `https://github.com/matterport/Mask_RCNN`

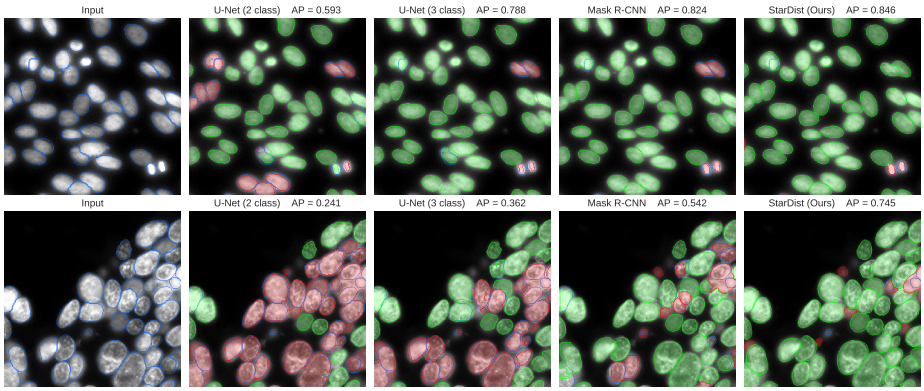

Fig. 4: Two segmentation results ($\tau = 0.5$) for DSB2018. See Fig. 2 caption for legend.

and Fig. 2 show that for moderate IoU thresholds ($\tau < 0.7$), StarDist and both U-Net baselines yield essentially perfect results. Mask R-CNN performs substantially worse due to the presence of many slanted and touching pairs of objects (which have almost identical bounding boxes, hence one is suppressed). This experiment highlights a fundamental limitation of object detection methods that predict axis-aligned bounding boxes.

On dataset TRAgen, U-Net (2 class) shows the lowest accuracy mainly due to the abundance of touching cells which are erroneously fused. Table 1 shows that all other methods attain almost perfect accuracy for many IoU thresholds even on very crowded images, which might be due to the stereotypical size and texture of the simulated cells. We show the most difficult test image in Fig. 3.

Finally, we turn to the real dataset DSB2018 where we find StarDist to outperform all other methods for IoU thresholds $\tau < 0.75$, followed by the next best method Mask R-CNN (*cf.* Table 1 and Fig. 5a). Fig. 4 shows the results and errors for two different types of cells. Common segmentation errors include merged cells (mostly for the 2 class U-Net), bounding box artifacts (Mask R-CNN) and missing cells (all methods). The bottom example of Fig. 4 is particularly challenging, where out-of-focus signal results in densely packed and partially overlapping

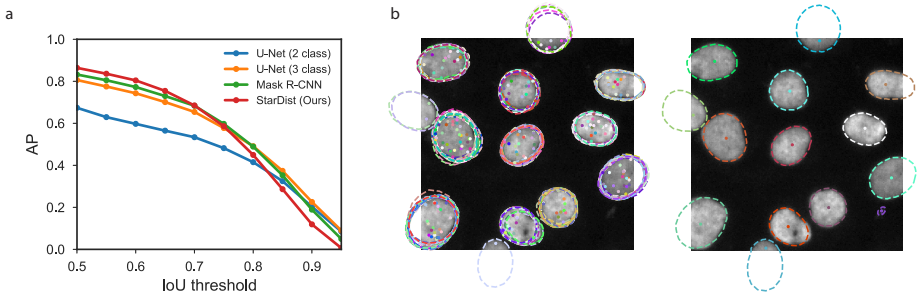

Fig. 5: *(a)* Detection scores on dataset DSB2018 (*cf.* Table 1, bottom). *(b)* Example of StarDist polygon predictions for 200 random pixels (left) and for all pixels after non-maximum suppression (right); pixels and associated polygons are color-matched.

Threshold τ	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
					Toy				
U-Net (2 class)	0.9994	0.9990	0.9977	0.9931	0.9641	0.8659	0.6229	0.2939	0.0667
U-Net (3 class)	**0.9998**	**0.9998**	**0.9998**	**0.9998**	**0.9998**	**0.9998**	**0.9990**	**0.9874**	**0.9243**
Mask R-CNN	0.9104	0.9061	0.9014	0.8944	0.8729	0.8471	0.7728	0.6075	0.3717
StarDist (Ours)	**0.9998**	**0.9998**	**0.9998**	**0.9998**	0.9994	0.9890	0.8695	0.4630	0.0748
					TRAgen				
U-Net (2 class)	0.9030	0.8908	0.8852	0.8815	0.8811	0.8783	0.8566	0.6937	0.4056
U-Net (3 class)	0.9918	0.9904	0.9899	0.9897	0.9890	0.9883	**0.9848**	**0.9679**	**0.8995**
Mask R-CNN	0.9924	0.9919	0.9912	0.9898	0.9863	0.9777	0.9594	0.8948	0.5280
StarDist (Ours)	**0.9984**	**0.9981**	**0.9976**	**0.9967**	**0.9953**	**0.9934**	0.9841	0.9465	0.4259
					DSB2018				
U-Net (2 class)	0.6739	0.6295	0.5975	0.5650	0.5339	0.4819	0.4151	0.3248	0.2032
U-Net (3 class)	0.8060	0.7753	0.7431	0.7011	0.6543	0.5777	**0.4910**	**0.3738**	**0.2258**
Mask R-CNN	0.8323	0.8051	0.7728	0.7299	0.6838	**0.5974**	0.4893	0.3525	0.1891
StarDist (Ours)	**0.8641**	**0.8361**	**0.8043**	**0.7545**	**0.6850**	0.5862	0.4495	0.2865	0.1191

Table 1: Cell detection results for three datasets and four methods, showing *average precision* (AP) for several *intersection over union* (IoU) thresholds τ.

cell shapes. Here, merging mistakes are pronounced for both U-Net baselines. All false positives predicted by StarDist retain a reasonable shape, whereas those predicted by Mask R-CNN sometimes exhibit obvious artifacts.

We observe that StarDist yields inferior results for the largest IoU thresholds τ for our synthetic datasets. This is not surprising, since we predict a parametric shape model based on only 32 radial directions, instead of a per-pixel segmentation as all other methods. However, an advantage of a parametric shape model is that it can be used to predict reasonable complete shape hypotheses from nuclei that are only partially visible at the image boundary (*cf.* Fig. 5b, also see [20]).

4 Discussion

We demonstrated that star-convex polygons are a good shape representation to accurately localize cell nuclei even under challenging conditions. Our approach is especially appealing for images of very crowded cells. When our StarDist model makes a mistake, it does so gracefully by either simply omitting a cell or by predicting at least a plausible cell shape. The same cannot by said for the methods that we compared to, whose predicted shapes are sometimes obviously implausible (*e.g.*, containing holes or ridges). While StarDist is competitive to the state-of-the-art Mask R-CNN method, a key advantage is that it has an order of magnitude fewer parameters and is much simpler to train and use. In contrast to Mask R-CNN, StarDist has only few hyper-parameters that do not need careful tuning to achieve good results.

Our approach could be particularly beneficial in the context of cell tracking. There, it is often desirable to have multiple diverse segmentation hypotheses [13,8], which could be achieved by suppressing fewer candidate polygons. Furthermore, StarDist can plausibly complete shapes for partially visible cells at the image boundary, which could make it easier to track cells that enter and leave the field of view over time.

References

1. Amat, F., Lemon, W., Mossing, D.P., McDole, K., Wan, Y., Branson, K., Myers, E.W., Keller, P.J.: Fast, accurate reconstruction of cell lineages from large-scale fluorescence microscopy data. Nature methods **11**(9), 951 (2014)
2. Boutros, M., Heigwer, F., Laufer, C.: Microscopy-based high-content screening. Cell **163**(6), 1314–1325 (2015)
3. Caicedo, J.C., Roth, J., Goodman, A., Becker, T., Karhohs, K.W., McQuin, C., Singh, S., Theis, F., Carpenter, A.E.: Evaluation of deep learning strategies for nucleus segmentation in fluorescence images. bioRxiv (2018)
4. Chen, H., Qi, X., Yu, L., Heng, P.A.: DCAN: Deep contour-aware networks for accurate gland segmentation. In: CVPR (2016)
5. Guerrero-Pena, F.A., Marrero Fernandez, P.D., Ren, T.I., Yui, M., Rothenberg, E., Cunha, A.: Multiclass weighted loss for instance segmentation of cluttered cells. arXiv (2018)
6. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: ICCV (2017)
7. Jetley, S., Sapienza, M., Golodetz, S., Torr, P.H.: Straight to shapes: Real-time detection of encoded shapes. In: CVPR (2017)
8. Jug, F., Levinkov, E., Blasse, C., Myers, E.W., Andres, B.: Moral lineage tracing. In: CVPR (2016)
9. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single shot multibox detector. In: ECCV (2016)
10. Ma, J., Shao, W., Ye, H., Wang, L., Wang, H., Zheng, Y., Xue, X.: Arbitrary-oriented scene text detection via rotation proposals. IEEE Transactions on Multimedia (2018)
11. Meijering, E.: Cell segmentation: 50 years down the road. IEEE Signal Processing Magazine **29**(5), 140–145 (2012)
12. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR (2016)
13. Rempfler, M., Kumar, S., Stierle, V., Paulitschke, P., Andres, B., Menze, B.H.: Cell lineage tracing in lens-free microscopy videos. In: MICCAI (2017)
14. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS (2015)
15. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015)
16. Sommer, C., Straehle, C., Koethe, U., Hamprecht, F.A.: Ilastik: Interactive learning and segmentation toolkit. In: Int. Symposium on Biomedical Imaging (2011)
17. Ulman, V., Maška, M., Magnusson, K.E., Ronneberger, O., Haubold, C., Harder, N., Matula, P., Matula, P., Svoboda, D., Radojevic, M., et al.: An objective comparison of cell-tracking algorithms. Nature methods **14**(12), 1141 (2017)
18. Ulman, V., Orémuš, Z., Svoboda, D.: TRAgen: a tool for generation of synthetic time-lapse image sequences of living cells. In: ICIAP (2015)
19. Xie, W., Noble, J.A., Zisserman, A.: Microscopy cell counting and detection with fully convolutional regression networks. Computer methods in biomechanics and biomedical engineering: Imaging & Visualization **6**(3), 283–292 (2018)
20. Yurchenko, V., Lempitsky, V.: Parsing images of overlapping organisms with deep singling-out networks. In: CVPR (2017)