Cell Segmentation Images Using U-Net and a Watershed Transform Network

Nirit Alkalay

nirit.alkalay@gmail.com

Hila Shacham

hilasha2@mail.tau.ac.il

Tel Aviv University

Abstract

Microscopic cells or nuclei segmentation has a great value for medical and biological applications, such as describing cell behaviour like proliferation or cell death, and characterizing changes in cells like growth or migration. In this project, we present a simple yet powerful end-to-end convolutional neural network to tackle this task. Our approach combines the classical U-Net network with watershed transform deep learning network to produce an energy map of the image where each cell is represented by energy basins, from which we can extract needed information like the edges, foreground, background of the cell and more.

1. Introduction

Image segmentation is the process of classifying each pixel in an image belonging to a certain class [1]. In the medical image segmentation problem, we are interested in classifying the morphology and pathology of organs or tissues from medical images like CT, MRI, and H&E stains of biopsies.

Earlier works on image analysis were based on mathematical methods such as edge detection filters and histogram-seeking techniques. Later, trainable machine learning approaches were introduced focusing mostly on feature extraction. In the 2000s, as hardware capabilities increased, application of automatic segmentations using deep learning approaches was possible. Thanks to its capabilities in image processing tasks, up to this day, deep learning is the state-of-the-art approach for many image segmentation problems. This led to the development of new deep learning models, such as U-Net [2], and introduction of many medical image segmentation challenges. One of those challenges is the the cell segmentation and tracking challenge (CTC) [3][4][5]. Held since 2013, the challenge tries to tackle two problems: 1) segmentation, 2) and tracking of microscopic time-lapse videos of cells. Cell segmentation and tracking can help study the mechanism of cell motility which is essential to various biomedical researches such as tumorigenesis and response to therapy. For this project's scope, we decided to focus on cell segmentation only.

The current leading groups for the segmentation benchmark in CTC utilized mainly the following approaches:

1) Improvement of the image pre-processing and post-processing steps.
2) Utilization of U-Net as the segmentation model
3) Improvement of the loss function
4) Changing the targeted classes of the images.

We took a similar approach to the leading groups as we incorporated U-Net as our model and added the pre-processing, post-processing steps. In addition, we tried to enhance U-Net by adding an automatic post-processing step using Deep Watershed Transform network (DWTNet) [6], trying to replace the classical post-processing procedure. We combined U-Net with DWTNet creating a network we called UWDTNet.

Our approach has several key advantages: Fast estimation of cell segmentation, regardless of the number of object instances. Improved separation of segmented adjoined cells. The possibility of extracting information such as cells area, edges, etc.

In the following section, we first review related work. We then present the details behind our intuition and model design, followed by an analysis of our model's performance.

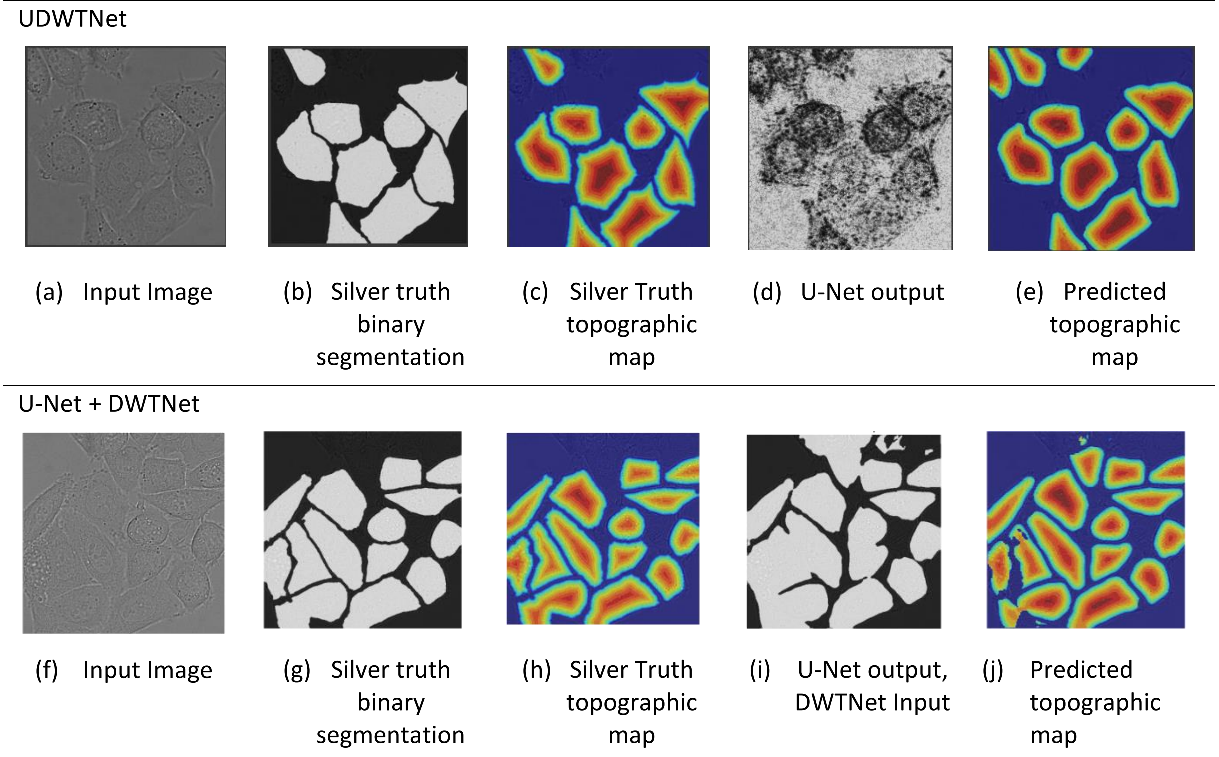

UDWTNet
(a) Input Image
(b) Silver truth binary segmentation
(c) Silver Truth topographic map
(d) U-Net output
(e) Predicted topographic map
U-Net + DWTNet
(f) Input Image
(g) Silver truth binary segmentation
(h) Silver Truth topographic map
(i) U-Net output, DWTNet Input
(j) Predicted topographic map

Figure 1: First row: Sample prediction from our UDWTNet. Input image (a) is taken into UDWTNet, which is a combined model of two networks, U-Net + DWTNet. The target of the training is the topographic mask (c), and the output is (e). The evaluation of accuracy is done in comparison of (e) with (b) binary silver truth or (c) the topographic silver truth. The output of U-Net's layer can be seen in (d).

Second row: Sample prediction from our DWTNet trained after U-Net separately. Image (f) and U-Net's output (i) are taken as the input to DWTNet. The target of the training is the topographic mask (h), and the output is (j). The evaluation of accuracy is done in comparison of (j) with (g) binary silver truth or (h) the topographic silver truth.

2. Related Work

CTC provides a leaderboard for the segmentation challenge. The first three winning groups for 2020 (CALT-US [Fidel AG Peña et al], Mu-Lux-Cz [Filip Lux and Petr Matula], UVA-NL[Andreas Panteli et al]) [7] all based their segmentation on the U-Net network.

U-Net [2] is a convolution neural network created for biomedical images segmentation. Its architecture aims to deal with a small training set of images, a problem prevalent in the deep learning medical field. It consists of a contracting path and an expansive path. The contracting path is built from successive convolutional layers. The spatial image dimensions are reduced while the feature channels increase. The expansive path includes consecutive up-convolutional layers with decreasing feature channels. The up-convolutional blocks are concatenated with the contracting path's convolutional blocks which include the same number of features. The high number of features throughout the network is said to compensate for the low amount of training images.

The leading groups from the segmentation benchmark for the data that we used, the HeLa cells DIC (differential interference contrast) microscopy images are denoted as (from 1st place to the 3rd): **CALT-US**, **Mu-Lux-Cz**, **UVA-NL** . The UVA-NL group used Mu-Lux-Cz's network, so we will discuss only the former two. Both groups did not change much of the U-Net network and instead decided to focus on fine tuning the pre-processing step, loss function, labels, and the post processing step.

CALT-US decided to focus on emphasizing the edges of the cells as U-Net poorly classifies adjoining regions [8] [9]. They separated the labels into four groups: background, cell, touching cell regions, and gaps between cells. The loss function used was cross entropy combined with J statistics. The modification to cross entropy is said to put more weight to the edges' classes. The post processing step involved reclassifying the labels into background and cells, removing small objects, and closing the edges.

Mu-Lux-Cz used a U-Net with fewer feature channels which outputted cell markers and cell boundaries masks [10][11]. Cell markers are points which represent the cells. The loss function was an average of the weighted cross entropy losses of the boundaries and markers. The pre-processing step involved data normalization and creating weight maps of the images which reflected the importance of each pixel. The post-processing step involved watershed to infer cell boundaries and background.

All groups used data augmentations to the input images such as random rotations, mirroring, gamma correction, random distortions, etc.

3. Our approach

We decided to take a similar approach to the winning groups by playing with the pre-processing step, labels, loss function and post processing. Though eventually we used mainly binary labels, foreground (cells) and background, and used cross entropy as the loss function. Our main difference is that we incorporated a watershed post-processing step as a neural network. As U-Net has a problem segmenting adjoining cells [8], a post processing step separating the cells is necessary. We based our network on Min Bai and Raquel Urtasun's Deep Watershed Transform network (DWTNet) [6]. Watershed transform is a technique in mathematical morphology. The technique is applied to a grayscale image, such that we get topographical regions of the image. The idea is that we metaphorically partition the image into basins, regions, separated by dams, such that if we flood the image with water, we prevent different waters from different regions to merge. The basins basically correspond to homogeneous regions of the image. Each basin is called energy level.

DWTNet combined a direction network and a watershed network. The direction network was necessary to learn the energy landscape of the image, the direction of descent of the watershed energy. The direction network predicts unit vector maps such that each vector at each cell pixel points directly away from the nearest cell boundary. This is then later taken by the watershed network which produces a watershed transform map.

We combined U-Net and DWTNet such that U-Net's output was DWTNet's input. We later combined the networks into one network we named UDWTNet (U-Net and DWTNet).

4. Data

Our data are microscopic images of HeLa cells taken from CTC's website [12]. The data is denoted DIC-C2DH-HeLa by CTC, which means we have high resolution 2-dimensional tiff images from DIC microscopy of HeLa cells. The challenge provided datasets from two different time-lapse videos, each containing 84 grayscale time series images of size 512 x 512 pixels each.

All the images had silver truth, computer-generated reference annotations. Only some image had gold truth, human-made reference annotations, hence we used only the silver truths. The silver truth contained instance segmentation masks that cover the whole area of the cells. Each cell was annotated differently.

We separated the data into training, validation and test data. The first time-lapse video's images were used for the training and split into 70% training and 30% validation, while the second time-lapse video's images were used as test data.

5. Methods

5.1. Pre-processing

5.1.1. Annotations preparation

The target for our U-Net is a semantic segmentation mask. Since U-Net performed poorly on instance segmentation masks, i.e. it could not classify each cell differently; we converted the instance segmentation into a semantic one. We mainly worked with two classes: background – 0, and foreground (cells) – 1. We added the possibility of a third class: cell boundaries – 2, although it did not improve our U-Net's result so we explored mainly the former two.

The target for DWTNet is a topographic map or 16-energy-level map. The instance segmentation was converted into a gradient grayscale image where each pixel of a cell represented the distance from the closest background element. Then we divided the gradient image into 16 regions to create a topographic map. Energy level 0 corresponds to the background, while higher energy levels correspond to the interior of a cell.

For visualization, see Figure 1.

5.1.2. Data Augmentation and normalization

Since the data was relatively scarce, we used data augmentation, i.e. slightly modifying the data, to increase the diversity of the data. We applied randomly: Gaussian blur, cropping (maximum cropping of 20% of the image), sharpening, rotation, shearing, zooming, translation, mirroring, elastic distortion, and perspective distortion. The same augmentation was applied for the semantic segmentation and topographic masks.

The images were also z-normalized.

5.2. Network architecture

To further help U-Net with the segmentation we added a network we called DWTNet which produces a watershed transform. The exact architecture of DWTNet is described in [6]. After experimenting with U-Net, we noticed it has a slight problem segmenting adjoining cells with faint boundaries between them. A supervised watershed transform is said to help with that.

The input to DWTNet is the images augmented with the semantic segmentation outputted from U-Net as a second channel. The target is the topographic watershed map we prepared ahead with 16 labels corresponding to the energy levels. DWTNet is composed of a direction network (DN) and a watershed transform network (WTN).

DN has consecutive convolutional layers, or a downsampling path, where the input size is reduced while the number of features increases. Some layers are bypassed via skip connections and concatenated to the downsampling path's result. These are then upsampled to the input's dimensions. Skip connections are said to preserve local contextual information from the downsampling path, while the upsampling recovers spatial information [13]. **WTN** is a generic CNN.

We based DWTNet on Bai and Urtasun's provided code [14], converting it from TensorFlow into PyTorch. In order to deal with the network's vanishing gradients, we changed the activation function ReLu into LeakyReLu, added BatchNorm layers, and used Xavier initialization for the convolutional blocks.

In addition, our DWTNet had four times fewer features. We have noticed that higher number of features did not help the network learn better and lower number of features made the network faster.

Finally, we created a combined network of U-Net and DWTNet which we called UDWTNet. The target was the topographic watershed maps.

5.3. Loss function

We used cross entropy as our loss function. We tried using dice loss instead, as it is said it improves the results when there is class imbalance, such when the background area is quite substantial [15]. Dice loss measures overlap between the ground truth and the prediction. Besides U-Net, there was not much difference between cross entropy and dice loss, so we stayed with cross entropy.

5.4 Post-Processing

The post-processing step involved removing small holes inside the cells and removing small objects.

6. Experimental Evaluation

We compare the results between our 3 models:

1) **U-Net**: Trained for 100 epochs.

2) **U-Net + DWTNet**: Trained U-Net and then trained DWTNet using the output of U-Net as its input. U-Net was trained for 100 epochs, and DWTNet for 100 epochs.
3) **UDWTNet**: Trained for 200 epochs.

Our evaluation measurement used **IoU**, the intersection-over-union precision, comparing the estimated results and the silver truth. We used **binary IoU** to compare between the semantic segmentation predictions vs. the silver truth. The binary classes were the foreground (cells) and the background. For U-Net+DWTNet and UDWTNet we also used a **topographic IoU**. The topographic IoU compared the predicted topographic map vs. the topographic silver truth.

6.1 U-Net

The loss and IoU evaluations for U-Net can be seen in Table 1. U-Net trained with dice loss did slightly better than U-Net trained with cross entropy loss, for example the test IoU for U-Net trained with dice loss was 0.596 while the one with cross entropy was 0.545. The loss difference was much more noticeable than the IoU measurement, for example, the test loss with dice was 0.298, while with cross entropy 0.414.

By comparing the behavior of the training and validation losses and IoU of U-Net trained with dice loss vs. cross entropy loss, as seen in Figure 2, one can see the net behavior with cross entropy included much more aberrations. That said, the segmentation images both U-Nets outputted, did not show much visible differences. Which might indicate that the loss measurement of U-Net is not the most reliable measurement, and IoU might be preferable.

Looking at Figure 3, The main problem with U-Net's segmentation is that it had problems segmenting adjoining cells. Touching, yet different, cells are segmented as a single instance. We tried training the net with three labels, where we added a third label of cell edges, in order to solve the issue with the adjoining cells. However, the training results seemed to be worse, hence we stayed with two labels only.

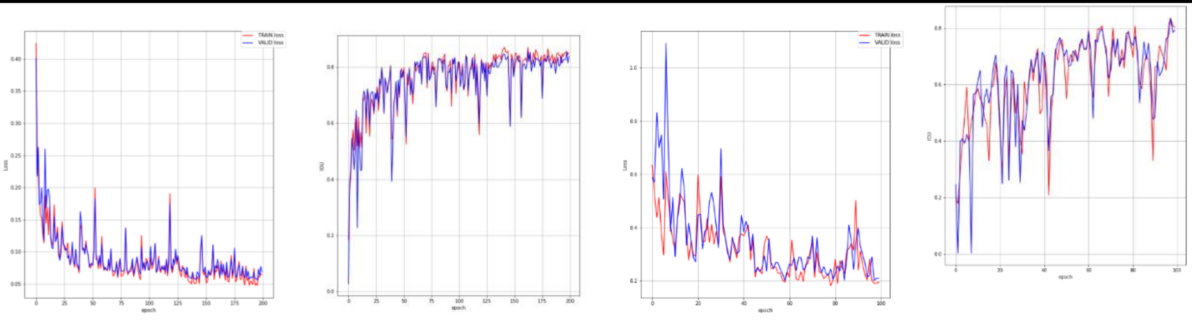
Another issue, which was similar with all the models, was a discrepancy between the results of the test vs. the training. We will discuss later why it might have happened.

6.2 U-Net + DWTNet

DWTNet received as its input an image combined with the semantic segmentation outputted from U-Net. Which meant that if

Model	Training			Validation			Test (Average)		
	Loss	Binary IoU	Topographic IoU	Loss	Binary IoU	Topographic IoU	Loss	Binary IoU	Topographic IoU
U-Net with dice loss	0.068	0.812	-	0.063	0.835	-	0.298	0.596	-
U-Net with cross entropy	0.194	0.802	-	0.209	0.792	-	0.414	0.545	-
U-Net + DWTNet	0.682	0.866	0.566	0.779	0.688	0.485	1.842	0.210	0.004
UDWTNet (epoch 199)	0.735	0.876	0.576	0.836	0.854	0.604	-	-	-
UDWTNet (epoch 200)	0.917	0.814	0.517	1.034	0.681	0.431	2.211	0.432	0.000

Table 1: Loss and IoU evaluations comparing the different models: U-Net, U-Net + DWTNet, UDWTNet. We compared U-Net when it was evaluated with dice loss vs. cross entropy. There was no topographic target in U-Net, hence the topographic IoU for U-Net is not relevant. We also displayed the results for the last two epochs in UDWTNet to emphasize the differences. We ran a test for UDWTNet with 200, hence the test results for 199 epochs are not relevant.



(a) Dice - Loss (b) Dice - Binary IoU (c) Cross entropy – Loss (d) Cross entropy – Binary IoU

Figure 2: Measurements of the loss and IoU of U-Net per epochs (100 epochs overall). Training results are marked in blue, while the validation in red. Graph (a) refers to the loss per epochs of U-Net trained with dice loss, while (b) is its IoU per epochs. Graph (c) refers to the loss per epochs of U-Net trained with cross entropy loss, while (b) is its IoU per epochs.

the semantic segmentation of U-Net was bad, it could have affected the performance of DWTNet. However, as can be seen from Figure 1, even though the segmentation from U-Net (i) is not optimal, the prediction (j) seems closer to the topographic silver truth (h) and the input (f). Actually, the predicted segmentation was closer to the input than the silver truth, as the silver truth missed some cells, while it seems our network tried to learn these missing cells anyway.

As with U-Net, U-Net + DWTNet seemed to have discrepancies between the results of the training and the test, see Table 1. The discrepancies are even greater than those from U-Net, for example, the binary training IoU of U-Net + DWTNet was 0.866, while the test's was 0.210.

Also, it seemed as the topographic IoU was always lower than the binary IoU. Which is logical as the regions of the 16-energy levels are smaller than the cells' area, so the topographic IoU is more affected by slight differences.

Another thing worth noting is that without batch normalization, DWTNet simply did not converge due to vanishing gradients.

Overall, it seemed as DWTNet helped U-Net solve the issue of adjoining cells. Even though the IoU and loss of U-Net + DWTNet are not necessarily better than those of U-Net alone, a quick glance at the training outputs seems to show that DWTNet did help U-Net segment the cells better.

6.3 UDWTNet

UDWNet is a combined network of U-Net and DWTNet. Even though the network is quite large, its size did not seem to affect the learning time of the network. U-Net and DWTNet seemed to train at a rate of 10-12 seconds per epoch, and so did UDWTNet. However, it did require more epochs to train.

In comparison to U-Net + DWTNet from Table 1, UDWTNet did not seem to improve the loss and the IoU. However, looking at the segmentation results, as can be seen in Figure 1, UDWTNet seemed to do a slight better job.

An issue that arose with both UDWTNet and U-Net + DWTNet models was that the results tended to oscillate a bit, as can be seen from Table 1 while comparing the results from the 199th epoch vs. the 200th epoch of UDWNet. UDWTNet did not necessarily oscillate more than DWTNet alone as can be seen from Figure 4.

An interesting result that we got from UDWTNet is that the last layer of U-Net's part in UDWTNet seemed to output a different result than the semantic segmentation. It seems as if this "U-Net" layer studied the contrast or the gradient of the picture, rather than the semantic segmentation.

6.4 Comparison to the leading groups

CALT-US, Mu-Lux-Cz, UVA-NL got 0.87, 0.863, 0.853 scores for their segmentation [7]. The segmentation was measured with IoU, however, it was measured for instance segmentation and not semantic segmentation. While our train binary IoU did seem to be on par with the leading groups, the test results were far from it.

7. Conclusions

We combined U-Net with a watershed transform network, DWTNet, creating a network we named UDWTNet, in order to segment microscopic images of cells. While accuracy measurements did not indicate any improvements to U-Net's segmentation, simple observations to the results did seem to indicate that UDWTNet helped U-Net segment the cells. It helped especially solving U-Net's problem of not segmenting adjoining cells.

One might ask what is the benefit of a watershed transform network over a non-trainable transform. According to Urtasun et al. [6], classical watershed tends to over segment. Also, from playing around with different types of post-processing methods for U-Net, such as erosion and watershed, these did not tend to segment well adjoining cells.

Another benefit is that we can extract much more information using a topographic segmentation. For example, we can extract edges or the centre of the cells.

The main problem with our UDWTNet was the difference between the training results versus the test ones. This might be due to overfitting of the training, or an unbalanced distribution of the training input. We trained on one set of time-lapse images, having cells stay in similar locations throughout the training. Perhaps mixing between the sets, and increasing the amount of training images, would have helped.

A serious issue to note was that our targets for training were silver truths and not ground truths. The silver truth clearly missed cells, while our models tried to learn those.

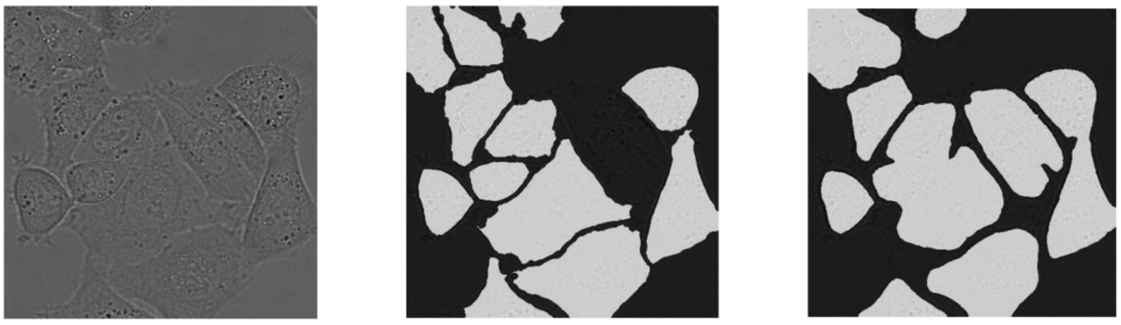
For future work, the first goal should be decreasing the gap between the training and test's results. As mentioned before, this might be done with mixing the training image sets and increasing the number of training inputs. Another method is perhaps improving the pre- and post-processing of the network, for example, enhancing the contrast of the edges and gaps between the cells.

Out network is also substantially big, so it might be beneficial to reduce its memory cost. One possible way is to include dilated convolutions instead of regular ones.

One thing we noticed throughout the training is that all the networks tended to pay attention to organelles and treated them as background. So a future work might be including an additional class for the organelles.

Finally, an upgrade to our network would be turning it from doing semantic segmentation into instance segmentation.

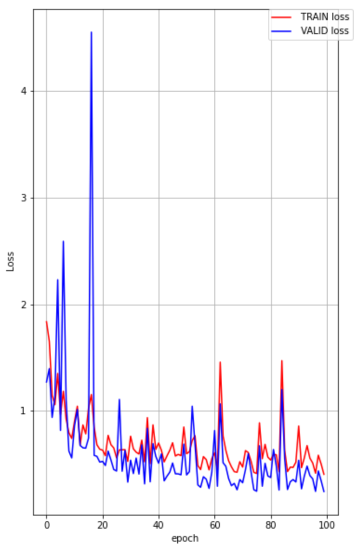
Our code can be found on:
https://github.com/hilasha2/UDWTNet



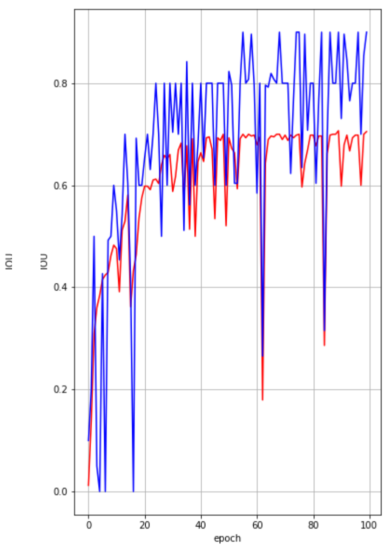
(a) Input Image (b) Silver truth binary segmentation (c) Our U-Net output

Figure 3: Sample prediction from our U-Net. Input image (a) is taken into our U-Net, which then outputs an estimation of the binary segmentation (c). The target of the training is a semantic mask (b).

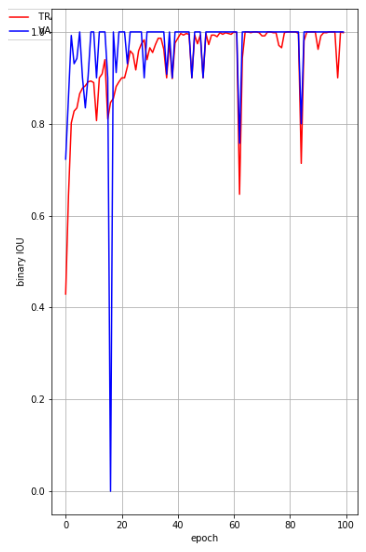
U-Net + DWTNet



(a) Cross entropy loss

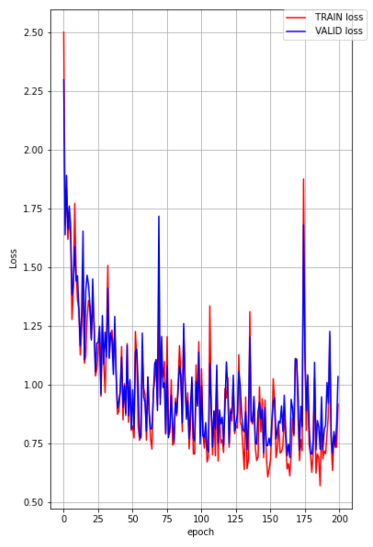


(b) Topographic IoU

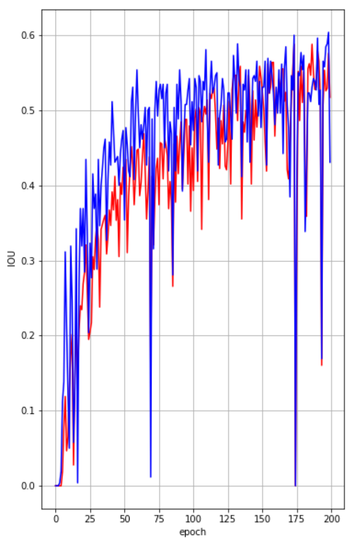


(c) Binary IoU

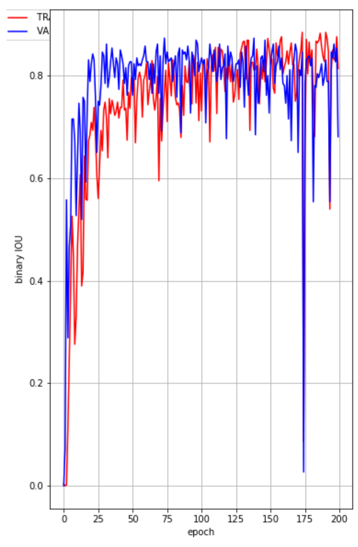
UDWTNet



(d) Cross entropy loss



(e) Topographic IoU



(f) Binary IoU

Figure 4: Training graphs of the U-Net + DWTNet model (first row), and UDWTNet (second row). Training results are marked in blue, while the validation in red.
Graph (a) refers to the loss per epochs of DWTNet after U-Net was train, (b) is its topographic IoU per epochs, (c) its binary IoU per epochs. DWTNet was trained for 100 epochs.
Graph (d) refers to the loss per epochs of UDWTNet, (e) is its topographic IoU per epochs, (f) its binary IoU per epochs. UDWTNet was trained for 200 epochs.

8. References

[1] Hesamian, M.H., Jia, W., He, X. *et al.* Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges. *J Digit Imaging* **32,** 582–596 (2019). https://doi.org/10.1007/s10278-019-00227-x

[2] Ronneberger O., Fischer P., Brox T. (2015) 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. https://doi.org/10.1007/978-3-319-24574-4_28

[3] "Cell Tracking Challenge", 2012. [Online]. Available: http://celltrackingchallenge.net/. [Accessed: Oct. 2020].

[4] Martin Maška, Vladimír Ulman, David Svoboda, Pavel Matula, Petr Matula, Cristina Ederra, Ainhoa Urbiola, Tomás España, Subramanian Venkatesan, Deepak M.W. Balak, Pavel Karas, Tereza Bolcková, Markéta Štreitová, Craig Carthel, Stefano Coraluppi, Nathalie Harder, Karl Rohr, Klas E. G. Magnusson, Joakim Jaldén, Helen M. Blau, Oleh Dzyubachyk, Pavel Křížek, Guy M. Hagen, David Pastor-Escuredo, Daniel Jimenez-Carretero, Maria J. Ledesma-Carbayo, Arrate Muñoz-Barrutia, Erik Meijering, Michal Kozubek, Carlos Ortiz-de-Solorzano, A benchmark for comparison of cell tracking algorithms, *Bioinformatics*, Volume 30, Issue 11, 1 June 2014, Pages 1609–1617, https://doi.org/10.1093/bioinformatics/btu080

[5] Ulman, V., Maška, M., Magnusson, K. *et al.* An objective comparison of cell-tracking algorithms. *Nat Methods* 14, 1141–1152 (2017). https://doi.org/10.1038/nmeth.4473

[6] M. Bai and R. Urtasun, "Deep Watershed Transform for Instance Segmentation," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 2858-2866, doi: 10.1109/CVPR.2017.305

[7] "Cell Segmentation Benchmark", 2020. Cell Tracking Challenge. Available: http://celltrackingchallenge.net/latest-csb-results/ [Accessed: Oct. 2020]

[8] Guerrero-Peña F.A., Fernandez P.D.M., Ren T.I., Cunha A. (2019) "A Weakly Supervised Method for Instance Segmentation of Biological Cells". In: Wang Q. et al. (eds) Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data. DART 2019, MIL3ID 2019. Lecture Notes in Computer Science, vol 11795. Springer, Cham. https://doi.org/10.1007/978-3-030-33391-1_25

[9] Guerrero-Peña F.A., Fernandez P.D.M., Ren T.I., Cunha A. . "CALT-US", 202. Cell Tracking Challenge. Available: https://public.celltrackingchallenge.net/participants/CALT-US.pdf [Accessed: Oct. 2020]

[10] Filip Lux and Petr Matula "MU-Lux-Cz", 2020. Cell Tracking Challenge. Available: https://public.celltrackingchallenge.net/participants/MU-Lux-CZ.pdf [Accessed: Oct. 2020]

[11] Filip Lux and Petr Matula (2020) "Cell Segmentation by Combining Marker-Controlled Watershed and Deep Learning". https://arxiv.org/abs/2004.01607

[12] "2D+Time Datasets", 2020. Cell Tracking Challenge. Available: http://celltrackingchallenge.net/2d-datasets [Accessed: Oct. 2020]

[13] Theano Development Team , "Fully Convolutional Networks (FCN) for 2D segmentation
". Deep Learning Tutorials. http://deeplearning.net/tutorial/fcn_2D_segm.html [Accessed: Oct. 2020]

[14] M. Bai and R. Urtasun, (2017). Deep Watershed Transform [Source Code] https://github.com/min2209/dwt

[15] Raj Bharath, "A Simple Guide to Semantic Segmentation", (May 2019). TopBots. Available: https://www.topbots.com/semantic-segmentation-guide/ [Accessed: Oct. 2020]