StemNet: A Temporally Trained Fully Convolutional Network for Segmentation of Muscular Stem Cells

Martin Isaksson*, Joakim Jaldén†
*School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden
Email: misakss@kth.se
†School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden
Email: jalden@kth.se

***Abstract*—In biomedical research, time-lapse microscopy is an important tool to be able to study processes which are too slow for humans to observe. This technique is powerful since it gives information about how parameters of single cells changes over time.**

The problem to be solved in this project is to segment MuSC (Muscular Stem Cells) in images and to classify them. This is done by using a discriminative model trained using supervised learning. The network is inspired by the architecture of the U-net, but extended by using temporal data to increase its performance.

The network is trained on images from the time-lapse sequence, where the temporal aspect is used to create a short-term memory for the network. The results are compared to a network of the same architecture but without the temporal aspect in the training.

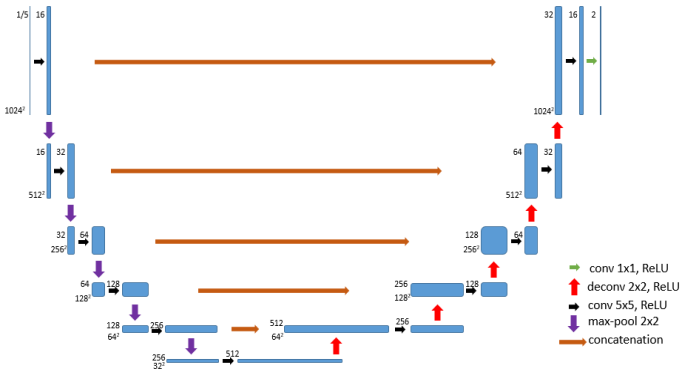

Fig. 1. The network architecture. For temporal training a 5 channel input is used, while a 1 channel input is used otherwise.

I. Introduction

To find the outlines, or area, of an object in images is known as segmentation. By segmenting stem cells, i.e. retrieving the contours/regions of the cells, and also tracking each cell and finding the patterns of its path and splits (also known as cell tracking), one can draw medical conclusions. This can be done by using algorithms on these outlines and tracking paths to extract biologically interesting information such as lineage trees, cell sizes and migration speeds [1].

To get a good tracking result, a good segmentation result is important. Today the state-of-the-art in image segmentation is given by fully convolutional networks [2]. Deep learning can be used for many different purposes. It can e.g. be used to classify objects in an image, to classify a speaker, generate words and music, generate images, and much more.

The problem to be solved here is to segment MuSC and to classify them. To do this we train a discriminative model using supervised learning. The network will consist of convolutional and deconvolutional layers, also known as a fully convolutional network. First the input image, containing stem cells, will be downsampled to compressed features and then these features will be upsampled where the output has the same size as the input image, with the stem cells segmented and classified. This is possible because the downsampling, and later upsampling steps, allow the segmentation to gain contextual knowledge from a wider spatial area when classifying single pixels as belonging to cells or background.

II. Network Architecture

The fully convolutional network contain 12 layers in total (downsampling and upsampling) and the architecture can be seen in Fig. 1.

The different operations used in the network are [3]:

- Convolutional layers.
- Max-pooling operations.
- ReLU as activation function.
- Transposed convolutional layers [4].
- Softmax per pixel.
- Weighted cross-entropy as cost function (weighted because of imalanced classes in the data).
- Momentum based SGD as optimizer.
- Dropout [5].

The networks will have the same architecture throughout the whole paper, but the input will be of different number of channels. The fully convolutional network architecture will be trained by using an input of 1 channel (grayscale) and by using an input of 5 channels (temporal information), generating two different networks. In more detail, for every input frame, frames in the sequence of the time-lapse that are before (two frames) and after (two frames) the input frame will also be used as input with that input frame. If the input frame is denoted as $X_{i,j}$ and the output label for that frame is denoted as $Y_{i,j}$, where

TABLE I
THE HYPERPARAMETERS OF THE NETWORKS.

Hyperparameter	Value
Dropout	0.25
Learning rate	0.15 (with decay)
Decay rate	0.95
Momentum	0.2
Max-pooling	2x2, stride 2
Deconvolution	2x2, stride 2
Convolution	5x5, stride 1
Cost function weight	2 for the unrepresented class (stem cells)

i is the sequence number and j is the frame number, then the other frames sent into the network for training will be $X_{i,j-2}$, $X_{i,j-1}$, $X_{i,j+1}$ and $X_{i,j+2}$ even though the label used to compare the output with is only $Y_{i,j}$.

III. TEMPORAL SEGMENTATION AND CLASSIFICATION

When using a neural network for the task of segmentation, one want that the output is the input image with a highlighted area of the object to be segmented. To train a neural network to perform this, it is necessary to use binary segmentation masks as labels (here with the same height and width as the input image), so that the network can backpropagate the pixel errors of each corresponding pixel of the output/prediction. In this way the network will be trained to learn on a single pixel level.

If one has access to sequential data, e.g. a time-lapse recording as in this case, it is possible to use the temporal information to make the network better learn what is e.g. a stem cell and what is not. Since the stem cells (when looking at this case) are moving and everything else is static in the sequences, the network can more easy understand what is a stem cell and what is not.

IV. EXPERIMENTS AND RESULTS

A. Experiments

Two different networks, Network_1 (1 channel input) and Network_5 (5 channel input), were trained on the same setup of hyperparameters. A momentum optimizer was used to minimize the cost function, which was weighted because of the class imbalance. Dropout was used to prevent overfitting, i.e. no regularization term was added to the cost function and no batch normalization since the batch size was one image. The networks were trained on a dataset of 24 997 images of size 1024x1024, where all images have been preprocessed by removing the sequential background and cropped or padded to the size 1024x1024. Each input image has a corresponding segmentation mask as label/truth. 19 998 images were used for training, 4 998 images were used for validation and 1 image was used for test/prediction. The (most relevant) experimental hyperparameter settings can be seen in Table I.

B. Results

Network_1 learns to finds the correct MuSC pixels quite fast, but it takes longer to learn that other objects in the image aren't MuSC. Network_5 is the opposite, it takes more time to find the correct MuSC pixels, but it learns to sort out the other objects in an image quite fast. The results of the two different networks can be seen in Table II, where the F1 score is applied to the two pixel classes, foreground (MuSC) and background.

TABLE II
THE RESULTS OF THE NETWORKS, WHERE THE RESULTS ARE AVERAGED OVER THE TEST DATA. THE NETWORK INDICES CORRESPONDS TO THE NUMBER OF INPUT CHANNELS.

Measure	Network_1	Network_5
F1 Score	0.83	0.70
Incorrectly classified background pixels	0.000157	0.000134
Correctly classified MuSC pixels	0.83	0.68

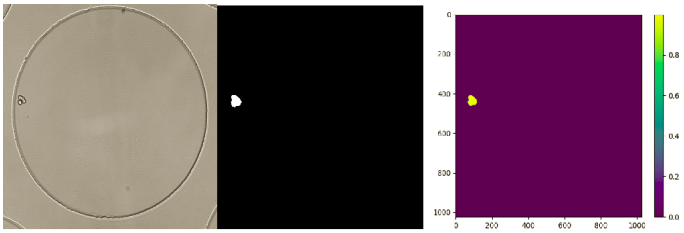

Fig. 2. The achieved result as a heat map. The left image is the input (1 channel), the image in the middle is the ground truth, i.e. the segmentation mask, and the right image is the prediction of the fully convolutional network.

An example of the performance can be seen in Fig. 2.

V. CONCLUSION

Network_1 performs well in this task, since there aren't too much complexity in the images. Network_5 learns faster to sort out objects that aren't MuSC because of the use of temporal information. Though it is quite slow to learn to be accurate of classifying the MuSC pixels correctly.

For future research it would be interesting to train a deeper network (better for more complex data) when using the 5 channel input, since that data is more complex than the 1 channel input.

ACKNOWLEDGMENT

The first author would like to thank his friend and business partner Robert Lundberg for his helpful discussions.

REFERENCES

[1] K. Magnusson, "Segmentation and tracking of cells and particles in time-lapse microscopy," School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Tech. Rep., 2016.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," Computer Science Department and BIOSS Centre for Biological Signalling Studies, University of Freiburg, Freiburg, Tech. Rep., 2015.

[3] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015.

[4] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," MILA, Universit de Montral and AIRLab, Politecnico di Milano, Montreal and Milano, Tech. Rep., 2016.

[5] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.