Nuclei detection and segmentation in divergent images

Farnoush Yousefi Kejani

University of Waterloo, Waterloo, Canada, fyousefi@uwaterloo.ca

Abstract. The ability to automatically detect and segment cells and tissues has been an active research field in medical informatics for decades, but it has recently attracted increased attention due to the developments in computer and microscopy hardware. Cell detection methods have evolved from employing hand-crafted features to deep learning-based techniques. In cancer research, these approaches play a key role in minimizing human intervention and providing traceable clinical information. The purpose of this report is to detect and segment nuclei, using machine learning algorithms. First, three different architectures (namely U-NET, CNN with no pooling layer and pixel-based CNN) of deep neural networks, for the medical image segmentation, are presented, and then their performances are evaluated on Kaggle challenging database. Second, to improve the segmentation accuracy by these methods, some new strategies are proposed in pre-processing and post processing steps (specially, separating connected nuclei). Experimental results and our score on Kaggle website shows the improvement of the segmentation accuracy with respect to original networks.

Keywords: Nuclei detection · Nuclei segmentation · Deep networks · Convolutional neural networks.

1 Introduction

In order to cure all diseases, we need to find effective new medicines. In this regard, spotting nuclei will help us enhancing drug development, advancing medical treatment and improving quality of life. Like Moore's in the technology industry, we have Eroom's law in the pharmaceutical industry, which is showing the number of new medicines per $1 billion of dollars of research and development.

Scientists should examine affect of thousands of new chemicals on cells, in order to find the most effective way to cure each specific illness. Currently, time consuming parts such as making cell patches, adding chemicals to them and capturing the images are done by robots. But analyzing these images are done by a human, as it is still a challenging task for computers. Scientists and researchers main goal for this challenge is to develop a robust automated method that is capable of analyzing vast range microscopic images, without human interference. One of the main approaches in analyzing microscopic images is to spot cell's

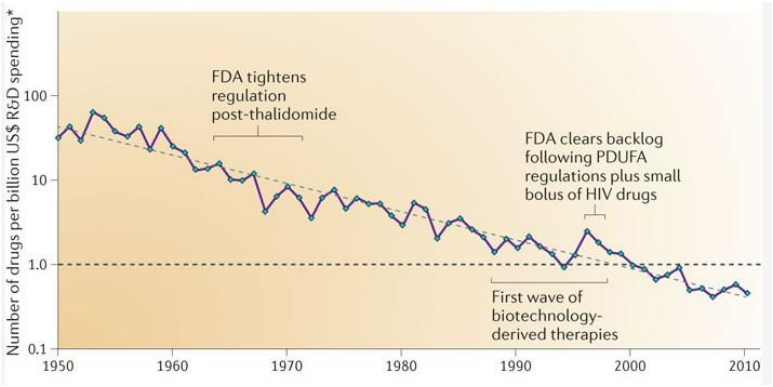

Fig. 1. Number of drugs per $1 billion R&D spending.

nucleus. As a result, developing a robust approach for nuclei extraction from the microscopic images, will enable us to approach a fully automatic system, which is going to not only reduce both the time and the cost of the research and development in the pharmaceutical industry, but also increase the accuracy and reliability of the our results.

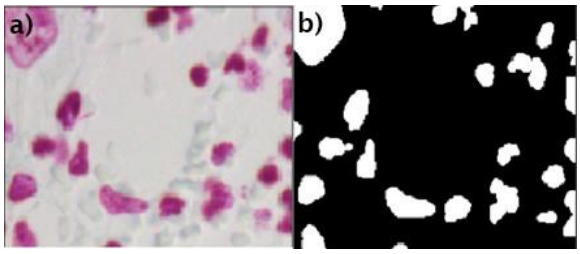

Fig. 2. (a) Sample microscopic image from the data-set, and (b) its corresponding mask. Our goal is to produce the binary masks for all input microscopic images.

For nuclei segmentation in microscopic images, there are two knowledge-based and data-driven approaches. knowledge-based approaches use a prior information on cell shape features to obtain the segmentation, which may introduce a bias favoring the detection of cell nuclei, only with a certain properties[1]. These approaches are based on pure Image Processing algorithms, relying on our knowledge from images. For an example, nuclei extraction can be done either with common methods such as simple thresholding, or more advanced image processing methods like Otsu's global thresholding and Adaptive local thresholding.

Large number of nuclei segmentation algorithms have been proposed in the literature. One of these approaches is based on finding a seed point within each

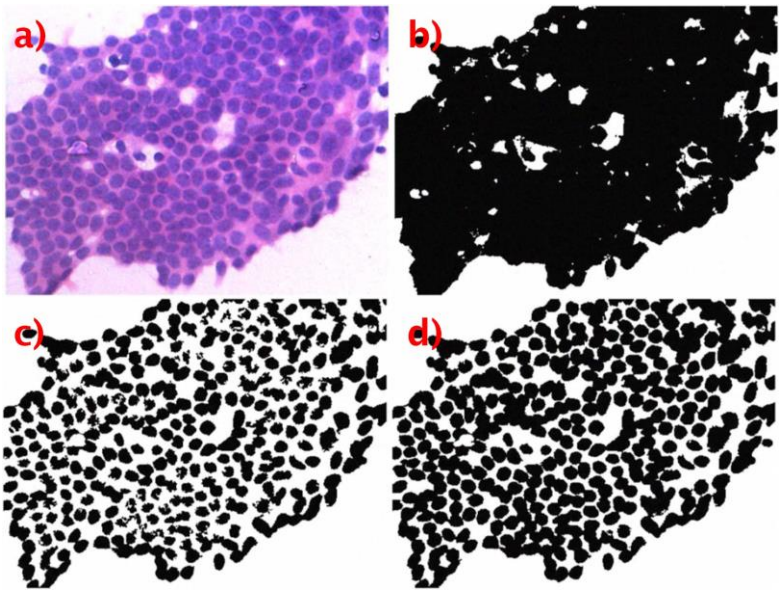

Fig. 3. (a) Photomicrograph of a cytology smear, (b) result of thresholding using Otsu's global thresholding method, (c) result of thresholding using Sauvola's method [6], (d) result of thresholding using the adaptive local thresholding[5].

nuclei region, and then deriving the boundary of the nuclei initializing at the seed point. In [2], application of the Hough transform technique for detecting nuclei seed points, is proposed, being used in initializing a shape- and texture-based active contour model. In [3], the nuclei seed points are found by the peaks of Euclidean distance map. Authors of [4], used an active contour to find the boundary of nuclei.

Unfortunately, these approaches(knowledge and model-based) are sensitive to the data, so that we can not achieve the same accuracy and outcome on different type of microscopic images. Also, improving results in this kind of approaches require having deep knowledge about the problem.

On the other hand, by the increase in the volume of our data, we can benefit from data-driven approaches, such as deep learning methods. These methods mainly focus on training machine learning models, in order to solve the problem by learning it through many examples. For instance, several researchers has worked on specialized deep neural network architectures, for medical image processing approaches. Jafari et al. [10] proposed pixel based approach for segmentation of the skin lesion pixels from the background. In other work, Ronneberger et al. [11], introduced new architecture of deep neural network, which is made of convolutional, max pooling and deconvolutional layers, for medical image processing purposes. Moreover, Kumar et al. [7], proposed a new deep learning architecture, for assigning 3 class probability to each pixel of the image. They defined 3, foreground, boundary and background classes, and they passed

a window around each pixel of interest into their network, in order to label each input pixel. Their network was made of both CNN and fully connected layers, in order to distinguish between boundary and non-boundary class. All of these methods are discussed in detail in the following sections of this paper.

In this paper, we mainly focus on CNN-based segmentation algorithms. First, three different deep learning architectures (namely U-NET, CNN with no pooling layer and pixel-based CNN), for the medical image segmentation, are presented. Then, their performances are evaluated on Kaggle challenging database. In addition, some new strategies are proposed in pre-processing and post-processing steps, inorder to tackle the challenges of this database. Experimental results and our score on Kaggle website shows the improvement of segmentation accuracy with respect to the original networks. The rest of the paper is as follow: In section 2, three CNN-based architecture are discussed in detail. Section 3, gives an explanation about the data-set and pre-processing steps, which is required before the training session. In section4, the initial results and the final results are discussed, and finally section5, summarizes the finding of this project.

2 CNN-based segmentation algorithms

When a large number training images are available in the database, the deep learning techniques, particularly CNNs, have shown the best performance and efficiency for the image classification[8]. CNNs are also being used on large medical images to produce probability maps for detecting various tissue segments, based on processing small and fixed size patches, sampled from the original large images.

We selected three different CNN-based deep network architectures to segment nuclei, namely common CNN with no fully connected layer, U-NET and pixel-based classification. In this section these three algorithms are described in detail and their components are presented.

2.1 Simple CNN with no pooling

The main advantage of CNN networks is that the input images can be plugged in directly to the input layer. So, it does not need any feature extraction. Also, the output can be in any form, matrix, vector or scalar. In our nucleus segmentation problem, there are three channel images, from which the nucleus should be detected and segmented. Therefore, the output will be a binary mask, with the same the size of the original image, where the white pixels show nucleus pixels and black pixels belong to the background. In a typical CNN, mostly after convolutional layer, a pooling layer is applied, in order to reduce or increase (up-convolution) the image size. In addition, since the output is an image, we do not have fully connected layer at the end. The designed CNN architecture is shown in the Table 1.

The output is going to be a $256 * 256$ matrix. Each element of this matrix shows the probability of being nucleus in the underlying pixel location. We

Table 1. Parameters of simple CNN with no pooling.

Layer	input	CONV1	CONV2	CONV3	CONV4	CONV4	output
filter size	-	3*3	3*3	5*5	5*5	7*7	-
activation	-	Relu	Relu	Relu	Relu	Relu	soft-max
outputsize	256*256*3	256*256*3	256*256*16	256*256*32	256*256*64	256*256*64	256*256
dropout	0.1	0.1	0.3	0.3	0.5	0.5	-

empirically determined that five convolutional layers were well suited for our problem. Adding more layers made the computation more expensive without an appreciable positive impact on the accuracy. Hyper-parameters like the size and the number of filters in each layer were selected in a way that the resulting architecture gives us the best performance on the validation samples. Rectified linear unit (ReLU) non-linearity is used for all hidden layers, which speeds up training by avoiding vanishing gradient problem. As it is seen in 1, the dropout rate is increased, while we are going deeper in the network. A high dropout rate in the initial layers results in information loss, but it acts as a good regularizer in the deeper layers and avoids the over-fitting[9].

2.2 Two class pixel segmentation by CNN

In this approach, instead of training the CNN with the full size images, and getting the binary mask at the output layer, we segment each pixel separately as background or nucleus. We implemented the architecture proposed in [10], in order to classify each pixel from the input image, into the three different classes; background, nucleus and boundary. For estimation of the three-class probability assignment for each pixel, they trained a three-output(see figure 4) node CNN that took a patch of size 31*31 centered at that pixel as input. The size of the patch is selected such that to cover the majority of the large nuclei (nuclei size varies from one pixel to 12000 pixels in the database).

We defined boundary class as a group of the pixels, being within a distance of 1 from the boundary, forming a ring around each nucleus. The output layer gives three values that show the probability of each class. Each pixel will eventually be labeled, either as the background or the nucleus (boundary pixels belong to background). Boundary pixels are included in the training data, in order to force the network to learn them, and segment the foreground pixels as accurate as possible. For each class, 100 windows of size 31* 31 are randomly selected from the train images for building the train data.

In [10], for each pixel, two different windows (or patches), namely the global with larger size and the local with smaller size, are considered. A small size window or patch reveals the local texture around the pixel, and it is going to be helpful for the segmentation purpose. On the other hand, the bigger window reveals the global feature. Experimentally, we only used the local patches around each pixel(the upper root in figure 4. The order of layers in the network is as *Conv1*, *MPool1*, *Conv2*, and *MPool2*. Kernel size in *Conv1* is $6 * 6 * 3$ and is

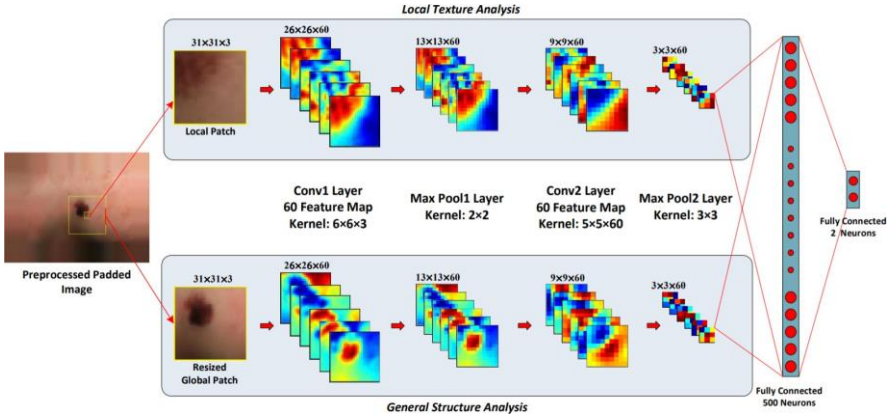

Fig. 4. Architecture of proposed network by Jafari et al [10].

5 × 5 × 60 in *Conv2*. The hyper parameters of the network, such as size of input patches and kernels and number of feature maps are set by test and trial. Outputs of the CNN network is fed into the two fully connected layers, with 1000 and 512 neurons respectively. Finally, this layer is followed by an output softmax layer with two neurons. The output of the network indicates a probability of membership of each pixel to the background or the nucleus class. During the testing phase, for each pixel, a window of size 31 × 31 centered at current pixel location is built and plugged in to our trained model. The output shows that if the current pixel is nucleus or background.

2.3 U-NET

In [11], a network and training methodology has been proposed that relies on the better use of data augmentation, for utilizing the available annotated samples more efficiently. The architecture consists of a contracting path to capture the context and a symmetric expanding path that makes the exact localization. The main idea of *U-NET* is to implement a usual contracting network by successive layers, where the authors used up-sampling operators instead of pooling operators. These layers increase the resolution of the output. In order to localize, high resolution features from the contracting path are combined with the up-sampled output.

The architecture of *U-NET* is illustrated in figure 5. It consists of a contracting path (left side with pooling), and an expansive path (right side with up-sampling). The contracting path follows the common architecture of a CNN, which consists of the repeated application of the two convolution layers with 3 × 3 kernels, each followed by a ReLU activation function and a 2 × 2 max pooling operation. At each down-sampling step, the number of features is doubled, in order to capture more information with the less time complexity. The opposite of all

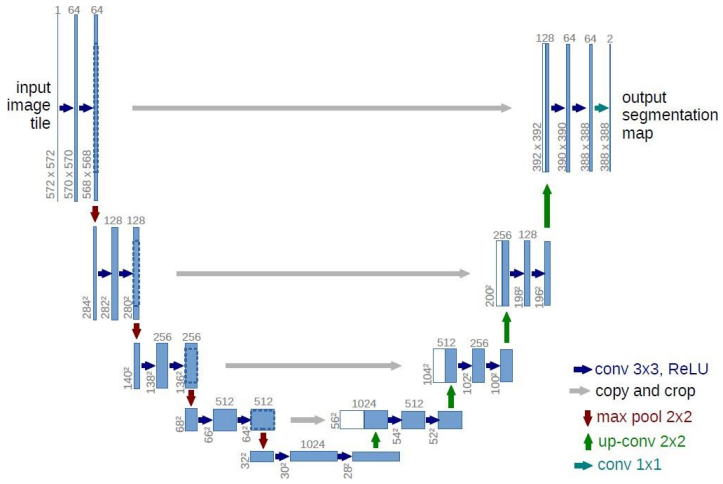

Fig. 5. Architecture of U-NET [11].

these operations are done on the other side of the network (at up-sampling). In total the network has 23 convolutional layers. We implemented the same structure with some changes. The input and output images are 256* 256, and also we added dropout rate for each layer, which is increased as network goes deeper. Similar to the first architecture (section 2.1), the network does not have any fully connected layers.

3 Dataset and Preprocessing

The dataset is from Kaggle competition and is publicly available [1]. It is made of about the 25,000 human annotated and segmented nuclei masks (it may contain some errors, as it was mentioned by the competition organizers), including vast range of the images, including both the easy and the hard images (sometimes it is hard to distinguish between single lumpy nucleus and two adjacent nuclei). This data-set contains a large number of segmented nuclei images, being acquired under a wide variety of conditions and vary in the cell type, magnification, and imaging modality (bright field vs fluorescence). The data-set is designed to challenge an algorithm's ability to generalize its solution, across these variations.

The data-set contains 670 three channel images, and each image contains a varying number of nuclei. The image intensity is in the [0, 255] range. Table 2 shows the statistics of the images and the masks.

In order to prepare images to be fed into network, we need to do some preprocessing to achieve the highest segmentation accuracy. The following operations were done to prepare the data-set for the network training.

[1] *https : //www.kaggle.com/c/data − science − bowl* − 2018

Table 2. Summery of dataset.

-	Image Height	Image Width	Image Ratio	# of masks per image	Nucleus size
mean	333.99	378.5	0.92	43.97	260.18
std	149.47	204.8	0.11	47.96	430.46
min	256	256	0.47	1	1
max	1040	1388	1	375	12064

1) All images should be re-sized in the same size. Since the most of the images in the database are 256 256, all images were re-sized to 256 256.

2) Histogram equalization and contrast enhancement were applied, in order to sharpen the images and produce stronger edges. This helps the network to find the borders of a nucleus easier. For applying contrast enhancement, the following operation is done:

$$y = (\frac{x}{255})^2 * 255, \ 0 \leq x \leq 255 \quad (1)$$

Where x is the pixel intensity in the original image and y is the enhanced intensity value.

3) One of the challenges for nuclei segmentation is a large variation in the image's colors. We applied color normalization as another pre-processing step, in order to improve nucleus segmentation by accounting for the variations in the staining and scanning processes[12].

4) Most of the nuclei in the database, are brighter than their background. However, there are some instances, in which the nucleus has smaller intensity value and it is darker respect to the background. These samples automatically were detected, and intensity inversion algorithm was applied to them, i.e.: $intensity = 255 - intensity$.

5)Data augmentation is essential, for teaching the network the desired invariance and robustness properties, when only a few training samples are available. For microscopical images, we primarily need shift and rotatation invariance, as well as the robustness to the deformations and the gray value variations. Random elastic deformations of the training samples seem to be essential to train a network with a small number of the annotated images. We generated smooth deformations using random displacement, rotation, zooming in and out and elastic deformation. These deformations were applied to the original 670 images and eventually we had 2100 images, with their corresponding masks to train the networks. It is worth to mention that number of colorful images are about one third of gray images. For balancing the number of the colorful and the gray images, we generated more number of colorful images.

4 Results and discussion

We trained and tested all three CNN architecture using 2100 pre-processed images and masks. 20 percent of these images were separated as the test data,

and the rest was used as train data. For *Two class pixel segmentation by CNN* architecture, 630000 windows of size 31 31 are randomly selected from train images, in order to build the train data for the network. 210000 windows were selected for each class (namely nucleus, background and boundary). 10 percent of training data were used for validation step. All three networks, written in Python, were tested on an Intel(R) Core(TM) 2 Quad Q9550 CPU at 2.83 GHz.

To evaluate the accuracy of each network the following criterion was used:

$$ACC = \frac{A \cap B}{A \cup B} \tag{2}$$

Where A is the true mask and B is the predicted mask. Therefore, the accuracy will be 100% if the intersection between A and B is equal to their union, i.e, if they are the same ($A = B$). Table 3 shows the *ACC* and the testing time per image on the test data.

Table 3. Results on test data.

-	ACC	Test time per image
Simple CNN	89%	less than one second
Pixel-based CNN	87%	above one minute
U-NET	92%	less than one second

U-NET outperforms the other two network, as it was expected. This architecture has been designed to segment medical images, and it had won the ISBI cell tracking challenge 2015 in, by a large margin.The time complexity of pixel-based CNN is much higher than the of other two architectures. In this architecture, we do the classification for each pixel separately. In other word, first we build a window for each pixel, and then we feed it to the network and get the result. Therefore, we need to run the network 256*256 times for each image. Hence, the time complexity will increase considerably.

As it was mentioned earlier, we used the data-set which is available for a Kaggle competition. For this competition, there are 67 test images which their mask (true answer) is not available. We run the trained U-NET on this test data and submitted it to Kaggle website. But the result was much worse than we were expecting. In next section(4.1) we will discuss the challenges and reasons for this low score (around 20% accuracy achieved).

4.1 Post processing

Nuclei were detected easily by thresholding the inside class probability map at 0.5. However, the resulted mask does not completely look like the true mask, and it needs some post processing, like morphological operation, to enhance the predicted mask. In this section we will discuss challenges that caused decrease in the network's accuracy. Eventually, we came up with some new ideas for addressing these issues.

Connected Nucleus: The accuracy metric used by Kaggle website, was far different from what we used as (ACC). To evaluate the performance of the network, the competition organizers proposed the following procedure. First, they consider a vector of thresholds t, starting from 0.5 to 0.95 with a step size of 0.05. At each threshold t, they computed ACC metric over each nucleus (object) pair (in the true mask and predicted mask). The predicted object is considered a "hit", if its intersection over union with a ground truth object is greater than t. At each threshold value t, a precision value is calculated based on the number of true positives (TP), false negatives (FN), and false positives (FP), resulting from comparing the predicted object to all ground truth objects:

$$PR = \frac{TP}{TP + FP + FN} \tag{3}$$

The average precision of a single image is then calculated as the mean of the above precision values at each threshold. This metric significantly penalizes connected nuclei in the predicted mask. Fore example, a sample true mask and its predicted result are shown in figure 6.

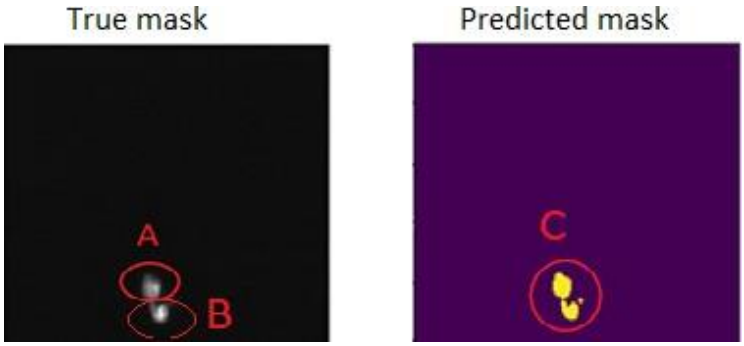

Fig. 6. Connected objects.

Although the ACC metric for these two mask is above 90%, the PR value is zero for them. In the predicted mask we only have one object(C) (two objects are connected). The ACC between A and B or C and B is always less than 0.5 which is minimum threshold. Therefore, PR will be zero. Hence, we need to separate the connected objects.

To separate these connected objects, we proposed two solutions. The first one is to train another U-NET which is only capable of detecting the boundary of a nucleus. To do so, we need to prepare new binary masks that are one in boundary location and zero in other pixel locations. We trained another U-net with the same structure described in previous sections, that only emphasizes on borders of nuclei. The output of this network is combined with the previous one to show a better accuracy.

The second method we came up with, is to do post-processing on predicted mask. In other word, we detect the connected objects and try to separate them at the best points. Intuitively it can be said that the pixels on the contour of a nucleus build a convex hull, i.e, the curvature of these points is toward the outside of object.

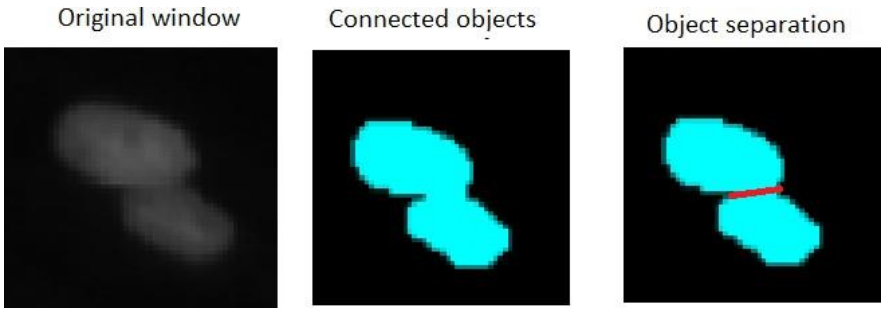

Fig. 7. Connected objects separating and convex hull.

Figure 7 shows the two connected objects. If two objects are connected, the points on the contour of the connected object is not convex hull anymore. To separate these objects, first we need to find the contour of each connected component and build the convex hull of these points. Then we check to see whether there are some points which are far from the convex hull border or not. If so, we find two non-neighbor points (on the contour points) which have the closest point to each other. So, we separate the object by the line crossing these two points (see Figure 7). To improve the original U-NET results, we combined the results of these two approaches, in order to separate the connected nuclei as much as possible.

Images with two different boundaries: There are some images in the data-set that have two distinct backgrounds. The first one is the global background which has the most dark gray scale, and the second one is local background (Cell's cytoplasm). The local background pixels, are those pixels around nuclei, which are brighter than the global background. All of trained three networks, do not work well on these images. Figure 8, shows a sample image with two background, which results in two kinds of edges and borders, beside the output of the U-NET network.

To tackle this problem, we used adaptive local thresholding method. For each local window, we first compute the histogram. In windows that both global and local backgrounds exist, around the nucleus location, the histogram will have two peaks. These two peaks show the most repetitive gray level in the window. Therefore, simply by removing all pixels with gray level smaller than or equal to

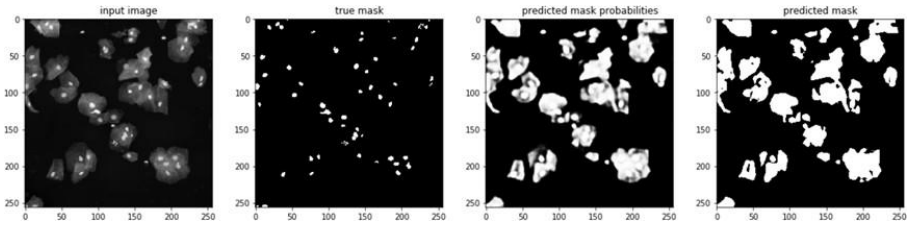

Fig. 8. Sample image with two distinct background and predicted mask by U-NET.

these peaks location, we can find the pixels, belonging to the true nucleus more likely. The resulted mask is combined with the output of network to enhance the final results.

Eventually, some morphological operations, like opening, dilation and filling, are applied to resulted mask to improve the segmentation accuracy.

4.2 Final results

We did all this post pre-cessing and modifications to improve final segmentation performance. We submitted our results to Kaggle website, and our score increased form 20% to 32.7%. Also, we updated the *ACC* metric for the U-NET (having the best performance among the implemented three networks) and other two architectures after post-processing. As it is seen in table 4, the *ACC* measure increased a little bit after post-processing. The increment of *ACC* after post-processing step, is not as much as the increment of *P R* score on Kaggle website. This is because of the fact that, *ACC* is not sensitive to connected objects and number of objects, and what we did in post processing is mostly for connected object separation.

Table 4. Results on test data before and after psot procesing.

-	Recal: $\frac{TP}{TP+FN}$	Recal after post-proc	ACC	ACC after postproc
Simple CNN	91.4%	91.8%	89%	90.6%
Pixel-based CNN	90%	90.5%	87%	88%
U-NET	93.9%	94.6%	92%	92.7%

5 Conclusion

In this report, we presented nucleus segmentation algorithms in microscopic medical images. Particularly, we focused on machine learning algorithms and studied three network architectures (namely U-NET, simple CNN with no pooling and pixel-based CNN) in details. We evaluated these three methods on a

challenging data-set provided by Kaggle website and it was shown that U-NET outperform the other two networks. To improve the segmentation accuracy we applied several pre-processing and post-processing strategies. Specially, color normalization, intensity inversion and histogram equalization were done in the pre-processing phase, and the connected object separation and the local adaptive thresholding (to address the images with two different backgrounds) were applied in the post-processing phase. Performing the post-processing step, our Kaggle score increased form 20% to 32.7%. Other architecture, like Fast-RCNN and Mask-RCNN, seems to be robust against connected objects, and they can find the number of objects (nuclei in our case) more precisely. Therefore, we are going to try these networks as our future work.

References

1. Wienert, S.,Heim, D., Saeger, K., Stenzinger, A.,:Detection and Segmentation of Cell Nuclei in Virtual Microscopy Images: A Minimum-Model Approach:, Scientific reports V2, 11 July 2012.
2. Cosatto, E., Miller, M., Graf, HP., Meyer, JS.,: Grading nuclear pleomorphism on histological micrographs: 19th Int Conf Pattern Recog2008. p. 14.
3. Dalle., J-., Li, H., Huang, C-H., Leow, WK., Racoceanu, D., Putti, TC.,: Nuclear pleomorphism scoring by selective cell nuclei detection: IEEE Workshop Appl Comput Vis; 2009.
4. Qi, X., Xing, F., Foran, DJ., Yang, L.,: Robust segmentation of overlapping cells in histopathology specimens using parallel seed detection and repulsive level set:IEEE Trans Biomed Eng. 2012;59(3):75465. pmid:2216755
5. Phansalkar,N., More, S., Sabale, A., Joshi, M., : Adaptive local thresholding for detection of nuclei in diversity stained cytology images, in ICCSP, 2011, pp. 218220.
6. Sauvola, M. Pietikainen: Adaptive Document ImageBinarization, Pattern Recognition, vol. 33, pp. 225-236, 2000.
7. Kumar, N. , Verma, R., Sharma,S., Bhargava, S., Va- hadane, A., Sethi, A., : A dataset and a technique for gen- eralized nuclear segmentation for computational pathol- ogy, IEEE Transactions on Medical Imaging, 2017.
8. Lecun, Y., Bengio, Y., Hinton, G.,:Deep learning, Nature, vol. 521, no. 7553, pp. 436444, 5 2015.
9. Nair, V., Hinton, G., : Rectified linear units improve restricted boltzmann machines. Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel, 2010, pp. 807814.
10. Jafari,M.H., Naser-Esfahani,E., Karimi, N., :Extraction of skin lesions from non-dermoscopic images for surgical excision of melanoma:, Int J Comput Assist Radiol Surg. 2017 Jun;12(6):1021-1030.
11. Ronneberger,O., Fischer, P., Brox,T., :U-Net: Convolutional Networks for Biomedical Image Segmentation:, Cs.Cv , May 2015.
12. Vahadane,A., Peng, T., Sethi A., Albarqouni, S., Wang,L. , Baust, M.,Steiger, K. , Schlitter, A. M. , Esposito, I., Navab, N. , :Structurepreserving color normalization and sparse stain separation for histological images:, IEEE Transactions on Medical Imaging, vol. 35, no. 8, pp. 19621971, Aug 2016.