Segmentation of cell nuclei in 2D microscopy images with CNNs

Sander de Haan, Aindrias Lefévère-Laoide, Carlo Refice

Introduction

The cell is the fundamental biological unit of life. Research in cellular biology and biomedical engineering, together with a wide range of other biological tasks, depend on the segmentation of cells, or parts of the cell, from microscopy images. A common segmentation objective is to separate the nuclei from the background in cell cultures. Often, identifying each nucleus individually, or *instance segmentation*, is the consequent target. This work will focus on nucleus instance segmentation from a fluorescent microscopy data set, consisting of 2-D images. The goal of this work is exemplified in Figure 1a.

In recent years, a growing trend towards employing more complex methods from Machine Learning has been observed in the literature [5]. Most of the newly proposed approaches find themselves in deep learning, especially convolutional neural networks (CNNs) based architectures such as U-Net [7] and DeepCell[10]. This paper builds on the former example to engage in the instance segmentation of nuclei.

I. Dataset

A. Description of the dataset

The dataset consists of florescent microscopy images of different cell lines from neuroblastoma. The dataset was designed and labeled for machine learning applications[4], and consists of 79 images of various sizes which were selected for their heterogeneity. Notably, images in this dataset include tightly aggregated nuclei which makes the segmentation task more challenging.

The dataset also provides hand-segmented ground truth labels, in which each pixel is mapped to the unique 16-bit identifier of the cell it belongs to, or to zero if the pixel belongs to the background.

B. Data pre-processing

The dataset presented some irregularities that we sought to remove:

- The provided images were not all of the same size, with some images being significantly smaller than the rest. As suggested in the original U-Net paper [7], this problem can be alleviated by padding images smaller than a size of 1024×1024 through mirrored tiling of the original image. Tiling has the advantage of not introducing frequencies not present in the original image.
- Images generated through microscopy have different brightness levels depending on ambient conditions. This creates undesirable variance in the dataset. Therefore, min-max normalization was applied, mapping the range of values in each image to the full range [0, 1] for each pixel.
- Target class labels have to be computed from the ground truth instance segmentation. This was easily performed through simple binary thresholding for binary segmentation, and through edge detection to find the boundaries for each cell in the three-class case.
- The size of the dataset only amounts to 79 images. Accordingly, the dataset has been augmented with flipped and rotated copies of each image, which also helped the model be invariant to these kinds of transformations.

II. Methods

A. Model

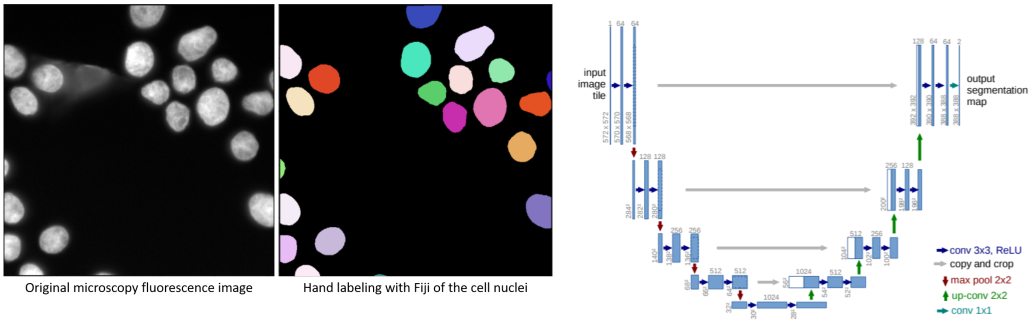
The U-Net architecture consists of a contracting path with five convolutional layers followed by a symmetric expanding path which combines the spatial information with the features obtained from the contraction. The two paths together form a *U*-shape as depicted in Figure 1b. The input of the first layer consists of the image to be processed which is padded by mirror-tiling on the edges. This is done such that all the information of the image is conserved throughout the shrinking convolution steps. The U-Net is interesting it requires very little training data for decent performance.

B. Application

The model is applied as follows. First, two-class segmentation, namely with background and nuclei classes, is performed. Second, three-class segmentation, with an additional boundary class, is executed. After the two- and three-class segmentation, the nuclei instances are labeled using the Watershed transform [6]. The Watershed transform is preferred over other connectivity-based algorithms [2], as these tend to not cope well with overlapping nuclei, resulting in split or merged objects [1]. Implementation wise, the distance matrix is found by computing the Euclidean distance between each cell pixel towards the closest background or boundary pixel. Further, the cell segmentation class is used as a heuristic for labeling; only the pixels that are marked as a cell are labeled.

More specifically, the U-Net model, data loading, and preprocessing are implemented using the *skimage* and *Pytorch* libraries, which provide many utilities for data transformations on images, defining model architectures, training, and other tasks.

Experiments with different optimizers were performed while training our model. First, a simple Stochastic Gradient Descent



(a) The goal of the task is to be able to segment all the cell nuclei in a fluorescence microscopy photo.

(b) Architecture of a U-Net.

Fig. 1: Aim of the task and structure of the model used.

optimization was performed, however, the learning rate we used produced unstable results. Further, the Adam [3] optimizer was compared against AdaDelta [11], and concluded that Adam is the preferred choice as the validation loss is significantly lower, i.e. 0.1 for Adam compared to 0.7 for AdaDelta.

Additionally, different loss functions were tested. For the binary classification problem, a binary cross-entropy loss preceded by a sigmoid activation layer to convert logits into prediction confidences was used. The two classes were fairly well-balanced in the dataset, which added to the success of this method. As for the three-class problem, an unweighted cross-entropy loss coupled with a softmax activation function was used for the for the training. This did not produce satisfactory results, since the boundary class is extremely unbalanced with respect to the background and nuclei class. Consequently, we introduced weights for each class c, which are computed as follows:

$$w(c) = 1/\sqrt{o(c)},$$

where $o(c)$ is the occupancy of class c in the entire dataset (the ratio of the pixels which are assigned to class c).

Dice loss [9] is another loss function that works well for imbalanced classes, and using it for boundary segmentation tasks has been shown to produce crisp, thin borders. Further experimentation revealed that a combination of Dice and cross-entropy loss, where both are weighed equally, produced the best results. For the final model, this combination, using Dice loss to capture the small details in the image and the borders and cross-entropy loss for the wider details such as the cell interiors, is employed.

The models employ a simple train-validation dataset split, with 80% of the data used for the training. The hyperparameters are tuned by a grid search.

C. Evaluation

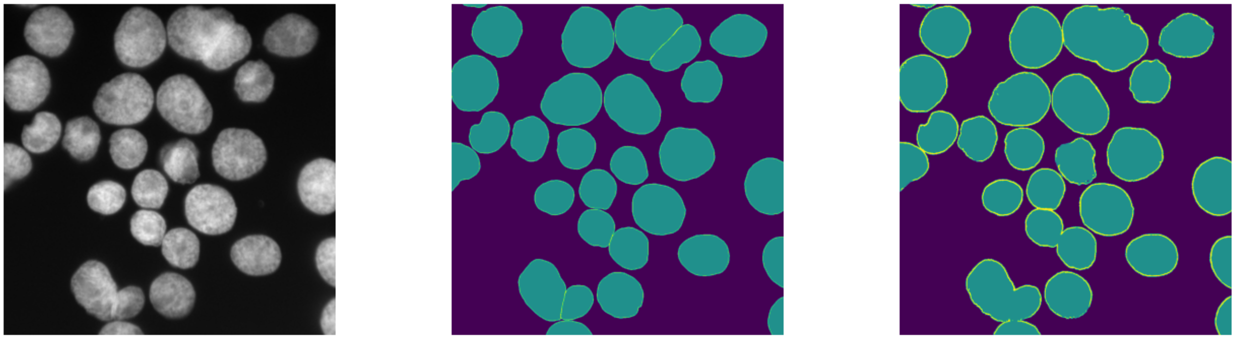
A second model was used for performance measurements, namely the Stardist model [8]. Stardist relies on star-convex polygons to localize cell nuclei, in contrast to the more traditional approach of using axis-aligned bounding boxes. Specifically, for each pixel, the model predicts whether it is part of an object, and regresses the distance along predefined axes to the boundary of the object it belongs to. Non-maximum suppression (NMS) is then performed to obtain one polygon for every object instance.

In order to compare the models, we used a number of evaluation metrics, all based on *Intersection over Union* (IoU), which is the area of overlap between the predicted segmentation and the ground truth, divided by the area of the union. The traditional binary classification metrics, True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), are parameterized by a threshold $\tau \in [0,1]$, such that a predicted entity I_{pred}, is considered a match if its IoU with the ground truth, I_{gt}, is above τ. Applying this, the models were compared using the following metrics:

$$\begin{aligned}
\textit{precision} &= \frac{TP}{TP+FP} \\
\textit{recall} &= \frac{TP}{TP+FN} \\
\textit{accuracy} &= \frac{TP+TN}{TP+TN+FP+FN} \\
f_1 &= 2 \cdot \frac{\textit{precision} \cdot \textit{recall}}{\textit{precision} + \textit{recall}} \\
\textit{panoptic quality} &= \frac{\sum_{(I_{pred},I_{gt})\in TP} IoU(I_{pred}, I_{gt})}{|TP|} \\
\textit{mean matched score} &= \frac{1}{|TP|} \sum_{(I_{pred},I_{gt})\in TP} IoU(I_{pred}, I_{gt}) \\
\textit{mean true score} &= \frac{1}{|GT|} \sum_{(I_{pred},I_{gt})\in TP} IoU(I_{pred}, I_{gt}),
\end{aligned}$$

where $|GT|$ is the total number of ground-truth entities in the image.

To evaluate StarDist, we used the model structure and package provided by the original authors of the paper [8]. We trained it on our data and compared its performance on the validation set with the U-Net with watershed and connected components segmentation. We trained both our U-Net model



(a) The input image from the dataset. (b) The ground truth class labels of the image. (c) The class labels predicted by the model.

Fig. 2: Results of the U-Net training for the three-class problem. Notice that the borders are much thicker than in the class label, and that the model by itself is unable to distinguish touching cells.

and the Stardist model using the same train/validation splits, and evaluated it using the same performance metrics.

III. Results

As can be seen in Figure 2, the U-Net performs well in detecting the inside and borders of cells, as long as they are not touching. The model instead does not seem to recognize the borders of touching cells, instead treating them all as one cell interior.

The Watershed algorithm we introduced was supposed to combat this problem. However, looking at the performance graphs in Figure 3, we can see that Watershed instance segmentation performs much worse than a simple connected-components segmentation: it generates close to one thousand false negatives, even for low thresholds. The only metric which is comparable to the other two segmentation methods is precision. Overall, performance does not seem reliable with this method.

Figure 4 gives us some insight into why that is the case: since we used the same distance threshold for the entire dataset, large enough cells are sometimes incorrectly subdivided, whereas ones that are too small are not.

Using different thresholds for different images could have produced better results, but this exemplifies the underlying weakness of the Watershed transform, namely that it requires prior knowledge of object sizes.

Concerning the U-Net using the connected components transform, the number of false negatives is lower than the Watershed with around 200 for low threshold values. However, the number of false positives is higher, which means that non-existing nuclei are detected. Again by looking at the results in Figure 2, the connected component model seems to be dealing with large and sparse nuclei better than the Watershed transform, although struggling with segmenting the groups of nuclei. This method allows a precision of around .75 and an accuracy of .65 for the first threshold, outperforming the Watershed model.

Finally, both the performance graph and the images show that the Stardist model performs this task in an excellent manner, providing precise segmentation of the nuclei no matter their size or density. With a low threshold for IoU, the model reaches a precision as high as 0.95.

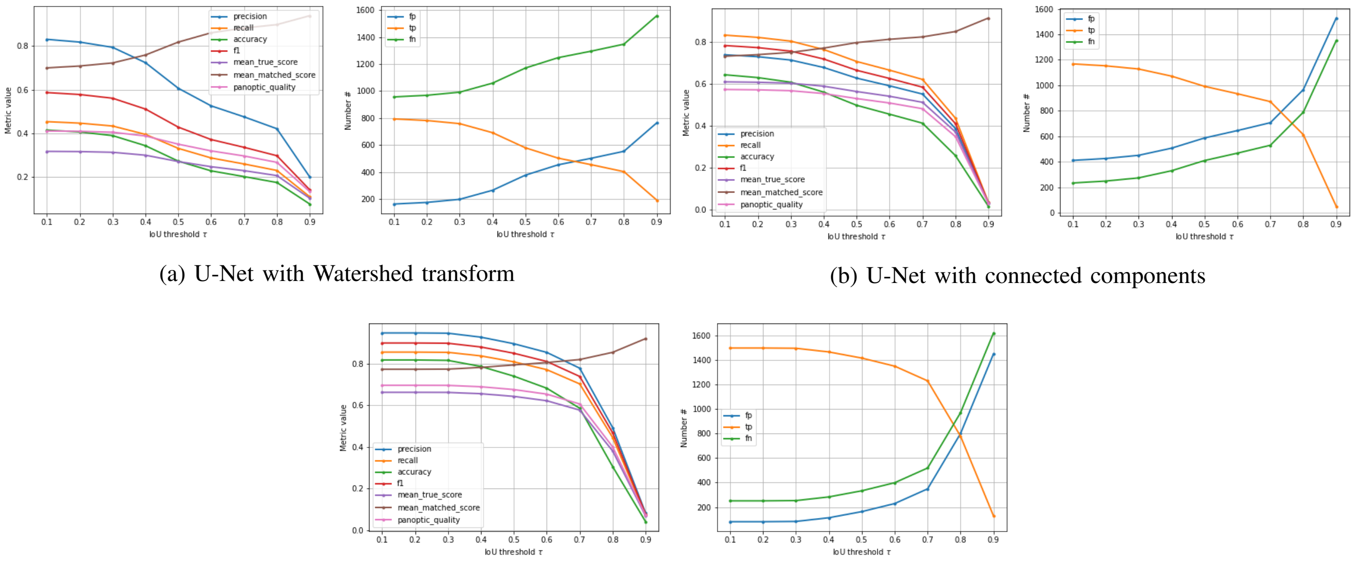
IV. Discussion

Our initial two class (nuclei vs background) classification yielded surprising results: some patches of pixels inside nuclei which were darker (i.e. less fluorescence) in the input data were predicted as background. Adding a border class and using three class classification allowed to get more solid predictions with fewer ”holes” in the nuclei. The results, both visual and in terms of metrics, show that the Watershed and connected components transform, when matched with the U-Net, do not allow reliable cell instance segmentation. The Stardist model, however, performs very well on our dataset, whether on small clusters of nuclei or on large isolated nuclei. These results are not very surprising: we expected the Stardist, which is a reference for instance segmentation for cell nuclei, to perform better than our model.

There are, however, some potential improvements that we thought of yet did not manage to implement:

- A higher amount of convolutional layers, or a different upward path (such as using max unpooling rather than transposed convolution), could be used to combat the ”holes” in the segmented nuclei, by giving the network less localized information to work with.
- To force the network to learn separation boundaries between touching nuclei, pixels belonging to cell boundaries in the ground truth labels could be given a higher relative weight the more ”narrow” the boundary is.

These improvements could hopefully help bring the performance of the U-Net more in line with that of Stardist.



(a) U-Net with Watershed transform

(b) U-Net with connected components

(c) Stardist model

Fig. 3: Performance metrics for the different models. Left shows a range of metric values based on threshold tau and left shows number of true positive, false positive and false negative

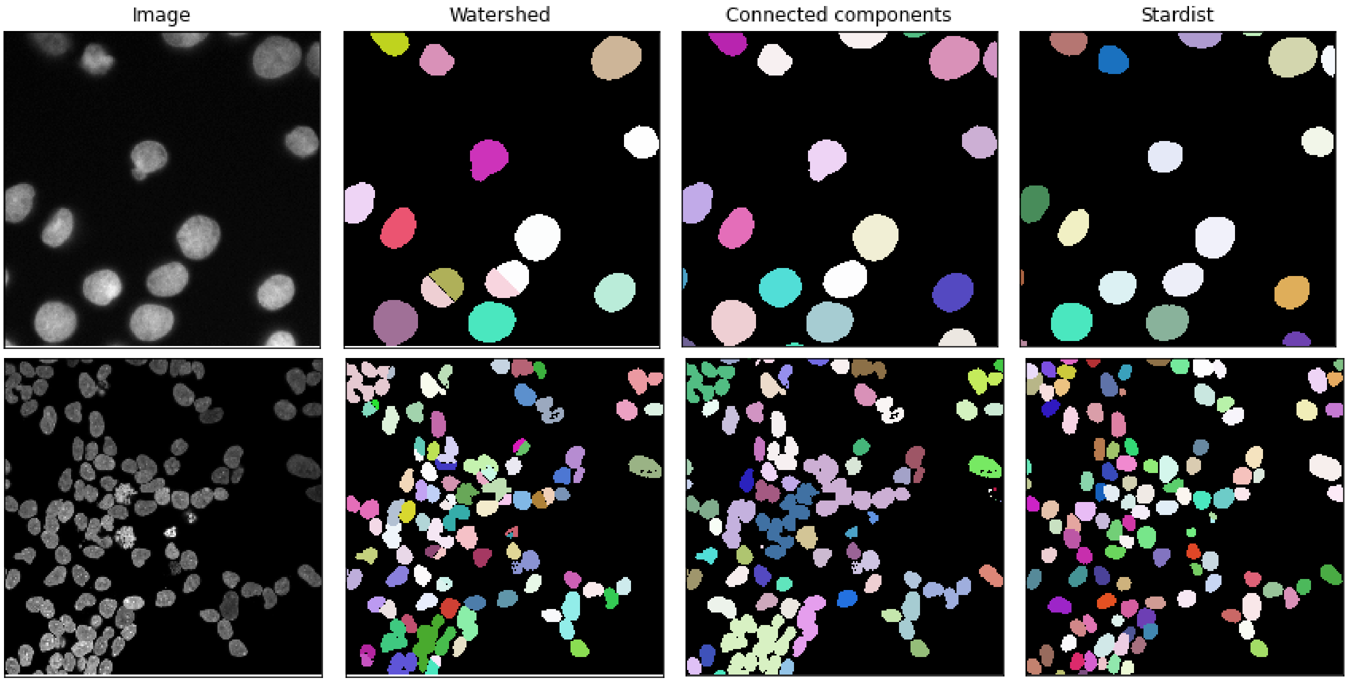


Fig. 4: Visual comparison of performance of U-Net with Watershed, with connected components and of stardist model on two different validation set images

References

[1] Juan C. Caicedo et al. “Evaluation of Deep Learning Strategies for Nucleus Segmentation in Fluorescence Images”. In: *Cytometry Part A* 95.9 (2019), pp. 952–965. DOI: https://doi.org/10.1002/cyto.a.23863.

[2] Christophe Fiorio and Jens Gustedt. “Two linear time Union-Find strategies for image processing”. In: *Theoretical Computer Science* 154.2 (1996), pp. 165–181. ISSN: 0304-3975. DOI: https://doi.org/10.1016/0304-3975(94)00262-2. URL: http://www.sciencedirect.com/science/article/pii/0304397594002622.

[3] Diederik Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *International Conference on Learning Representations* (Dec. 2014).

[4] Florian Kromp et al. “An annotated fluorescence image dataset for training nuclear segmentation methods”. In: *Scientific Data* 7 (Aug. 2020), p. 262. DOI: 10.1038/s41597-020-00608-w.

[5] E. Meijering. “Cell Segmentation: 50 Years Down the Road [Life Sciences]”. In: *IEEE Signal Processing Magazine* 29.5 (2012), pp. 140–145. DOI: 10.1109/MSP.2012.2204190.

[6] Jos Roerdink and A. Meijster. “The Watershed Transform: Definitions, Algorithms and Parallelization Strategies”. In: *Fundam Inf* 41 (Oct. 2003). DOI: 10.3233/FI-2000-411207.

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Springer International Publishing, 2015, pp. 234–241.

[8] Uwe Schmidt et al. “Cell Detection with Star-convex Polygons”. In: (June 2018).

[9] Carole H. Sudre et al. “Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations”. In: *Lecture Notes in Computer Science* (2017), pp. 240–248. ISSN: 1611-3349. DOI: 10.1007/978-3-319-67558-9_28. URL: http://dx.doi.org/10.1007/978-3-319-67558-9_28.

[10] David A. Van Valen et al. “Deep Learning Automates the Quantitative Analysis of Individual Cells in Live-Cell Imaging Experiments”. In: *PLOS Computational Biology* 12.11 (Nov. 2016), pp. 1–24. DOI: 10.1371/journal.pcbi.1005177. URL: https://doi.org/10.1371/journal.pcbi.1005177.

[11] Matthew D. Zeiler. *ADADELTA: An Adaptive Learning Rate Method*. 2012.