DeepSplit: Segmentation of Microscopy Images Using Multi-Task Convolutional Networks

Andrew Torr[1], Doga Basaran[1], Julia Sero[2], Jens Rittscher[1], and Heba Sailem[1][0000-0002-6600-1255]

[1] Department of Engineering Science, University of Oxford, Oxford, United Kingdom, OX1 4BH

[2] Centre for Biosensors, Bioelectronics and Biodevices, University of Bath, Bath BA2 7AY

Abstract. Accurate segmentation of cellular structures is critical for automating the analysis of microscopy data. Advances in deep learning have facilitated extensive improvements in semantic image segmentation. In particular, U-Net, a model specifically developed for biomedical image data, performs multi-instance segmentation through pixel-based classification. However, approaches based on U-Net tend to merge touching cells in dense cell cultures, resulting in under-segmentation. To address this issue, we propose DeepSplit; a multi-task convolutional neural network architecture where one encoding path splits into two decoding branches. DeepSplit first learns segmentation masks, then explicitly learns the more challenging cell-cell contact regions. We test our approach on a challenging dataset of cells that are highly variable in terms of shape and intensity. DeepSplit achieves 90% cell detection coefficient and 90% Dice Similarity Coefficient (DSC) which is a significant improvement on the state-of-the-art U-Net that scored 70% and 84% respectively.

1 Introduction

Cellular imaging is a prevalent tool in biomedical research as it facilitates studying changes in cellular behaviour under different conditions. These include detecting changes in cell shape in cancer cells and characterising cellular response to various genetic and pharmacological treatments [15]. Analysis of these image datasets requires accurate segmentation of various biological entities, such as cells and nuclei. In particular, various measurements of cell shape can be used to infer different cellular states such as cell death, division, motility, and differentiation [12]. High throughput microscopy techniques have dramatically increased the rate at which cellular images can be obtained, making it infeasible for experts to manually segment each image. Therefore, robust automatic segmentation of microscopy images is essential to draw accurate scientific conclusions from the obtained measurements [6].

Our key contribution is to define the problem of multiple instance segmentation of cellular images as a multi-task problem involving 1) semantic segmentation of cell masks, and 2) separation of adjacent cells. This is achieved through an architecture termed DeepSplit. DeepSplit is composed of one encoding branch followed by two decoding branches where each branch optimises for one of these tasks. By combining established semantic segmentation of cell masks with the separation mask of adjacent cells we explicitly tackle the challenge of cell merging. This approach leads to significantly improved overall segmentation results.

Furthermore, our work utilises a cumulative learning approach where these two tasks are trained in two stages. Firstly the segmentation decoding branch is trained while freezing the learning in the separation branch. Once segmentation is learned, the separation decoding branch is trained to classify pixels in cell-cell contact regions. We demonstrate that DeepSplit can successfully segment challenging cellular imaging data where cells vary highly in both their intensity and shape. To our knowledge, this is one of the most effective ways to enforce separation between touching cells.

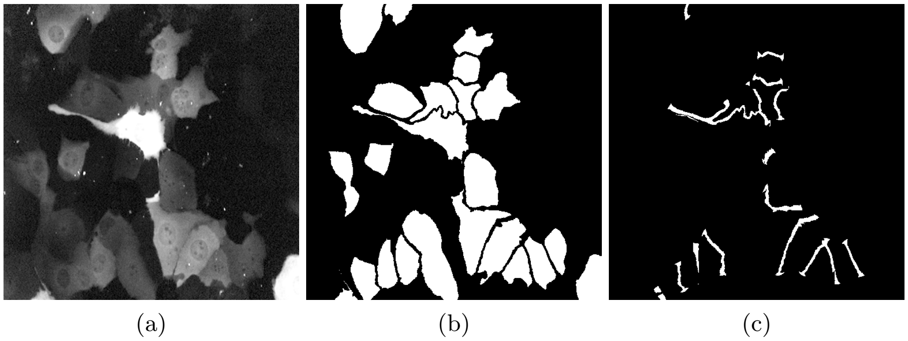


Fig. 1: **Example image of Breast Cancer cells.** a) Raw image. b) Ground truth segmentation mask. c) Ground truth separation mask.

Related Work. Classical image segmentation techniques such as intensity thresholding and watershed segmentation do not perform sufficiently well on images of overlapping and densely packed cells [18]. The main challenge is that the boundaries between touching cells are often indistinct, making it difficult for a non-expert to accurately identify the boundaries (Fig. 1). Machine learning, and particularly deep learning, approaches have proven highly successful in segmentation tasks [1, 8, 9, 19]. These are thoroughly reviewed by Taghanaki *et al.* [17]. Machine learning's effectiveness in this task is owed to its ability to learn features more complex than any classical algorithm could detect, making better use of the information contained in the images. However, these methods require large, labelled data sets for training purposes. Accurate labelling is a challenging and laborious task making it infeasible when diverse cellular imaging datasets are generated on a regular basis.

U-Net is one of the most used architectures for segmenting cellular images as it requires far less training data while achieving accurate segmentation [14]. Additionally, U-Net utilises a weighted loss function which prioritises correct assignment of pixels in the boundaries between touching cells. This is critical to its performance in separating adjacent cells. However, increasing the weights assigned to boundary pixels extensively has been noted to result in inaccurate delineation of cell boundaries [7]. Furthermore, it has been illustrated through the nuclear segmentation challenge that the merging of adjacent nuclei remains a problem even when large amounts of annotations are available (23,165 annotated nuclei) [3]. The task of cellular segmentation, which is the focus of this work, is even more challenging as there is a higher degree of variation in terms of morphology and cell size, and because the boundaries between cells are less well defined.

Inspired by the success of U-Net, a number of extensions have been proposed. For example, V-Net [13] incorporates additional residual connections. Jeugo *et al.* [11] modified DenseNet [10] for segmentation purposes by adapting the U-Net architecture. There have also been many experiments varying the depth, channel count, and number of attention blocks. However, despite impressive DSCs when applied to standard datasets, all of these share the fundamental problem of overfitting to the foreground pixels, resulting in poor boundary detection and merging of adjacent cells. One approach proposed to correct the merging errors is applying a global probabilistic model to refine U-Net segmentation results [7]. This work uses simple size and shape checks to detect possible incorrect merges of nuclei and employs a geometric elliptical model to separate these nuclei. However, this approach assumes a specific shape of segmented objects as well as the presence of a boundary with a distinct texture, making it difficult to generalise this approach to more challenging cellular imaging datasets. Therefore, there is a pressing need for advanced neural network architectures that explicitly aim to reduce merging error.

Böhm *et al.* [2] employed a multi-task approach for segmenting overlapping translucent objects. They formulated the segmentation problem as two tasks: 1) object detection, and 2) object segmentation. For the object segmentation problem, they further encoded the 2D segmentation masks into 3D sheared masks. Although this approach can reduce merging errors, it does not guarantee accurate segmentation of cell boundaries. DCAN [5] is another multi-task architecture that won the 2015 MICCAI Gland Segmentation Challenge by adding an additional branch for learning gland contours. However, in our case cell boundaries are mostly detected correctly except between neighbouring cells. Here, we aim to achieve accurate detection and delineation of cell boundaries by explicitly learning to classify pixels falling between adjacent cells.

2 Methods

We introduce DeepSplit; an architecture with one encoder branch and two decoder branches, as shown in Figure 2. The first decoder branch is for traditional segmentation. The second decoder branch (the separation branch) is explicitly used to predict the boundary pixels between adjacent cells, which motivates the name DeepSplit. Features that are learned by the segmentation decoder branch

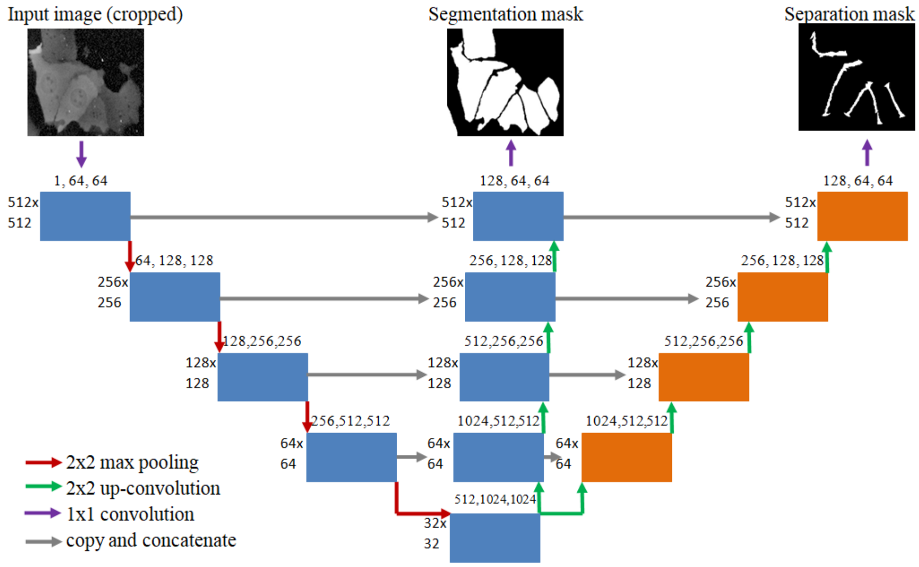


Fig. 2: **DeepSplit Architecture**. Each block consists of two 3x3 convolutional layers followed by a ReLU activation function with random dropout at a rate of 20%. The resolution of the feature maps is indicated on the left of each block (e.g. 512x512) while the number of feature maps is indicated on the top of each block (e.g. 1, 64, 64).

are shared by adding cross connections to the separation branch. Importantly, this provides substantial contextual information that aids with the separation task. These cross connections are critical to DeepSplit's performance.

2.1 Segmentation Task

The encoding branch has five blocks, where each block consists of two 3x3 convolutional layers interleaved with ReLU activation layers. Each block is followed by a 2x2 max pooling operation with stride of 2 pixels to decrease the resolution of the feature maps, descending to the next block down. The decoding branches use the same blocks but replace the max pooling operations with 2x2 up-sampling operations. Like the typical U-Net architecture, cross connections are also added from the encoding to the segmentation decoding branch. It is the combination of high-resolution, high-level context from the encoding branch, and the low resolution features describing global context from the decoding branch which permits accurate segmentation with minimal training data.

2.2 Separation Task

Since many segmentation errors are due to mistakenly merging two cells, we added an additional branch for learning the challenging cell-cell contact regions.

This branch has exactly the same configuration as the segmentation decoding branch.

2.3 Alternative architectures

In addition to DeepSplit and U-Net, we tested two other variations on the DeepSplit architecture; Branch-Net and Double-U-Net. Like DeepSplit, Branch-Net and Double-U-Net are U-Nets with added layers and a separation output. The Branch-Net architecture adds an auxiliary branch to U-Net with 3 additional convolutional blocks to retune the trained U-Net to improve separation. The Double-U-Net architecture follows the U-Net with a second, smaller U-Net (Appendix Fig. A.1 and Fig. A.2 respectively).

3 Dataset and Training

3.1 Dataset

Training is performed using an original dataset of MCF-10a epithelial breast cells *in vitro* expressing a Green Fluorescent Protein (GFP) that binds to YAP protein. This poses a challenge as different cells express different levels of the protein, making some cells much brighter than others. There also exists an enormous variety of cell shapes in the dataset. Another major challenge is that these cells tend to adhere to one another. This makes the segmentation of these images a difficult task. As these types of images are routinely acquired and tend to vary from one experiment to another, it is essential to develop flexible approaches that can work on a limited number of annotations. 50 images were manually annotated, each with a resolution of 512x512. 80% of the images are used for training, 10% for validation, and 10% for testing.

3.2 Training

Pre-processing. Histogram equalisation is applied to the raw images to enhance image contrast. We have not applied any image denoising in our experiments. Although some noise is present in the images, the loss of information in the image associated with denoising is found to be too detrimental to justify its use.
Data Augmentation. Augmentation is a key step in compensating for the shortage of labelled data. Furthermore, by introducing various operations on the image, augmentation enforces the network to learn transition invariant features. Images are shifted, flipped, rotated, and subjected to elastic deformations [16] such that 19 additional images are derived from each original image. The intuition as to why elastic transformations in particular are so effective is that cells are capable of deforming in all of these ways, so the new images contain cells which look natural and realistic.

Loss Function. A binary cross-entropy loss is used. The loss function for the segmentation task is weighted with a variety of schemes based on that used by Ronneberger *et al*:

$$\omega(\mathbf{x}) = \omega_c(\mathbf{x}) + \omega_0 * \exp(-(d_1(\mathbf{x})^2 + d_2(\mathbf{x})^2)/2\sigma^2) \quad (1)$$

$\omega(\mathbf{x})$ is the weight map. d_1 and d_2 are the distances to the nearest and second nearest cells for each background pixel in $(\mathbf{x})$. ω_0 and σ are constants that need to be set explicitly. $\omega_c(\mathbf{x})$ is a weight map attributing the same weight to every foreground pixel such that the total weight attributed to all foreground pixels is equal to that of all background pixels.

For the separation task, pixels which are part of cells are assigned a weight of 5. Pixels between adjacent cells, determined to be the white regions in the separation reference data, have a weight of 22.6. As errors in classifying background pixels will be corrected when intersected with results from the segmentation branch, we assign background pixels a weight of 1. The high weight of pixels between adjacent cells is critical to split incorrectly merged cells. The medium weight of cell pixels ensures that as few of them as possible are classified as boundary pixels, however if some are then that is an acceptable price to pay for accurate cell separation.

Random **Dropout** is used to increase the robustness and generalisability of the model. Dropout incorporates degrees of redundancy during training by forcing some nodes to have a weight of zero. This ensures that accurate segmentation or separation is not dependent on any one feature. A dropout rate of 20% is used.

Optimisation. Adam, stochastic gradient descent with and without momentum, AdaGrad, and AdaDelta were tested. The Adam optimiser is used because it converged faster than other tested optimisers in all experiments where convergence was achieved within 10 epochs.

Initialisation of the weights in the convolutional layers is conducted randomly. In the cumulative learning approach, the separation task is trained based on the pretraining from the segmentation task.

Post-processing is used primarily to clean up artefacts arising in the results. Standard Scikit Learn functions are employed to fill small holes in segmented objects or filter small segments. Additionally, a custom watershed algorithm is developed to aid the separation of incorrectly merged cells. The seed points for this algorithm are placed at the centres of cells. Labels are then propagated out from each of these seed points, with labels from different seed points theoretically meeting at places where cells have been incorrectly merged. As the GFP marker used is localised differentially between the cytoplasm and the nucleus, boundaries can occur within cells. So cutting at every border between labels would incorrectly cut many cells in half. To account for that, cuts are only performed along borders below a certain length criterion. This is found to marginally improve cell separation accuracy.

3.3 Evaluation Metrics

One of the key metrics employed is the aforementioned Dice Similarity Coefficient (DSC). This is the most commonly used metric for gauging segmentation performance, however it fails to capture the accuracy of cell separation. An incorrect merge of two cells reduces DSC very little. Consequently, models that overfit to the foreground pixels can still obtain excellent DSC scores.

To address this challenge, we propose a second metric based on cell detection performance. Cells in the predicted and ground truth mask are matched by comparing their position and area. Cells present in the ground truth masks but not detected are defined as false negatives (FN), while those present in the segmentation mask but not in the ground truth are defined as false positives (FP). Cells present in both masks are defined as true positives (TP). The cell detection score (CDS) is then calculated using the same formula as the DSC, but using these different definitions of true and false positives and negatives:

$$CDS = 2TP/(2TP + FP + FN) \tag{2}$$

Here CDS achieves two goals as if it is high, not only have the cells been well separated, the cells in the results will also have approximately correct morphologies. This is important to ensure that the network is capable of generating sufficiently accurate results for useful analysis. It is worth noting that the CDS is exceptionally punitive towards incorrectly merged cells, as two merged cells will count not only as two false negatives, but also one false positive.

4 Experiments and Results

4.1 Cumulative Learning

Cumulative DeepSplit is trained in two stages. First, we train the network to learn segmentation masks through the first decoding branch while setting the weight for the separation loss to zero. Once converged, the results from this stage are taken to be the results of the segmentation task. Effectively, this corresponds to training a typical U-Net. In the second stage, DeepSplit is retrained where the weights for the segmentation loss and the separation loss are determined empirically. We found that best results can be obtained when weighting segmentation loss by a factor of 0.3 compared to a factor of 1 for the separation loss. The cross connections between the two decoding branches provide context from the segmentation task without interfering with the training of the separation task, leading to better separation results. The final results from the separation branch are then intersected with the segmentation masks from the the first stage. In other words, pixels that are false positives in the segmentation task are suppressed by the separation results and vice versa. Table 1 shows that this approach significantly outperforms U-Net on our data set. Interestingly, training two independent U-Nets for each of the tasks did not perform well, illustrating the benefits of the proposed multi-task convolutional network.

We explored other architectures that, like DeepSplit, specify an additional separation task to gain insights into DeepSplit performance. The first adds an auxiliary branch to the end of the decoding segmentation branch (Branch-U-Net). The second follows the segmentation U-Net with another smaller U-Net (Double-U-Net). Both of these architectures achieve significant improvements on U-Net performance. However, DeepSplit produces better separation results than the other architectures, leading to superior DSC and CDS. These results demonstrate that architectures that formulate multi-instance segmentation as two tasks (segmentation and separation) can significantly improve the segmentation results. Furthermore, DeepSplit outperforming Branch-Net and Double-U-Net highlights the value of cross connections between the segmentation and separation branches, which allows sharing information from the segmentation task with the separation task.

Table 1: Proposed multi-task cumulative DeepSplit results compared to U-Net. In cumulative learning, the network is trained in two stages. Results are presented before and after post-processing to facilitate easier comparisons.

Experiment	DSC	CDS	Post-Processed DSC	Post-Processed CDS
U-Net	0.841	0.699	0.842	0.704
Cumulative DeepSplit	0.911	0.903	0.911	0.903
Cumulative Branch-Net	0.878	0.886	0.881	0.890
Cumulative Double-U-Net	0.903	0.895	0.903	0.895

4.2 Simultaneous Learning

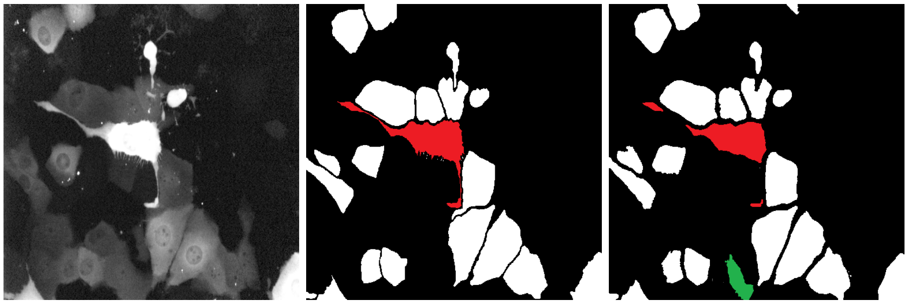
We found that training DeepSplit in two stages is critical to DeepSplit performance. Training DeepSplit from scratch to learn segmentation and separation simultaneously performed worse than a typical U-Net (Table 2). This was also true of simultaneous training for Branch-U-Net and Double-U-Net. This suggests that learning the separation mask impedes the learning of the segmentation mask. A simultaneously trained DeepSplit performs slightly better than the other architectures because the segmentation and separation branches are trained in parallel, reducing the number of layers being influenced by two competing tasks. These results reflect the effectiveness of the proposed cumulative learning approach.

5 Discussion

The results obtained in this work demonstrate the advantage of multi-task cumulative learning. The cumulative DeepSplit results in a significant improvement in

Table 2: Results for simultaneous learning of the segmentation and separation tasks as compared to U-Net.

Experiment	DSC	CDS	Post-Processed DSC	Post-Processed CDS
U-Net	0.841	0.699	0.842	0.704
Simultaneous DeepSplit	0.805	0.748	0.812	0.770
Simultaneous Branch-Net	0.688	0.702	0.689	0.706
Simultaneous Double-U-Net	0.796	0.720	0.802	0.742

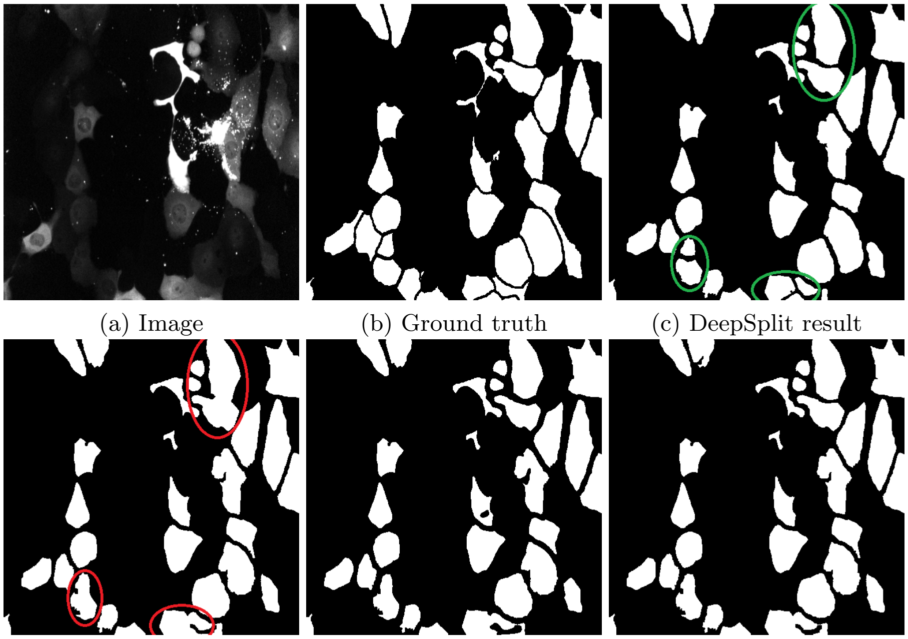


(a) Image (b) Ground truth (c) DeepSplit result

Fig. 3: Example result from Cumulative DeepSplit. The green cell has been detected despite not being present in the ground truth, demonstrating that DeepSplit can slightly improve on reference data. The red cell has long thin protrusions which the network fails to detect as part of the cell.

segmenting adjacent cells when compared to a simple U-Net approach. We illustrate the power of DeepSplit based on a small and challenging training dataset. Furthermore, upon close inspection of the results, instances are found of supposed errors in the results arguably being minor improvements on the ground truth, as seen in Fig. 3. This is in part a reflection of the extreme difficulty in accurately generating segmentation ground truth by hand, especially when there is a substantial variation in cell brightness and the cell boundaries are indistinct.

By adding a second decoding branch solely for predicting where touching segments need to be split, DeepSplit addresses one of the major challenges in cellular segmentation. U-Net has a tendency to overfit to the foreground pixels, accepting errors in boundary pixels as there are not many of them. To our knowledge, DeepSplit is the first architecture that attempts to learn to separate adjacent cells by explicitly learning to segment pixels at the boundaries between cells. Critical to DeepSplit's performance is learning segmentation and separation in two stages. This suggests that learning the segmentation task first helps the network focus its attention on features that are discriminative of foreground versus background. Once these features are learned, we then focus the attention of the network towards features that are predictive of the boundaries, while having access to segmentation features via the cross connections. Giving some



(a) Image (b) Ground truth (c) DeepSplit result

(d) U-Net result (e) Branch-Net result (f) Double-U-Net result

Fig. 4: Example results obtained by the various proposed architectures and U-Net versus the ground truth segmentation masks. Instances where DeepSplit correctly separates cells that were merged in the U-Net results are circled.

limited weight to the segmentation task allows its layers to continue producing reasonable segmentation results, ensuring that the context it provides to the separation task is useful throughout the training process. The order of these stages is based on the task difficulty and contribution to the final results. Specifically, segmenting the foreground is easier than classifying pixels between adjacent cells. Furthermore, as there are many more foreground pixels it is intuitive to learn those first.

Although this work is a significant advance from U-Net, more work is needed to address more challenging cell shapes. In particular cells with thin protrusions are being incorrectly split. This is also an issue with typical U-Nets, and remains an open challenge. More annotations of the challenging cells might be needed to circumvent this problem. Furthermore, the multi-stage training can significantly increase the computational workload associated with training the model. Future work would include developing adaptive weighting schemes of the different tasks to ensure minimal training time.

Additionally, it is observed that the evaluation metrics employed have a profound impact on the solutions found and results obtained. Each metric comes with its own biases. A naive approach employing only DSC would lead to a result that solely minimises erosion, but allows cells to incorrectly merge. Focusing on CDS alone would result in huge erosion. There is therefore a requirement to employ a range of evaluation metrics to achieve a trade-off that yields the most scientifically accurate measurements. This approach facilitates a more nuanced understanding of the outputs, which is crucial in developing biologically meaningful results.

6 Conclusions

This paper explores the problem of accurate boundary detection in cell segmentation when limited training data is available. Firstly, it is found that there exists a trade-off between boundary detection and accurate pixel-wise segmentation, such that these two tasks are best approached as separately as possible. Secondly, it is shown that access to context from a segmentation task is essential for CNNs to learn accurate separation results. The more contextual features provided from the segmentation task, the better the separation results become. Thirdly, new network architectures are explored, culminating in the development of the successful DeepSplit architecture adopting a cumulative learning approach. Together, these three developments are found to significantly improve segmentation results.

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv:1511.00561 (2015)
2. Böhm, A., Ücker, A., Jäger, T., Ronneberger, O., Falk, T.: ISOODL: Instance segmentation of overlapping biological objects using deep learning. In Proceedings of International Symposium on Biomedical Imaging,IEEE (2018)
3. Caicedo J., Roth J., Goodman A., Becker T., Karhohs K., Mcquin C., Singh S., Carpenter A.: Evaluation of Deep Learning Strategies for Nucleus Segmentation in Fluorescence Images. Journal of Quantitative Science **95**(9) (2019)
4. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the Devil in the Details: Delving Deep into Convolutional Nets. In Proceedings of the British Machine Vision Conference (2014)
5. Chen, H., Qi, X., Yu, L., Heng, PA.: DCAN: deep contour-aware networks for accurate gland segmentation. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition 2016, pp. 2487–2496 (2016)
6. Dima, A.A., Elliott, J.T., Filliben, J.J., Halter, M., Peskin, A., Bernal, J., Kociolek, M., Brady, M.C., Tang, H.C., Plant, A.L.: Comparison of segmentation algorithms for fluorescence microscopy images of cells. Journal of Quantitative Cell Science **79A**(7), 545–559 (2011)
7. Fan, M., Rittscher, J.: Global probabilistic models for enhancing segmentation with convolutional networks. In 2018 IEEE 15th International Symposium on Biomedical Imaging, pp. 1234–1238. IEEE, Washington, DC (2018)
8. Fu, J., Liu J., Wang, Y., Zhou, J., Wang, C., Lu, H.: Stacked deconvolutional network for semantic segmentation. IEEE Transactions on Image Processin. (2019)
9. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision 2017, pp. 2961–2969 (2017)

10. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K. Q.: Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4700–4708 (2017)
11. J'egou, S. ,Drozdzal, M., Vazquez, D., Romero, A., Bengio, Y.: The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 11–19 (2017)
12. Luxenburg, C., Zaidel-Bar, R.: From cell shape to cell fate via the cytoskeleton - Insights from the epidermis. Experimental Cell Research **378**(2), 232–237 (2019). https://doi.org/10.1016/j.yexcr.2019.03.016
13. Milletari , Navab, N., Ahmadi, S.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In Proceedings of International Conference on 3D Vision, pp 565–571 (2016)
14. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol. 9351. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4-28
15. Sailem, H., Rittscher, J., Pelkmans, L.: KCML: a machine-learning framework for inference of multi-scale gene functions from genetic perturbation screens. Molecular Systems Biology (2020)
16. Simard, P. Y., Steinkraus, D., Platt, J. C.: Best practices for convolutional neural networks applied to visual document analysis. In Proceedings of Seventh International Conference on Document Analysis and Recognition, pp. 958–963 (2003)
17. Taghanaki, S., Abhishek, K., Cohen, J., Cohen-Adad, J., Hamarneh, G.: Deep Semantic Segmentation of Natural and Medical Images: A Review. arXiv preprint arXiv:1910.07655 (2019)
18. Vicar, T., Balvan, J., Jaros, J., Jug, F., Kolar, R., Masarik, M., Gumulec, J.: Cell segmentation methods for label-free contrast microscopy: review and comprehensive comparison. BMC bioinformatics **20**(1),360 (2019) https://doi.org/10.1186/s12859-019-2880-8
19. Wu, Z., Shen, C., Van Den Hengel, A.: Wider or deeper: Revisiting the resnet model for visual recognition. Pattern Recognition **90**, 119-–133 (2019)

A Appendix

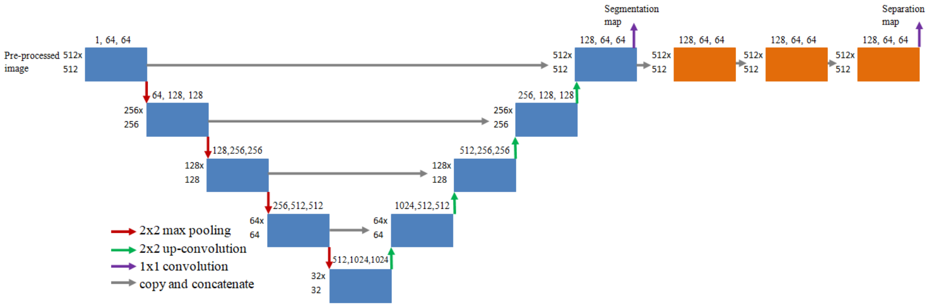


Fig. A.1: Branch-Net Architecture. The orange blocks highlight the additional convolutional layers.

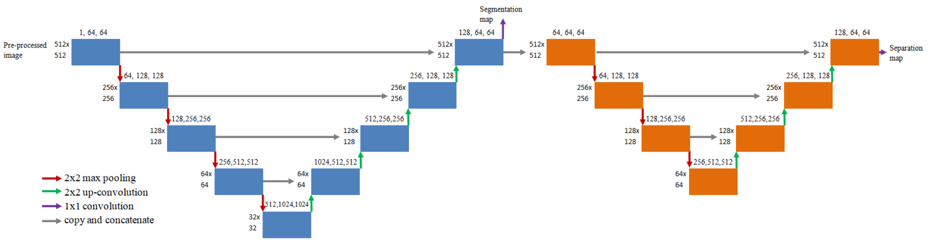


Fig. A.2: Double-U-Net Architecture. Orange highlights the additional U used for separation.