



Segmentation of Cell Nuclei in Fluorescence Microscopy Images Using Deep Learning

Hemaxi Narotamo, J. Miguel Sanches, and Margarida Silveira^(✉)

Institute for Systems and Robotics (ISR/IST), LARSyS, Instituto Superior Técnico,
Universidade de Lisboa, Lisbon, Portugal
{hemaxi.narotamo,jmrs}@tecnico.ulisboa.pt,
msilveira@isr.tecnico.ulisboa.pt

Abstract. Cell nuclei segmentation is important for several applications, such as the detection of cancerous cells and cell cycle staging. The main challenges and difficulties, associated with this task, arise due to the presence of overlapping nuclei, image intensity inhomogeneities and image noise.

Several classical methods have been proposed for cell nuclei segmentation. However, they depend strongly on manual setting of parameters and they are sensitive to noise. Recently, deep learning is becoming state-of-the-art, due to its enhanced performance in many tasks of computer vision, such as object detection, classification and segmentation. Deep learning models are robust to the presence of noise and are able to automatically extract meaningful features from the image. Although deep learning models perform significantly better than the traditional methods, they are computationally more expensive.

In this paper we present a computationally efficient approach for high throughput nuclei segmentation based on deep learning. Our approach combines the object detection capability of Fast YOLO with the segmentation ability of U-Net. We applied our method to 2D fluorescence microscopy images with DAPI stained nuclei. Our results show that our method is competitive with Mask R-CNN, but significantly faster. In fact, with our method, an image of size 1388×1040 is segmented in approximately 1.6s which is about nine times faster than the Mask R-CNN (15.1s). Additionally, our results show that the improvements in computational efficiency come at only a small cost in performance.

Keywords: Deep learning · Nuclei segmentation · Cell imaging

This work was supported by Portuguese funds through FCT (Fundação para a Ciência e Tecnologia) through the projects TRACE (PTDC/BBB-IMG/0283/2014) and reference UID/EEA/50009/2019.

© Springer Nature Switzerland AG 2019
A. Morales et al. (Eds.): IbPRIA 2019, LNCS 11867, pp. 53–64, 2019.
https://doi.org/10.1007/978-3-030-31332-6_5

1 Introduction

Nuclear segmentation provides valuable information about nuclei morphology, DNA content and chromatin condensation. For instance, morphological and textural features can be used for cell cycle staging [9, 18] and detection of pathological mutations associated with cancer [14]. Cell overlapping, image noise and non-uniform acquisition and preparation parameters make the segmentation procedure a challenging task.

Manual segmentation is time-consuming and depends on the subjective assessment of the human operator [8]. Thus, it is not a practical approach in high-throughput applications where a huge number of nuclei need to be accurately detected. Hence, new automatic segmentation tools are needed and machine learning and computer vision approaches are the most common choices [21].

Classical approaches include Otsu's thresholding followed by watershed algorithm, graph-cuts based methods, K-means clustering and region growing [12, 20]. However, these techniques often require the tuning of manual parameters, they are sensitive to noise and sometimes can be very specific for given types of images. Deep learning approaches, successfully applied in many other fields, are obvious choices for nuclei segmentation because they are robust to noise, able to learn automatically the parameters and present a good generalization capacity.

Several deep learning based approaches have been proposed for cell nuclei segmentation and it is shown that their performance is better in terms of accuracy when compared to traditional techniques mentioned above. U-Net [17], a simple and computationally efficient convolutional network, winner of the Cell Tracking Challenge in 2015, is one of the most used architectures in biomedical image segmentation and cell nuclei as well. It performs semantic segmentation, that is, it makes classification in a pixel wise basis. It is able to classify single pixels but not objects (sets of pixels). For example, if two or more nuclei are touching, it will classify them as being a single object. Since in nuclei segmentation task each nucleus needs to be identified separately, several authors proposed methods, based on U-net, to address this difficulty. Ronneberger et al. had already proposed, in the original U-net paper [17], the use of a weighted cross entropy loss function where the weight maps are created in a way to give higher weights to pixels that are closer to two or more boundaries, in that way the model can learn the separation between close objects. Other approaches convert the binary problem into a ternary one, by changing the last layer of U-Net to predict not only the nuclei but also the contour of each nucleus [5, 7]. Recently, the winners of the Kaggle data competition 2018 [1, 2] have shown a novel way to tackle the problem of nuclei segmentation. They changed the ground truth masks by adding a third channel that represents the touching borders between nuclei. In this way the masks contain three classes: background, nuclei and touching borders. Furthermore, they used an encoder-decoder type architecture based on U-Net and the encoder was initialized with pre-trained weights. Since then several studies [10, 19] have applied similar approaches using U-Net by allowing it to predict both the nuclei and touching borders.

Recently, He et al. [11] proposed Mask R-CNN. This is an architecture designed for instance segmentation, where the main goal is to obtain a segmentation mask for each object in the image. It corresponds to the segmentation of individual objects in an image. Instance segmentation is a combination of object detection, where the goal is to identify each object’s category and bounding box, and segmentation. Therefore, in instance segmentation, different instances of the same object have different labels. Mask R-CNN has essentially two stages, the first stage is a region proposal network (RPN) which generates region proposals. For each pixel, it proposes k bounding boxes and a score that tells if the bounding box contains an object or not. In the second stage, for each of the bounding boxes proposed by RPN, features are extracted and classification and bounding box regression is performed. Additionally, the mask branch generates a segmentation mask for the object enclosed in the bounding box. Although Mask R-CNN was developed for segmentation of natural images, Johnson et al. [13] have demonstrated that it can be used for the task of nuclei segmentation. Similar conclusions are drawn by Vuola et al. [19], in a study where a comparison between U-Net, Mask R-CNN and an ensemble of these two models was made. Their results showed that Mask R-CNN performs better in the nuclei detection task and U-Net performs better in the segmentation task. Finally, by combining the strengths of both models, the ensemble model performs better than both models separately. However, the main problem associated with Mask R-CNN is its high computational cost.

In this work we present an alternative to Mask R-CNN for nuclei instance segmentation. Speed is an important factor to take into consideration if the method is going to be implemented in clinical routine. We propose a deep learning based approach that achieves good segmentation results and is computationally efficient. Our approach is based on a combination of Fast YOLO for instance detection and U-Net for segmentation. According to [15], Fast YOLO can be used for real time object detection in video and it is one of the fastest object detection methods, hence its superiority compared to Mask R-CNN with respect to computational efficiency.

This paper is organized as follows: in Sect. 1 a review of the methods for cell nuclei segmentation and the goal of this work were presented. In Sect. 2 a novel deep learning approach is presented for cell nuclei instance segmentation. Section 3 describes the dataset used in this project, training of the proposed approach and evaluation metrics. In Sect. 4, the main experimental results regarding nuclei segmentation are presented as well as a comparison with some state-of-the-art methods mentioned in Sect. 1. Finally, in Sect. 5 conclusions and topics requiring future studies are presented.

2 Proposed Approach

The proposed approach for cell nuclei instance segmentation is based on a combination of two deep learning models: Fast YOLO [15] and U-Net [17], as illustrated in Fig. 1. YOLO is an architecture designed for object detection and classification, which is faster than Mask R-CNN. This is due to the fact that instead of

using an RPN, which is based on a sliding window approach, YOLO applies a single network to the full image. In YOLO the image is divided into regions and then bounding boxes and class probabilities are predicted for each region. There is just one single network that divides the image and predicts objects and its corresponding classes, additionally this network can be trained end-to-end.

We used a smaller version of YOLO (Fast YOLO¹), which has fewer convolutional layers, hence it is faster than YOLO. Nevertheless, Fast YOLO only gives a bounding box for each detected nucleus, and we want to obtain a segmentation mask for each nucleus. Therefore, we combine Fast YOLO with an U-Net trained to segment individual nuclei. We start by feeding the input image to the Fast YOLO, this will provide us bounding boxes for all detected objects in that image, steps A and B in Fig. 1, respectively. After this step, for each bounding box, the corresponding image patch is extracted and resized to a patch of size 80×80 (see step C in Fig. 1), this patch is then fed to the U-Net which will give as output a binary mask, where 0 and 1 denote pixels belonging to the background and nucleus, respectively, (step D in Fig. 1). Then, the output of the U-Net, which has size 80×80 , is resized again to its original size. Finally, a spatial arrangement (step F in Fig. 1) is necessary to obtain the final segmentation mask.

The objective of the proposed approach is to first minimize the loss function of the Fast YOLO network, as described in [15], and then minimize the loss function of the U-Net, which we've defined as:

$$Loss = 0.5 \times \text{binary cross entropy} + 0.5 \times (1 - \text{dice coefficient}) \quad (1)$$

where the definition of binary cross entropy (BCE) and dice coefficient (DC) is represented in Eqs. 2 and 3, respectively.²

$$BCE = - \sum_{i=1}^N y_i \times \log(\hat{y}_i) + (1 - y_i) \times \log(1 - \hat{y}_i) \quad (2)$$

$$DC = \frac{2 \times |X \cap Y|}{|X| + |Y|} \quad (3)$$

3 Experiments

In this section the dataset used in the experiments is described. Additionally, details regarding the training of the proposed deep learning approach and evaluation metrics used to measure the performance of the model are presented.

¹ The architecture of Fast YOLO (illustrated in Fig. 1), is different from the one presented in the original YOLO paper [15]. In fact, the architecture of Fast YOLO used in this work is a smaller version of YOLOv2, which has some improvements over YOLO, as stated in [16].

² N is the total number of pixels in a given image, y_i is the true label of the pixel i and \hat{y}_i denotes the predicted label for the pixel i.

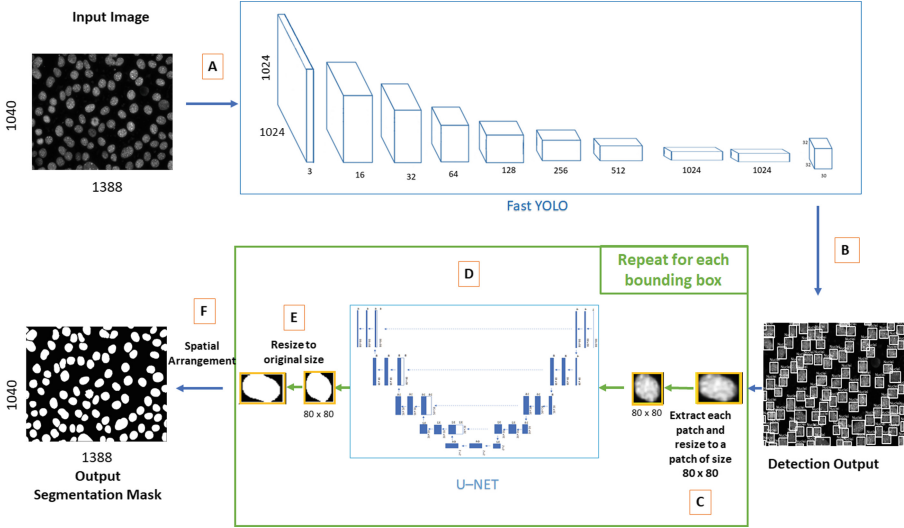


Fig. 1. Overview of the proposed approach for cell nuclei instance segmentation. The input image is fed to the Fast YOLO architecture (step A). Fast YOLO will give as output bounding boxes for all of the detected objects in the input image (step B). Afterwards, each patch inside the bounding box proposed by the previous architecture is resized to 80×80 (step C) and fed to the U-Net (step D). The output patch of the U-Net is then resized to the original size (step E). Finally, after spatial arrangement, the final segmentation mask is obtained, (step F).

3.1 Data

The training dataset used in the experiments consists of 130 fluorescence microscopy images of normal murine mammary gland cells stained with DAPI, with size 1388×1040 . This dataset comes from the study presented by Ferro et al. [9]. Additionally, another dataset with one nucleus per image and patch size 80×80 was necessary to train the U-Net. This dataset was obtained from the original one by using the skimage tool regionprops. For each image, the ground truth mask was labeled, then regionprops tool was applied to measure the properties of the labeled mask regions, which include the bounding box coordinates for each object. This operation allows to extract one patch per nucleus, which is then resized to a patch of size 80×80 .

3.2 Training

Fast YOLO. The implementation used for Fast YOLO is based on a publicly available implementation by Thtrieu which was released under the GNU General Public License v3.0 [3]. In order to train the Fast YOLO with our dataset, we had to generate XML files based on Pascal VOC format. These files were generated from the ground truth data using skimage tool regionprops and lxml.etree

module. We’ve adapted the network for our problem, the number of classes in our problem is one, therefore we changed the number of filters of the last layer to 30, according to the formula $5 \times (classes + 5)$ [3]. We also changed the number of classes to 1 and resized the original image to an image of size 1024×1024 . All of the other parameters remained unchanged. Finally, Fast YOLO was trained from scratch, using Adam optimizer, first it was trained for 200 epochs with learning rate 0.0001, then it was trained for another 600 epochs with learning rate 0.00001.

U-Net. We implemented the U-Net model using Keras with Tensorflow backend. The architecture of the model implemented is represented in Fig. 2. This model was trained for 100 epochs, with a learning rate of 0.001, using Adam optimizer, without dropout, with Xavier initialization and with ReLU as activation functions, except the final activation function of the last layer which is a sigmoid activation function.

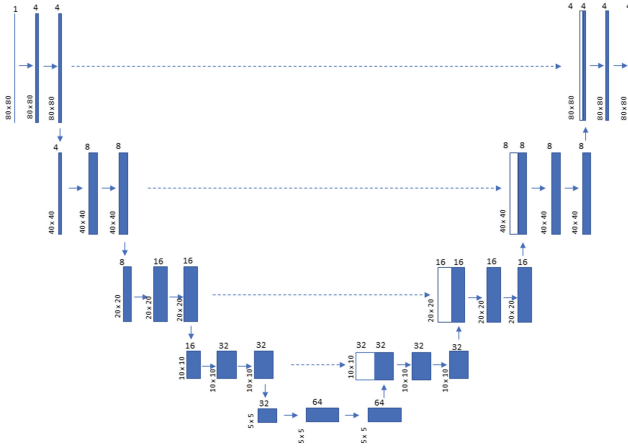


Fig. 2. U-Net architecture used for the segmentation step of the proposed approach.

3.3 Evaluation Criteria

To test the performance of the proposed segmentation model, we calculated the F1 score (see Eq. 7), at different thresholds of the intersection over union (IoU). The IoU between two objects is given by:

$$IoU = \frac{Area\ of\ overlap}{Area\ of\ union} \tag{4}$$

For each image, an $m \times n$ matrix is built. Where m denotes the total number of objects in the ground truth mask, n the total number of objects in the predicted mask. And the component (i,j) corresponds to the IoU (Eq. 4) between

object i and object j . The F1 score was calculated after applying different thresholds to this matrix. That is, the F1 score was computed by varying the IoU threshold (T) from 0.5 to 0.95, by steps of size 0.05. The F1 score requires the calculation of the precision (Eq. 5) and recall (Eq. 6). Where TP, FP and FN stand for true positives, false positives and false negatives, respectively. In one hand, a nucleus detected by an automatic technique is considered as TP if, for a given IoU threshold (T), its IoU with some ground truth nucleus is higher than T . On the other hand, if its IoU is lower than T , it is considered as FP (extra object). Finally, if for a given ground truth nucleus there isn't a corresponding detection, it will be considered as FN (miss detection).

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{FN + TP} \quad (6)$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (7)$$

3.4 Performance Comparison

We measured the performance of our approach and compared it with the performance of four approaches: Yen's thresholding plus watershed [12,22], Original U-Net [17], similar approach to the winning solution of Kaggle 2018 [2] and Mask R-CNN [11]. To simplify we denote these models as: Yen + watershed, Original U-Net, Kaggle_2018 and Mask R-CNN, respectively. To compare the performance between different models, a 13-fold cross validation was performed. In other words, for each approach, except for Yen + watershed, we've trained 13 models with 120 images each and tested their performance on 10 images. We perform leave-one-experiment-out cross-validation in order to avoid the bias introduced during the evaluation, that is, to avoid the bias that would be introduced when testing the model in images that were acquired in the same experiment as some images that were used to train this model. The final F1 Score for each approach is an average over the 13 models.

All experiments were carried out on an NVIDIA GPU GTX 1050 (4 GB) and in Python 3.6. Additionally, all implementations are based on open-source deep learning libraries Tensorflow and Keras [4,6].

4 Results

In this section, results regarding nuclei segmentation, F1 score and computational efficiency are presented. Additionally, a comparison with four state-of-the-art methods is made.

4.1 Nuclei Segmentation

We compared the performance of our approach with other state-of-the-art methods. Figure 3 shows a visual comparison between different models, regarding nuclei segmentation. The four images (in the first row) were chosen in order to emphasize the variability that exists between different input images. These images represent the blue channel of the corresponding original fluorescence images. As stated before, our dataset contains images stained with DAPI, which is a nuclear stain that binds to the DNA and emits blue fluorescence. Therefore, the blue channel of the original images contains information regarding the nuclei.

The third row in Fig. 3 shows the segmentation masks obtained by applying Yen + watershed, by observing this row it can be concluded that in some cases the segmentation masks are bigger than the ground truth masks. This is a disadvantage of the thresholding methods. Additionally, in comparison with other approaches, this is the approach that presents more merges, i.e., two or more nuclei that are joined into a big object. Results regarding Original U-Net (fourth row) and Kaggle.2018 (fifth row) show that although these approaches separate better the touching nuclei, in some cases there are gaps between these nuclei. This can be explained by our ground truth data which also has gaps between touching nuclei, in order to solve the problem as an instance segmentation problem. On the other hand, in the results obtained with Mask R-CNN and our approach those gaps disappear, since these two approaches are designed specifically to solve the problem of instance segmentation.

The last column in Fig. 3 illustrates why Mask R-CNN model outperforms all the other models. In this case there is high intensity variation along the input image and the image contains a lot of touching and occluded nuclei. Therefore, the classical method (Yen + watershed) struggles in detecting nuclei located on the left side of the image. Interestingly, our approach performs better than Yen + watershed, but still it fails to identify some of the occluded nuclei. This is due to the detection performance of Fast YOLO, which is worse than that of Mask R-CNN, specially in regions with occluded nuclei, where some of the nuclei aren't detected. Mask R-CNN is the one that provides the best segmentation mask for this input image. However, note that for the other three images the results obtained with Mask R-CNN and the ones obtained with our model are quite similar.

4.2 F1 Score vs IoU Threshold

Figure 4 shows a plot of average F1 score across increasing thresholds of IoU. The accentuated decrease of the F1 score, at $IoU \approx 0.80$, can be explained by the presence of inaccurate boundaries on our ground truth data. For example, since our ground truth masks are binary, in order to separate touching nuclei and to solve the problem as an instance segmentation problem, we have drawn lines to separate touching nuclei and considered the pixels contained in these lines as belonging to the background, (this can be observed in the second row in Fig. 3).

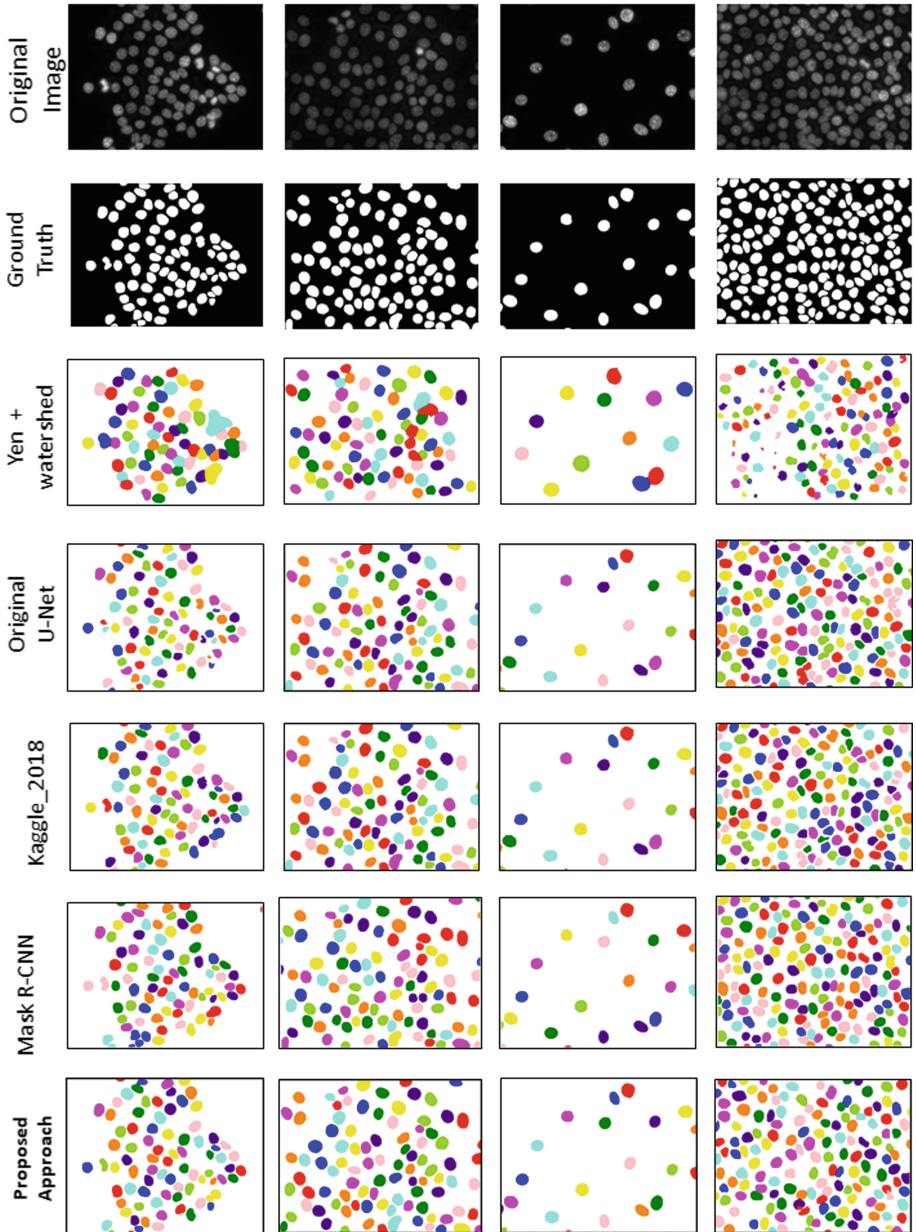


Fig. 3. Nuclei segmentation results obtained by applying different methods. The first row represents examples of the original images, for which we want to obtain the segmentation mask. The second row represents the corresponding ground truth masks. Finally, the third, fourth, fifth, sixth and seventh rows represent the corresponding segmentation results obtained by applying Yen + watershed, Original U-Net, Kaggle_2018, Mask R-CNN and the proposed approach, respectively. (Color figure online)

By comparing the deep learning approaches with the traditional method (Yen + watershed), we can conclude that deep learning models significantly outperform this classical method. Additionally, for $IoU < 0.75$ the performance of our approach is similar to the performance of Mask R-CNN and better than that of all of the other methods.

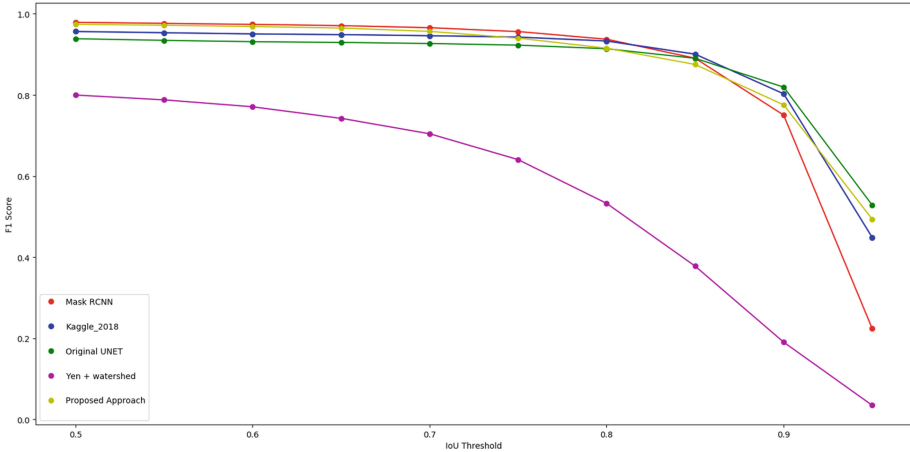


Fig. 4. Average F1-Score vs IoU threshold, comparison between different models: Yen + watershed (purple), original U-Net (green), Kaggle_2018 (blue), Mask R-CNN (red), proposed approach (yellow). (Color figure online)

4.3 Computational Efficiency

Regarding computational efficiency we compared the training time and the test time required by all the methods. Training time corresponds to the time a model needs to learn a given task, in our case, the task of nuclei instance segmentation. By observing Fig. 5(a), we can conclude that Mask R-CNN requires significantly more time to learn the task of nuclei segmentation (about 1420 min), in comparison with all the other models. Although our approach requires more time to train (450 min) than the Original U-Net (14 min) and Kaggle_2018 (100 min), it also provides better segmentation masks as illustrated in Fig. 3.

On the other hand, regarding test time, which is the time required for a model to give a segmentation prediction for an image, our results are presented in Fig. 5(b). These results show that Mask R-CNN is the model that presents the highest test time (15.1 s). Our approach in comparison with Mask R-CNN is about nine times faster. Furthermore, Yen + watershed requires 1.8 s, which is of the same order of magnitude as the test time of our approach (1.6 s), however Yen + watershed presents the worst performance, as observed in Fig. 4.

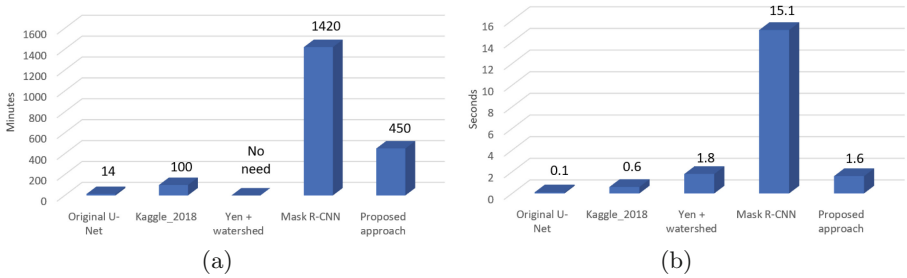


Fig. 5. (a) Training time (in minutes) associated with each model. (b) Mean test time per image (in seconds) for each model, for images of size 1388×1040 .

5 Conclusions and Future Work

This paper addresses the important problem of nuclei segmentation for high throughput applications.

We proposed a new approach that combines the Fast YOLO architecture, specially designed for detection, with the U-Net that was conceived mainly for segmentation purposes.

The segmentation quality obtained with the proposed method is comparable to the existing deep-learning based state-of-the-art methods, e.g. Mask R-CNN, but a significant reduction of almost $10\times$ on the segmentation time was obtained.

In the future, morphological and textural features will be extracted from the segmented nuclei for diagnosis of pathogenic mutations associated with cancer.

Acknowledgments. We acknowledge the EPIC (Epithelial Interactions in Cancer) from i3S/Ipatimup by the biological support of this research. In particular for providing the image data used in this paper.

References

1. Find the nuclei in divergent images to advance medical discovery. <https://www.kaggle.com/c/data-science-bowl-2018/discussion>. Accessed 20 Feb 2019
2. [ods.ai] topcoders, 1st place solution on data science bowl 2018. <https://www.kaggle.com/c/data-science-bowl-2018/discussion/54741#latest-477226>. Accessed 20 Feb 2019
3. Darkflow (2018). <https://github.com/thtrieu/darkflow>
4. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous systems (2015). <https://www.tensorflow.org/>, software available from tensorflow.org
5. Caicedo, J.C., et al.: Evaluation of deep learning strategies for nucleus segmentation in fluorescence images. *BioRxiv*, p. 335216 (2019)
6. Chollet, F., et al.: Keras (2015). <https://keras.io>
7. Cui, Y., Zhang, G., Liu, Z., Xiong, Z., Hu, J.: A deep learning algorithm for one-step contour aware nuclei segmentation of histopathological images. *arXiv preprint arXiv:1803.02786* (2018)

8. Deshmukh, B.S., Mankar, V.H.: Segmentation of microscopic images: a survey. In: 2014 International Conference on Electronic Systems, Signal Processing and Computing Technologies, pp. 362–364. IEEE (2014)
9. Ferro, A., Mestre, T., Carneiro, P., Sahumbaiev, I., Seruca, R., Sanches, J.M.: Blue intensity matters for cell cycle profiling in fluorescence DAPI-stained images. *Lab. Investig.* **97**(5), 615 (2017)
10. Guerrero-Pena, F.A., Fernandez, P.D.M., Ren, T.I., Yui, M., Rothenberg, E., Cunha, A.: Multiclass weighted loss for instance segmentation of cluttered cells. In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 2451–2455. IEEE (2018)
11. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
12. Irshad, H., Veillard, A., Roux, L., Racoceanu, D.: Methods for nuclei detection, segmentation, and classification in digital histopathology: a review-current status and future potential. *IEEE Rev. Biomed. Eng.* **7**, 97–114 (2014)
13. Johnson, J.W.: Adapting Mask-RCNN for automatic nucleus segmentation. arXiv preprint [arXiv:1805.00500](https://arxiv.org/abs/1805.00500) (2018)
14. Mestre, T., Figueiredo, J., Ribeiro, A.S., Paredes, J., Seruca, R., Sanches, J.M.: Quantification of topological features in cell meshes to explore E-cadherin dysfunction. *Sci. Rep.* **6**, 25101 (2016)
15. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
16. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271 (2017)
17. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
18. Roukos, V., Pegoraro, G., Voss, T.C., Misteli, T.: Cell cycle staging of individual cells by fluorescence microscopy. *Nat. Protoc.* **10**(2), 334 (2015)
19. Vuola, A.O., Akram, S.U., Kannala, J.: Mask-RCNN and U-net ensemble for nuclei segmentation. arXiv preprint [arXiv:1901.10170](https://arxiv.org/abs/1901.10170) (2019)
20. Xing, F., Yang, L.: Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: a comprehensive review. *IEEE Rev. Biomed. Eng.* **9**, 234–263 (2016)
21. Xue, Y., Ray, N.: Cell detection in microscopy images with deep convolutional neural network and compressed sensing. arXiv preprint [arXiv:1708.03307](https://arxiv.org/abs/1708.03307) (2017)
22. Yen, J.C., Chang, F.J., Chang, S.: A new criterion for automatic multilevel thresholding. *IEEE Trans. Image Process.* **4**(3), 370–378 (1995)