

# CNNs for Segmenting Confluent Cellular Culture

Bruno Beltran (bbeltr1) and Nalin Ratnayake (nalinr)

## I. BACKGROUND

One of the prime examples of single-cell high-throughput microscopy’s success is the study of the biochemical and structural basis of cell migration. Cell migration is integral to the dramatic rearrangement of cells during normal human development, as well as in the spread of cancers during metastasis and tumor growth. While chemotaxis, the migration of *isolated* cells in response to chemical queues, has been well studied over the past few decades, the mechanisms by which *groups* of cells migrate collectively in tissues, the more physiologically relevant environment to development and cancer, is an open question in modern cell biology.

Surprisingly, one of the largest challenges in analyzing data from cell migration experiments is reliable identification and tracking of individual cells. While nuclear markers can be used for tracking the cell centers, this does not allow one to delineate the extremities of cells and distinguish between the cytoplasm of neighboring cells. In development and cancer, cellular migration takes place in a crowded environment, and recent studies have attempted

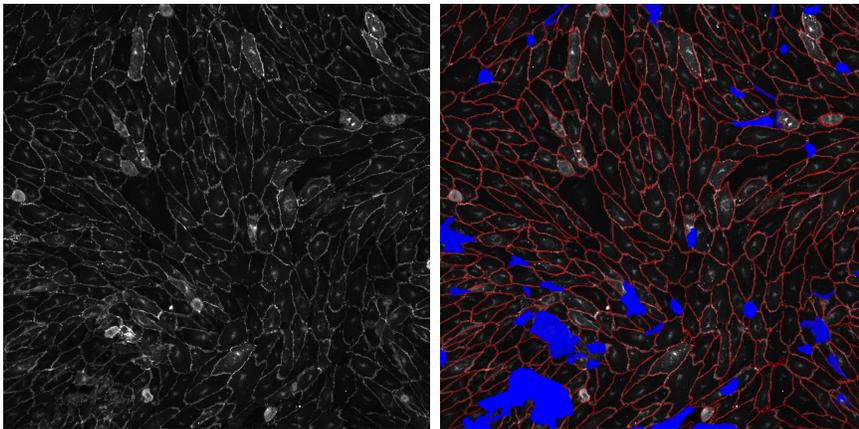


FIG. 1: Representative example of our data. LEFT: Unlabeled data. Pixel intensity corresponds to abundance of VE-cadherin-YFP. RIGHT: Labeled data. Red pixels represent ground-truth “is membrane” label. Blue pixels correspond to sections omitted from training.

to replicate this environment using confluent cell cultures (i.e. cultures where there is not empty space between neighboring cells). This increase in biological relevance has led to a concomitant increase in the difficulty of segmentation. Fluorescently labeling the plasma membrane of cells allows for researchers to manually identify inter-cell boundaries, but are not clean enough for traditional segmentation methods to produce robust results. Thus, *the application of single-cell techniques to studying collective migration is currently hindered by a lack of robust segmentation software for confluent culture.*

We propose to implement a pixel-by-pixel “sliding window” convolutional neural network (CNN)-based classifier to identify the fluorescently labeled cell membranes in confluent cell culture. The network input will be a fluorescent image, as in the left panel of Figure 1, and the output will be a binary image labeling the pixels that are contained in a cell membrane (as in the red pixels of the right panel of Figure 1). Generating pixel labels for each frame of a movie in this way will allow us to identify the spatial extent of cells over time and study their morphological properties as they migrate collectively.

## II. DATA SET

By using a VE-cadherin-YFP construct, which localizes to each cell’s plasma membrane (see Figure 1), the Meyer lab at Stanford has imaged confluent sheets of live migrating human vascular epithelial cells as they move around and interact with each other. Movies of these cells were taken with a 20x (.75 NA) objective on an epifluorescence microscope.

On disk, this produced several dozen 60 frame movies, with 2160-by-2160px resolution in two channels. The first, YFP channel contained the absorbance corresponding to the abundance of our VE-cadherin label, roughly labeling the cell membrane. The second, DAPI channel was used to track the cell nuclei, allowing us to robustly label the cell membranes by hand for supervised training.

For the actual training, we used 8 frames randomly sampled from different movies (and containing some several thousand cells altogether), which we annotated by hand in GIMP pixel-by-pixel. Each pixel was labeled as either M—part of a cell’s membrane—or NM—not part of a cell’s membrane. Each pixel was then also labeled as T—to be used during training—or NT—to not be used during training. Finally, since the pixel intensity distribution is strongly heavy-tailed, we normalized each image to the minimum intensity that contained 95% of the pixels.

A representative frame before and after labeling is shown in Figure 1.

## III. THRESHOLD-BASED SEGMENTATION

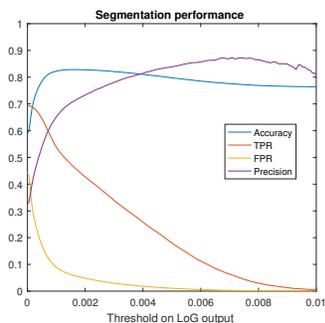


FIG. 2: Performance curves vs LoG threshold (AU).

This is impossible to extend to the epithelial cultures of interest due to their amorphous and rapidly changing shapes. Additionally, significant background noise is added to the system from internalized plasma membrane vesicles which result in bright fluorescent regions within the cell, which are classified as membrane by threshold-based methods. Both team members’ labs specialize in automatic image segmentation of cellular culture, so we were able to implement a segmentation pipeline following accepted best practices [1, 2] to compare to the output of the CNN.

While there are many schemes for threshold-based segmentation, our general work flow was as follows: (1) smoothing using an adaptive Wiener filter ( $8 \times 8$  pixel adaptive window), (2) edge detection using a threshold after applying a Laplacian of Gaussian filter

The standard methods for automated cell segmentation in biological research primarily rely on thresholding of filtered images combined with ad-hoc downstream morphological analysis of connected components to identify regions of interest. These techniques have been quite successful for segmenting isolated cells. However, attempts to extend these techniques to images of confluent culture suffer from the “leakiness” of the detected cellular boundaries. That is, if the entire cell membrane is not detected, it is impossible for the algorithm to know that two adjacent cells are not just one large cell.

One common way to get around this in high-throughput bacterial experiments is to use a model-fitting based approach, where the shape of the bacterial cell is assumed to take some simple functional form which is then fit to the images locally.

( $13 \times 13$  pixel kernel) (3) morphological closing (disk-shaped structuring element of radius 5) to smooth the segmentation and bridge adjacent disconnected regions, and (4) filtering out connected components smaller than 100 pixels to reduce noise. To optimize performance, we performed a grid search on the edge detection threshold (Figure 2) and found a maximum pixel-wise accuracy of 0.8263 at a threshold of 0.0016 using our manual annotations as ground truth. We chose to use pixel-wise accuracy as an approximate measure of algorithm performance because throughout our testing, it seemed to align well by eye with our true measure of performance—suitability for downstream cellular segmentation—which is a difficult to quantify combination of the “cleanliness” of the output with how simple the task of programatically correcting misclassified pixels would be.

#### IV. CNN-BASED SEGMENTATION

Having established the performance of the current state-of-the-art techniques on our data set, we now will attempt to train a convolutional neural network to identify the cellular membranes in our images.

##### A. Existing Methods

Needless to say, the last few years have led to an explosion of papers applying deep convolutional nets to a veritable zoo of image processing problems. The problem of “semantic segmentation” of “real world” images has seen a considerable number of contributions, including [3, 4], among others. Most of the nets in these papers take an entire image as input and output a corresponding image of pixel labels, learning via a multinomial loss across pixels. Such techniques have recently been applied by the Covert lab here at Stanford to advance the state of the art in segmentation of heterogenous culture, by training a deep convolutional net to label individual pixels as “in a cell” or “not in a cell”.

This works like a dream for isolated cells. However, for the case of confluent culture, every pixel is “in a cell”, so such a classifier does not add any information to the segmentation problem.

Therefore, we decided instead to treat our problem as one of edge detection instead. This means we’re once again performing semantic segmentation, but with only two classes: “is an edge” and “is not an edge”.

Significant work has also been done on edge detection [5, 6]. Although this work applies directly to our problem, these previous papers have focused on attempting to reproduce human labels on heterogenous, “real world” images. Our application domain is much more narrow, and so both these networks, and the previously cited deep segmentation networks seemed like overkill for our problem. What’s more, they have been designed to learn from a massive corpus of labeled images, which we do not have access to in our case, since we must painstakingly handlabel our own images. Thus, we will take an altogether “new” approach by going with the most simple convolution network architecture possible, and training with a totally different input, output, and loss setup.

## B. New Method

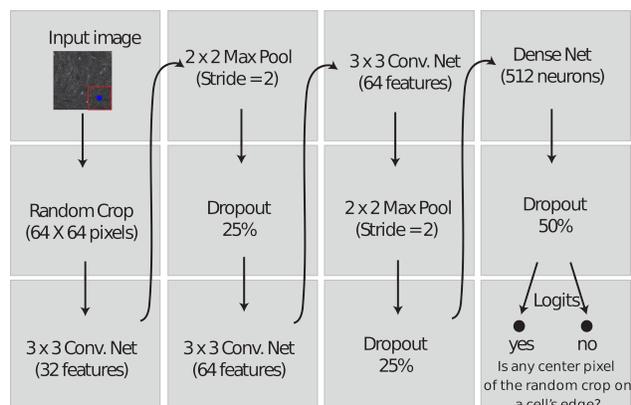


FIG. 3: The architecture of the neural network we used.

using random subsections of the image to train on, we transform the difficult problem of training a deep net with only eight example images into the easy problem of training a sort-of-deep net with  $num\_images \times 2160 \times 2160$  example pixels.

We constructed our neural network in Keras. Weights were initialized randomly using Keras’s defaults. The network was trained via the stochastic gradient descent with momentum using the ADAM [7] optimizer and the default Keras parameters of  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 1 \times 10^{-08}$ . The learning rate was set to 0.1 and the momentum to 0.9 following the example of a CIFAR-10 Keras tutorial that uses a similar network to ours. 1000 epochs of training, 5120 input pixels per epoch, on an NVIDIA GTX 680 in my laptop required mere hours. No hyperparameter tuning was required.

## C. Results

We initially used 31-by-31 pixel grids for classification, had no dropout layers, and did not exclude any pixels from the training set. This initial network quickly learned to simply classify everything as “not membrane”, since that gave it a reasonable accuracy of 70% percent. Moving to larger grids helped our training error, but not our test error. To combat overfitting, we added the dropout layers in Figure 3.

This modified network attained 95% test accuracy using just one training image (of 2160-by-2160 pixels, as in Figure 1), and one test image of the same size, in 1000 epochs of 5120 pixels each. However, its output was still unsuitable for downstream segmentation, since this accuracy was achieved by classifying hard-to-classify sections of the image as uniformly “not membrane”, and doing well elsewhere. This effect was always present within a dozen epochs of training. The center panel of Figure 4 shows what this looked like.

Looking through the sections that had been classified as uniformly “not membrane”, we hypothesized that the network had in fact learned identify regions of the image where cells were crawling on top of each other, and had decided that in those regions, it is more efficient to simply blanket classify as “not membrane”. Looking back through our hand-labeled data,

In order to use our limited data as efficiently as possible, we train on random 64-by-64 pixels crops of our input images (the red box in Figure 3). At each pass through the network, we ask our algorithm to classify that grid as “membrane” or “not membrane” depending on whether the center pixel of the grid (the small blue dot in Figure 3) is contained in the membrane of one of the cells or not (i.e. is labeled red in Figure 1).

Since we have not shown that we need a more complicated network (and it will turn out that we do not!), we use the unimaginative architecture in Figure 3. By pairing this simple architecture with our strategy of

Confusion Matrix at Epoch 21

	Predicted Positive	Predicted Negative
True Positive	0.81	0.04
True Negative	0.03	0.12

Validation Accuracy by Epoch

Epoch (#)	0	1	19	20	21
Accuracy (%)	76.5%	81.4%	93.0%	92.6%	93.1%

TABLE I: Although my GPU broke during the first training run of the full network, the network almost immediately attained 92% accuracy, the critical point after which it takes many epochs to make further progress.

we realized that in those regions, we could not agree even by eye on where the cell membrane actual was, making the labels in these sections basically random.

To test our hypothesis that these sections of bad labels were causing the artifact that we observed, we manually went back through the images and excluded the sections where the cells were not in a monolayer. This mask corresponds to the blue pixels in Figure 1.

It is important to note that this did not take away from the potential of the network to classify pixels of interest to us. Even if we had been able to identify membrane pixels in sections of image where the cells were on top of each other, we would not have been able to use this information to uniquely determine the outlines of each cell due to the overlap and their amorphous shapes, even by eye.

Surprisingly, this training set mask immediately eliminated the overlapping cells artifact. The third panel in Figure 4 shows the segmentation performance after just 21 epochs. Table I summarizes the learning curve and final confusion matrix.

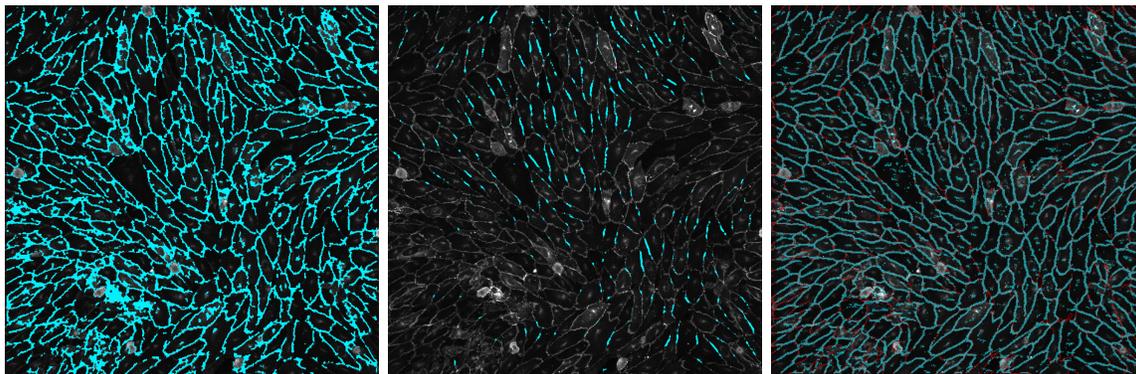


FIG. 4: LEFT: Results of best threshold-based segmentation attempt. CENTER: Test results after training for 1000 epochs with no mask to exclude certain pixels from training. RIGHT: Results after just 20 epochs of training the network and excluding “impossible to classify” pixels.

While the trial network above has already given us significant accuracy gains, we will also fine-tune the pretrained VGG-16 network included in Keras to see if we can improve our accuracy even further. Moving forward, we are confident that the CNN-based segmentation approach will allow us to more robustly segment the cells. At the end of the day, this is the only metric that matters.

---

[1] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, arXiv:1412.7062 [cs] (2014), arXiv: 1412.7062, URL <http://arxiv.org/abs/1412.7062>.

- [2] A. Paintdakhi, B. Parry, M. Campos, I. Irnov, J. Elf, I. Surovtsev, and C. Jacobs-Wagner, *Molecular microbiology* **99**, 767 (2016), URL <http://onlinelibrary.wiley.com/doi/10.1111/mmi.13264/full>.
- [3] G. Lin, C. Shen, I. Reid, and others, arXiv preprint arXiv:1504.01013 (2015), URL <http://arxiv.org/abs/1504.01013>.
- [4] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang, in *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 1377–1385, URL [http://www.cv-foundation.org/openaccess/content\\_iccv\\_2015/html/Liu\\_Semantic\\_Image\\_Segmentation\\_ICCV\\_2015\\_paper.html](http://www.cv-foundation.org/openaccess/content_iccv_2015/html/Liu_Semantic_Image_Segmentation_ICCV_2015_paper.html).
- [5] S. Xie and Z. Tu, in *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 1395–1403, URL [http://www.cv-foundation.org/openaccess/content\\_iccv\\_2015/html/Xie\\_Holistically-Nested\\_Edge\\_Detection\\_ICCV\\_2015\\_paper.html](http://www.cv-foundation.org/openaccess/content_iccv_2015/html/Xie_Holistically-Nested_Edge_Detection_ICCV_2015_paper.html).
- [6] G. Bertasius, J. Shi, and L. Torresani, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 4380–4389, URL [http://www.cv-foundation.org/openaccess/content\\_cvpr\\_2015/html/Bertasius\\_DeepEdge\\_A\\_Multi-Scale\\_2015\\_CVPR\\_paper.html](http://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Bertasius_DeepEdge_A_Multi-Scale_2015_CVPR_paper.html).
- [7] D. Kingma and J. Ba, arXiv preprint arXiv:1412.6980 (2014), URL <http://arxiv.org/abs/1412.6980>.