# Microscopy Cell Segmentation via Adversarial Neural Networks

Assaf Arbelle, Tammy Riklin Raviv

Ben Gurion University of the Negev

**Abstract.** We present a novel approach for the segmentation of microscopy images. This method utilizes recent development in the field of Deep Artificial Neural Networks in general and specifically the advances in Generative Adversarial Neural Networks (GAN). We propose a pair of two competitive networks which are trained simultaneously and together define a min-max game resulting in an accurate segmentation of a given image. The system is an expansion of the well know GAN model to conditional probabilities given an input image. This approach has two main strengths as it is weakly supervised, i.e. can be easily trained on a limited amount of data, and does not require a definition of a loss function for the optimization. Promising results are presented. The code is freely available at: https://github.com/arbellea/DeepCellSeg.git

## 1 Introduction

Live cell microscopy imaging is a key component in the biological research process. However, without the proper analysis tools, the raw images are a diamond in the rough. One must obtain the segmentation of the raw images defining the individual cells. Only then can a proper quantitative analysis of the cells' properties be calculated. Manual segmentation is infeasible due to the large quantity of images and cells per image.

Automatic segmentation tools are available and roughly split into two groups, supervised and unsupervised methods. The methods vary from automatic gray level thresholding as is done in [10] to more spatially aware approaches such as the watershed algorithm [25]. Other methods [3,14] use algorithms based on Active Contours (AC) [4] which aim to minimize an energy functional and obtain the most likely segmentation from a global perspective. Another approach is to support the segmentation algorithm with temporal information from tracking algorithms as was proposed by [21,2,1] All these methods assume some structure in the data that may not fit every case.

Supervised methods, on the other hand, do not assume any structure rather try to learn it. Classic machine learning methods generally require two independent steps, feature extraction and classification. In most cases the feature extraction is based either on prior knowledge of the image properties such as in [24] or general image properties such as smoothing filters, edge filters, etc. Many classifiers, which assign a label to a given feature, have been developed.

A widely used toolbox which takes a pixel classification approach is Ilastik [23], using a random forest classifier trained on predefined features extracted from a user's scribbles on the image.

Recent developments in the computer vision community have shown the strength Convolutional Neural Networks (CNN) as end-to-end classifiers, where no predefined feature calculation is needed. CNNs have shown to surpass previous state of the art methods in every aspect of computer vision ranging from object classification [7,22,12], object detection [18,19], semantic segmentation [13,16] and many other tasks. Recent attempts at cell segmentation using CNNs include [20,11]. The common ground of all the CNN methods is the need for an extensive training set along side predefined loss function, be it the cross-entropy loss for the classification task, intersection-over-union for the detection task or $L_1$ distance for the segmentation loss.

In this work we present a novel approach for microscopy cell segmentation using adversarial networks. We base our algorithm on the concepts introduced by [6] in their work on Generative Adversarial Networks (GAN) and extension thereof [17,15,8]. The GAN framework is based on two networks, a generator and a discriminator, trained simultaneously, with opposing objectives. This allows the discriminator to act as an abstract loss function in contrast to the common $L_1$ and $L_2$ loss. We a pair of adversarial networks, an **estimator** and a discriminator for the task of microscopy cell segmentation. Unlike the GAN we do not generate images from random noise vectors, rather estimate the underlaying variables of an image. The estimator learns to output some segmentation of the image while the discriminator learns to distinguish between manual segmentations and estimated segmentations given the associated image. The discriminator is trained to *minimize* a classification loss on the two classes, manual and estimated, i.e. minimizing the similarity between the two. The estimator, on the other hand, is trained to *maximize* the discriminator's loss and effectively, maximize the similarity. The closest work to ours, to the best of our knowledge is [9], generating images in a certain domain given images in a different domain. However, the authors show that their method falls short in the task of object segmentation and conclude the the $L_1$ loss is more fit for the task. We believe, however that the essence lies in the desire to **generate** (using a random noise vector) rather than **estimate** the segmentation image. We show in Section 3 the results for a non adversarial loss compared to the proposed method.

Our contribution is three-fold. We expand the concepts of the GAN for the task of cell segmentation and in that reduce the dependency on a selection of loss function. We propose a novel architecture for the discriminator, referred to as the "rib cage" architecture (See section 2.4), which is adapted to the problem. The "rib cage" architecture includes several cross connections between the image and the segmentation, allowing the network to model complicated correlation between the two. Furthermore we show that accurate segmentations can be achieved with minimal number of training examples which can dramatically reduce the manual workload needed for training.

Section 2 defines the problem and elaborates on the proposed solution. Section 3 shows promising initial results and Section 4 summarizes and concludes the work thus far.

## 2 Methods

### 2.1 Problem Formulation

Let $\Omega$ define the image domain and let the image $I : \Omega \to \mathbb{R}^+$ be an example generated by the random variable $\mathcal{I}$. Our objective is to partition the image into individual cells, where the main difficulty is separating adjacent cells. Let the segmentation image $\Gamma : \Omega \to \{0, 1, 2\}$ be a partitioning of $\Omega$ to three disjoint sets, background, foreground (cell nuclei) and cell contour, also generated by some random variable $\mathcal{S}$. The two random variables are statistically dependent with some unknown joint probability $P_{\mathcal{I},\mathcal{S}}$. The problem we address can be formulated as the most likely partitioning $\hat{\Gamma}$ from the data $I$ given only a small number, $N$, of example pairs $\{I_n, \Gamma_n\}_{n=1}^N$. Had the probability function $P_{\mathcal{S}|\mathcal{I}}(\Gamma|I)$ been known, the optimal estimator would be the Maximum Likelihood (ML) estimator:

$$\hat{\Gamma}_{opt} = \arg \max_{\Gamma} P_{\mathcal{S}|\mathcal{I}}(\Gamma|I) \tag{1}$$

However, since $P_{\mathcal{S}|\mathcal{I}}(\Gamma|I)$ is unknown and $\hat{\Gamma}_{opt}$ cannot be calculated, we learn the near-optimal estimator of $\Gamma$ using the manual segmentation, $\Gamma_M$, as our target.

### 2.2 Estimation Network

We propose an estimator $\hat{\Gamma} = \mathcal{E}\left(I, \hat{\theta}_{\mathcal{E}}\right)$ in the form of a CNN with parameters $\theta_{\mathcal{E}}$. We wish to train the estimator $\mathcal{E}$ such that the estimated $\hat{\Gamma}$ will be as close as possible to the optimal ML estimation $\hat{\Gamma}_{opt}$. This is achieved by solving:

$$\hat{\theta}_{\mathcal{E}} = \arg \min_{\theta_{\mathcal{E}}} L_{\mathcal{E}}\left(\mathcal{E}(I, \theta_{\mathcal{E}}), \hat{\Gamma}_{opt}\right) \tag{2}$$

The loss function $L_{\mathcal{E}}$ will be defined in section 2.3.

### 2.3 Adversarial Networks

Unlike the GAN, aiming to *generate examples* from an unknown distribution , we aim to *estimate the variables* of an unknown conditional distribution $P_{\mathcal{S}|\mathcal{I}}(\Gamma|I)$. Defining the loss $L_{\mathcal{E}}$ either in a supervised pixel-based way, e.g. $L_2$ norm, or in an unsupervised global method, e.g. Active Contour energy functional, is usually not well defined. We define the loss $L_{\mathcal{E}}$ by pairing our estimator with a discriminator. Let $\mathcal{E}_{\theta_{\mathcal{E}}}$ and $\mathcal{D}_{\theta_{\mathcal{D}}}$ denote the estimator and discriminator respectively, both implemented as CNN with parameters $\theta_{\mathcal{E}}$ and $\theta_{\mathcal{D}}$ respectively. The estimator aims to find the best estimation $\hat{\Gamma}$ of the partitioning $\Gamma$ given the image

$I$. The discriminator on the other hand tries to distinguish between $\Gamma_M$ and $\hat{\Gamma}$ given pairs of either $(I, \Gamma_M)$ or $\left(I, \hat{\Gamma}\right)$ and outputs the probability that the input is manual rather than estimated denoted as $D\left(I, \hat{\Gamma}\right)$. As is in the GAN case, the objectives of the estimator and the discriminator are exactly opposing and so are the losses for training $\mathcal{E}_{\theta_\mathcal{E}}$ and $\mathcal{D}_{\theta_\mathcal{D}}$. We train $\mathcal{D}_{\theta_\mathcal{D}}$ to *maximize* the probability of assigning the correct label to both manual examples and examples estimated by $\mathcal{E}_{\theta_\mathcal{E}}$. We simultaneously train $\mathcal{E}_{\theta_\mathcal{E}}$ to *minimize* the same probability, essentially trying to make $\hat{\Gamma}$ and $\Gamma_M$ as similar as possible:

$$L_\mathcal{D} = \mathbb{E}\left[\log\left(D\left(I, \Gamma_M\right)\right) + \log\left(1 - D\left(I, \mathcal{E}_{\theta_\mathcal{E}}\left(I\right)\right)\right)\right] \tag{3}$$

$$L_\mathcal{E} = \mathbb{E}\left[\log\left(D\left(I, \mathcal{E}_{\theta_\mathcal{E}}\left(I\right)\right)\right)\right] \tag{4}$$

In other words, $\mathcal{E}_{\theta_\mathcal{E}}$ and $\mathcal{D}_{\theta_\mathcal{D}}$ are players in a mini-max game with the value function:

$$\min_{\mathcal{E}_{\theta_\mathcal{E}}} \max_{\mathcal{D}_{\theta_\mathcal{D}}} \mathbb{E}\left[\log\left(D\left(I, \Gamma_M\right)\right) + \log\left(1 - D\left(I, \mathcal{E}_{\theta_\mathcal{E}}\left(I\right)\right)\right)\right] \tag{5}$$

The equilibrium is achieved when $\hat{\Gamma}$ and $\Gamma_M$ are similar such that the discriminator can not distinguish between the pairs $\left(I, \hat{\Gamma}\right)$ and $(I, \Gamma_M)$.

## 2.4   Implementation Details

**Network Architecture**   The estimator $\mathcal{E}_{\theta_\mathcal{E}}$ net is designed as a five layer fully CNN, each layer constructed of a convolution followed by batch normalization and leaky-ReLU activation. The output of the estimator is an image with the same size as the input image with three channels corresponding to the probability that a pixel belongs to the background (red), foreground (green) or cell contour (blue).

The discriminator $\mathcal{D}_{\theta_\mathcal{D}}$ is designed with a more complex structure. We refer to the basic block as a "rib cage". Each "rib cage" has three inputs and three outputs referred to as "left rib", "right rib" and "spine". The output of the left and right are ribs are blocks of convolution followed by batch normalization and leaky-ReLU activation on the left and right inputs respectively. The output of the "spine" however is a mixture of the two ribs and is designed as a concatenation of all three inputs followed by convolution, batch normalization and leaky-ReLU activation. See Figure 1 for an illustration of the "Rib Cage" block. The discriminator is then designed as three consecutive "rib cage" followed by two fully-connected layers with leaky-ReLU activations and a final fully connected layer with one output and a sigmoid activation for classification. Figure 2 illustrates the discriminator design. The architecture parameters for the convolution layers are describes as $C\left(kernel-size, \#filters\right)$ and fully connected layers as $F\left(\#filters\right)$. The parameters for the estimator: $C\left(9, 16\right), C\left(7, 32\right), C\left(5, 64\right), C\left(4, 64\right), C\left(1, 3\right)$. The discriminator center rib used half the number of filters as the left and right: $C\left(9, 8\right), C\left(5, 32\right), C\left(3, 64\right), C\left(4, 64\right), F\left(64\right), F\left(64\right), F\left(1\right)$
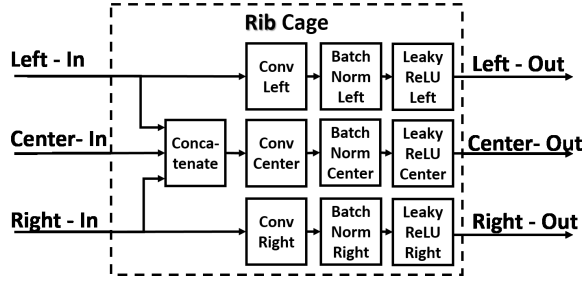
**Fig. 1.** The design of the basic building block for the discriminator. Each block has three inputs and three outputs: Left, Center and Right
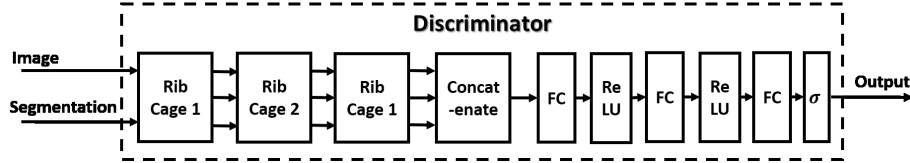


**Fig. 2.** The design of the Discriminator $\mathcal{D}_{\theta_{\mathcal{D}}}$. Each "Rib Cage" block is explained in Figure 1. Three "Rib Cage" blocks are followed by two fully connected (FC) layers with ReLU activations and a last FC layer with a sigmoid activation, $\sigma$, for classification.

**Training Regime** We trained the networks for ~50K iterations with an RMS Prop optimizer as implemented in Tensorflow with learning rate to 0.0001. We trained the estimator with a 1:2 ratio (10 estimator iterations for every 20 discriminator iterations). The batch size was set to 100 for the estimator and 200 for the discriminator (100 estimated and 100 real examples). The training ran on a CPU, Intel Xeon E5 2680.

**Data** We trained the networks on the H1299 data set [5] consisting of 72 frames of size $512x640$ pixels. Each frame captures approximately 50 cells. Manual annotation of 15 randomly selected frames was done by an expert. The annotated set was split into a training set and validation set. From the training set we selected $N_{Train} \in [1, 2, 4, 11]$ examples for training. The training set was augmented using randomly cropped areas of the input images of size $64x64$ along with random flip and random rotation. The images where annotated using three labels for the background, cell nucleus and nucleus contour. These annotations where then encoded as RGB images, red indicating the background, green indicating the cell nucleus and blue indicating the contour.

## 3   Experiments and Results

We conducted four experiments, training the networks with different values for $N_{Train}$. All other parameters where set identically. We evaluated the segmen-
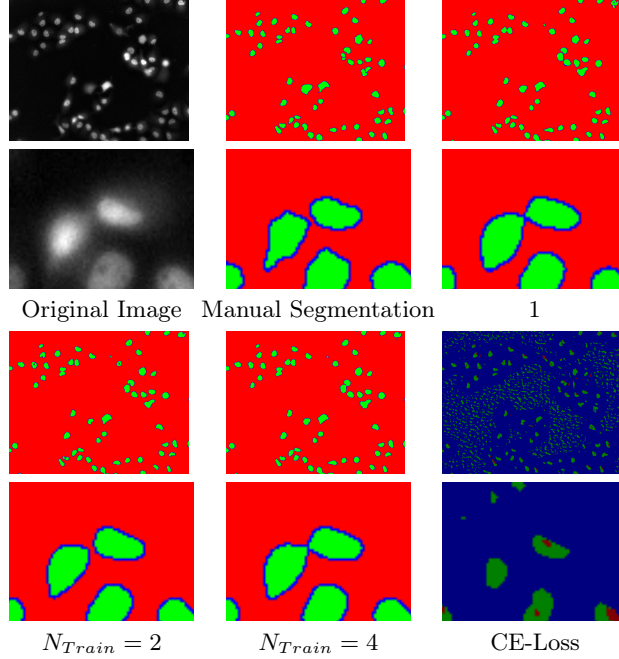
**Fig. 3.** An example for the segmentation of a validation image given a different number of training examples. The odd rows show the full image and the even rows show a zoom on a specific area. Notice that in all cases the cells where correctly separated even though they are very close together. The bottom right pair shows the result for the estimator trained only with the cross entropy loss, without using a discriminator.

| Experiment | ADV-1 | ADV-2 | ADV-4 | ADV-11 | CE Loss - 11 | Ilastik |
|---|---|---|---|---|---|---|
| Precision | **89.9%** | 85.4% | 86.8% | 85.8% | 2.43% | 88.5 % |
| Recall | **82%** | **87.2%** | **86.8%** | **86.5%** | 67.11% | 82.4% |
| F-Measure | **85.8%** | **86.3%** | **86.8%** | **86.1%** | 4.7% | 85.3% |

**Table 1. Quantitative Results:** Each column represents an experiment with a different number of training examples, ADV-$N_{Train}$. CE Loss- 11 is an experiment of the same estimator network trained only with the pixel-based cross entropy loss and without the adversarial training regime. The last column is the comparison to the state of the art tool, Ilastik [23]. The rows are the results for individual cell segmentation. As is explained in [21] True positives (TP) are cells with Jaccard measure greater than 0.5. False positives (FP) are automatic segmentation not appearing in the manual segmentation and false negatives (FN) is the opposite. The measures are defines as $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$, $F - Measure = 2\frac{Precision*Recall}{Precision+Recall}$

tation using the common method as is described in [21]. We compared the adversarial training regime to the common cross entropy loss, training only the estimator without a discriminator. In this experiment the estimator architecture was identical to the adversarial experiments and all training examples were used. We also compared our results to state of the art semi-manual segmentation tool, Ilastik [23]. For the comparison, an expert examined the full data set (72 Frames) and added extensive annotations with focus on difficult scenarios. The quantitative results of the individual cell segmentation are detailed in Table1. Note that the amount of images in the training data did not have major effect on the results and with only two training examples we surpass the state of the art tool. Figure 3 shows an example of a segmented frame. It is clear that the networks learned a few distinct properties of the segmentation. First, each cell (in green) is encircled by a blue line. Second, The shape of the contour follows the correct shape of the cell. Some drawbacks are still seen especially in cases when two cells completely touch and the boundary is difficult to determine. However,

## 4    Summary

In this work we propose a new concept for microscopy cell segmentation using CNN with adversarial loss. The contribution of such an approach is two-fold. First, the loss function does not need to be defined as it is learned along side the estimator, making this a simple to use algorithm with no tuning necessary. Second, we show that this method is very robust to low number of training examples surpassing the state of the art with only a single annotated frame.

The quantitative results, as well as the visual results, show clearly that both the estimator and our unique "rib cage" discriminator learn both global and local properties of the segmentation, i.e the shape of the cell and the contour surrounding the cell, and the fitting of segmentation edges to cell edges. These properties could not be learned using only a pixel-bases cross entropy loss as is commonly done.

Future work will aim at addressing the major bottlenecks of acquiring manual annotations and training separate networks, by utilizing a pre-trained network on a different data set.

## References

1. F. Amat, W. Lemon, D. P. Mossing, K. McDole, Y. Wan, K. Branson, E. W. Myers, and P. J. Keller. Fast, accurate reconstruction of cell lineages from large-scale fluorescence microscopy data. *Nature methods*, 2014.
2. A. Arbelle, N. Drayman, M. Bray, U. Alon, and T. Carpenter, Anne anr Riklin-Raviv. Analysis of high throughput micrscopy videos: Catching up with cell dynamics. In *MICCAI 2015*, pages 218–225. Springer, 2015.
3. P. Bamford and B. Lovell. Unsupervised cell nucleus segmentation with active contours. *Signal Processing*, 71(2):203–213, 1998.
4. T. F. Chan and L. A. Vese. Active contours without edges. *Image processing, IEEE transactions on*, 10(2):266–277, 2001.

5. A. A. Cohen, N. Geva-Zatorsky, E. Eden, M. Frenkel-Morgenstern, I. Issaeva, A. Sigal, R. Milo, C. Cohen-Saidon, Y. Liron, Z. Kam, et al. Dynamic proteomics of individual cancer cells in response to a drug. *science*, 322(5907):1511–1516, 2008.

6. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

7. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of CVPR*, pages 770–778, 2016.

8. D. J. Im, C. D. Kim, H. Jiang, and R. Memisevic. Generating images with recurrent adversarial networks. *arXiv preprint arXiv:1602.05110*, 2016.

9. P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.

10. T. Kanade, Z. Yin, R. Bise, S. Huh, S. Eom, M. F. Sandbothe, and M. Chen. Cell image analysis: Algorithms, system and applications. In *WACV*, pages 374–381. IEEE, 2011.

11. O. Z. Kraus, J. L. Ba, and B. J. Frey. Classifying and segmenting microscopy images with deep multiple instance learning. *Bioinformatics*, 32(12):i52–i59, 2016.

12. A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

13. J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE CVPR*, pages 3431–3440, 2015.

14. E. Meijering, O. Dzyubachyk, I. Smal, et al. Methods for cell and particle tracking. *Methods Enzymol*, 504(9):183–200, 2012.

15. M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

16. H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *ICCV*, December 2015.

17. A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

18. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of CVPR*, pages 779–788, 2016.

19. S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

20. O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *arXiv preprint arXiv:1505.04597*, 2015.

21. M. Schiegg, P. Hanslovsky, C. Haubold, U. Koethe, L. Hufnagel, and F. A. Hamprecht. Graphical model for joint segmentation and tracking of multiple dividing cells. *Bioinformatics*, page btu764, 2014.

22. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

23. C. Sommer, C. Straehle, U. Kothe, and F. Hamprecht. Ilastik: Interactive learning and segmentation toolkit. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 230–233, March 2011.

24. H. Su, Z. Yin, S. Huh, and T. Kanade. Cell segmentation in phase contrast microscopy images via semi-supervised classification over optics-related features. *Medical image analysis*, 17(7):746–765, 2013.

25. L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE PAMI*, 13(6):583–598, 1991.