Deep Learning Based Cell Nuclei Segmentation: An Application for Cell Cycle Staging

Hemaxi Narotamo
hemaxi.narotamo@tecnico.ulisboa.pt
Instituto Superior Técnico, Lisbon, Portugal

October 2019

Abstract

Cell nuclei segmentation is crucial in several bioimaging tasks such as cell counting and tracking, analysis of cell morphology and quantification of molecular expression. Accurate automatic nuclei segmentation is of special interest in high-throughput applications of microscopic images of cells or tissues. Cell nuclei segmentation is an open problem and an active field of research in the image processing community. In this work, a novel deep learning based approach for high-throughput nuclei instance segmentation is proposed. It combines the detection capability of Fast YOLO with the segmentation ability of U-Net. With this method significant improvements on the processing time (approximately 9 times less) are achieved in comparison with the Mask R-CNN, without compromising the F1 score of the results. Furthermore, the masks obtained with the proposed segmentation method were used to address the important problem of cell cycle staging from 4',6-diamidino-2-phenylindole (DAPI) stained nuclei. A (supervised) support vector machine (SVM) based approach is proposed, where the input features and ground truth labels used to train the classifier were automatically computed from nuclei stained with DAPI and fluorescence ubiquitination cell cycle indicator (FUCCI), respectively. Results leading to F1 score of almost 90% (F1 score: 0.877 ± 0.010) were obtained, where in 21 out of the 130 test images a F1 score of 100% was achieved. Testing the classifier on images from a different cell line from the one used to train the classifier lead to the impressive recall value of 93%, which demonstrates the robustness of the proposed approach for cell cycle staging.
Keywords: Deep learning, Nuclei segmentation, Cell imaging, Cell cycle staging

1. Introduction

Nuclear segmentation provides valuable information about the deoxyribonucleic acid (DNA) content [1], chromatin condensation [2] and nuclei morphology [3]. For instance, morphological and textural features can be used for cell cycle staging [1, 4] and detection of pathological mutations associated with cancer [5]. Cell overlapping, image noise and non-uniform acquisition and preparation parameters make the nuclei segmentation procedure a challenging task [2, 3]. Manual segmentation is time-consuming and depends on the subjective assessment of the human operator [6]. 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 [2, 3].

Moreover, nuclei segmentation is important for several applications, for instance, it can be used to address the important problem of cell cycle staging. Cell cycle is a fundamental mechanism of living organisms. The main goal is to ensure that when a cell divides it passes correctly the genetic information to the next generation. Additionally, dysregulation of the cell cycle is at the origin of many diseases, such as cancer, ischemia and neurodegenerative disorders. Hence, by studying and controlling the cell cycle one can understand the mechanisms of various diseases, and consequently find ways to tackle them [7]. Cell cycle comprises the interphase and the mitosis. The interphase has three stages: G1, S and G2. In G1 phase the DNA content of cells is 2N. In S phase, DNA replication occurs and its content varies from 2N to 4N. This phase is followed by G2 phase where the DNA content is 4N. Cell cycle can therefore be studied based on the cell's DNA content [7, 8]. The DNA content can be studied by staining the cells with fluorescent molecules that bind to the DNA (e.g. DAPI) and, afterwards, by analyzing the fluorescence intensity of individual cells [8].

In this work, the main goal is twofold (fig. 1):

1. Development of an automatic tool for cell nuclei instance segmentation based on deep learning.
2. Development of an automatic tool for cell cycle staging from DAPI images.

Part of this work was already published in the proceedings of the international conference IbPRIA 2019 [9].

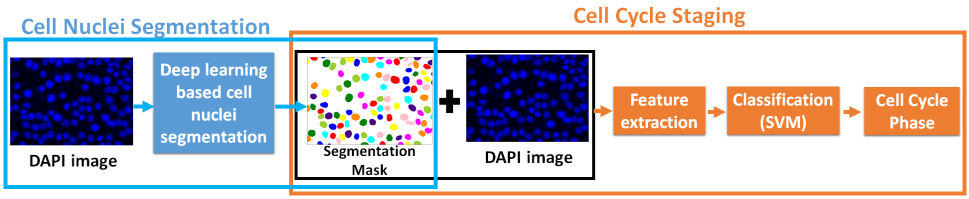


Figure 1: Schematic representation of this project's goal.

2. State-of-the-Art

This subsection presents the state-of-the-art regarding cell nuclei segmentation and cell cycle staging from microscopy images.

2.1. Cell Nuclei Segmentation

Classical approaches for nuclei segmentation include Otsu's thresholding followed by watershed algorithm, graph-cuts based methods, K-means clustering and region growing [2, 3]. 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 [3]. Deep learning approaches, successfully applied in many other fields, are obvious choices for nuclei segmentation because they are able to automatically extract meaningful features from the image and they can be more robust to noise when compared to traditional machine learning approaches [10, 11].

Several deep learning based approaches have been proposed for cell nuclei segmentation. U-Net [12], 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 [12], 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 the U-Net to predict not only the nuclei but also the contour of each nucleus [13, 14]. Recently, the winners of the Kaggle data competition 2018 [15, 16] have shown a novel way to tackle the problem of nuclei segmentation. They used an architecture based on the U-Net, and 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. Since then, several studies [17, 18] have applied similar approaches using U-Net by allowing it to predict both the nuclei and touching borders.

Recently, He et al. [19] proposed Mask R-CNN. This is an architecture designed for instance segmentation, where 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 the segmentation of natural images, Johnson [20] has shown that it can be used for the task of nuclei segmentation. Similar conclusions are drawn by Vuola et. al [18], in a study where a comparison between U-Net, Mask R-CNN and an ensemble of these two models was made. However, the main problem associated with Mask R-CNN is its high computational cost.

To sum up, the two architectures that have been most commonly used in order to tackle the problem of cell nuclei segmentation are the U-Net and Mask R-CNN. U-Net is designed for semantic segmentation, so it doesn't have the ability to distinguish different instances of the same object. Mask R-CNN is designed for instance segmentation, however it is a computationally expensive architecture. And if the method is going to be implemented in the clinical routine, speed is an important factor to take into consideration. Therefore, research is needed in the area of nuclei segmentation, in order to develop an instance segmentation approach based on deep learning, which provides good segmentation results and is computationally efficient.

2.2. Cell Cycle Staging

Throughout the years several works have been proposed to infer the cell cycle phase of cells based on their DNA content, shape and texture [21, 22, 23, 24]. However, they typically focus on the following phases: interphase and the stages of the mitosis (prophase, metaphase, anaphase(/telophase)). That is, none of these

works focuses on the stages of the interphase (G1, S, G2). Furthermore, the nuclei segmentation step of these works relies on traditional nuclei segmentation methods.

Recently, Roukos et al. (2015) [4] proposed a protocol to quantify DNA content on fluorescence images of cells stained with DAPI. For each nucleus, the integrated DAPI intensity was calculated and represented in a histogram. This histogram can be divided in three regions that correspond to the three interphase stages: G1, S and G2. Although this method can be used for interphase cell cycle staging of individual cells, it is based on manually defined thresholds which can be subjective.

Ferro et al. (2017) [1] proposed a new framework for interphase cell cycle staging based on two DAPI features: the area and the total intensity. For each image, nuclei segmentation was performed and for each nucleus the two DAPI features were extracted. These features form the feature space in which a modified k-means clustering is performed. It classifies each nucleus regarding its cell cycle phase (G1, S or G2). Although this work presents an automatic algorithm for cell cycle staging, it presents some drawbacks. For instance, the features that are used in this work can vary between different images. For example, the total DAPI intensity depends on the acquisition parameters, which may be different across different images. Additionally, the area of the nuclei may vary between different cell lines. Therefore, the k-means clustering algorithm needs to be applied per image, which may take a lot of time if there are several images.

DeepFlow, is a deep learning based architecture that was proposed by Eulenberg et al. (2017) [25] for cell cycle staging. DeepFlow takes as input two images obtained from imaging flow cytometry (IFC), and extracts features from these images providing a classification label. In this work an accuracy of 79.40 % is obtained when considering 7 classes (G1, S, G2 and the four stages of the mitosis). And an accuracy of 98.73 % when considering five classes (in this case the G1, S and G2 phases were considered as a single class, the interphase), which is higher than the accuracy obtained when considering the 7 classes. Thus, their results suggest that it is difficult to distinguish cells in G1, S and G2 phases based on the images provided by IFC, since these images are very similar.

To sum up, there aren't many tools for interphase cell cycle staging based on supervised machine learning techniques. The one proposed in [25] presents good results, however it uses IFC which requires cells in suspension. Most of human cells are adherent cells. Flow cytometry analysis can be affected when suspension cell culture is created from adherent cells [7, 26]. Therefore, it is important to design new tools for interphase cell cycle staging of adherent cells.

3. Proposed Approaches

This section presents the proposed approach for cell nuclei segmentation and the proposed approach for cell cycle staging.

3.1. Cell Nuclei Segmentation

In this work a new approach for cell nuclei instance segmentation is proposed. This approach combines the object detection ability of Fast YOLO [27] with the segmentation capability of U-Net [12]. [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 a 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 the objects and its corresponding classes.

A schematic representation of the proposed approach is presented in fig. 2. First, the input image is fed to Fast YOLO (step A), which will divide the image into regions and propose bounding boxes and confidence scores for each region. After non-maximal suppression, Fast YOLO will provide an output with all detected objects and respective classes (step B). In this problem there is only one class: the nucleus. Thereafter, for each nucleus detected by the Fast YOLO, the respective patch will be extracted from the input image and it will be resized to a patch of size 80×80 (step C). This step is repeated for each object detected by the Fast YOLO. Considering that there are N objects detected, after step C there is a batch of images of size $80 \times 80 \times N$, this is fed to the U-Net (step D). The output of the U-Net is a binary segmentation mask for each image (step D). That is, pixels belonging to the foreground (nucleus) will have value 1 and pixels belonging to background will have value 0. Afterwards, steps E and F are repeated for each nucleus's binary mask. Step E resizes the nucleus patch to its original size and finally step F places the patch in the final segmentation mask on the location where it belongs to. Furthermore, each nucleus detected by Fast YOLO will have an unique label. The output segmentation mask is colored to visualize better the quality of the segmentation masks. Note that the colormap used doesn't have enough distinct colors to cover all nuclei, this is used just for visualization purposes.

The goal of the proposed approach is to first minimize the loss function of the Fast YOLO and then

[1] In [27] the authors proposed YOLO and Fast YOLO. The architecture used in the approach proposed in this work is a smaller version of YOLOv2 [28]. Therefore, it is Fast YOLOv2, however, to simplify it will be designated as Fast YOLO.

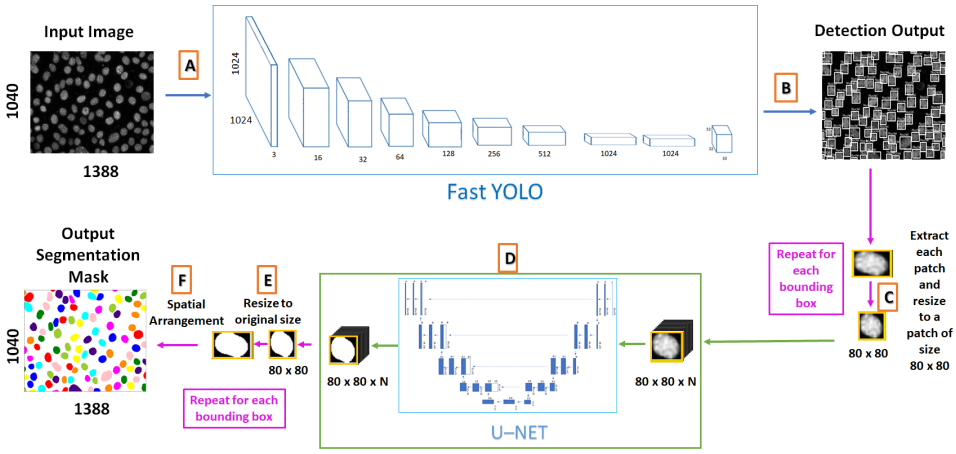


Figure 2: Overview of the proposed approach for cell nuclei instance segmentation.

the loss function of the U-Net. Fast YOLO's loss function is similar to the one proposed in [27]. To train the U-Net the loss function proposed in [16] is minimized, which is defined as:

$$L = \frac{1}{2}BCE + \frac{1}{2}(1 - DC) \quad (1)$$

where the definition of BCE and DC is represented in eqs. 2 and 3, respectively.

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

where N is the total number of pixels in the image, $y_i \in \{0, 1\}$ is the true label of pixel i and $\hat{y}_i \in [0, 1]$ denotes the predicted probability for pixel i.

$$DC = \frac{1}{N}\frac{2|X \cap Y|}{|X| + |Y|} \quad (3)$$

where X is the ground truth binary mask and Y is the predicted probability map.

3.2. Cell Cycle Staging

In this work, a new approach for interphase cell cycle staging from images of cells stained with DAPI is presented. First, nuclei segmentation is performed in DAPI images according to the method proposed in subsection 3.1. This method gives a segmentation mask for each DAPI image. Thereafter, for each nucleus in the segmentation mask, features are extracted from the mask and from the corresponding DAPI image (these features are presented in subsection 4.5). Afterwards, the features are fed to the SVM (a supervised machine learning classifier) which will output a label regarding the cell cycle phase (G1 or S/G2) for each nucleus (see fig. 3).

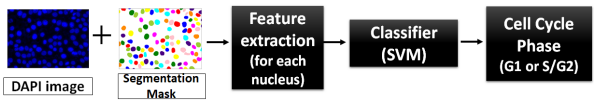


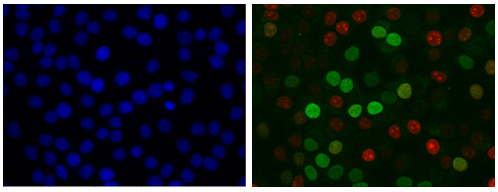
Figure 3: Supervised machine learning classifier (SVM) for interphase cell cycle staging.

4. Experiments

In this section the dataset used in the experiments is described. Additionally, details regarding the image preprocessing steps are presented. This subsection also presents details regarding the training of the proposed approach for nuclei segmentation and the steps followed to train the proposed approach for cell cycle staging. Finally, evaluation metrics used to measure the performance of the models are presented.

4.1. 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, and the corresponding 130 fluorescence microscopy images stained with FUCCI (fig. 4). This dataset comes from the study presented by Ferro et al. [1]. Additionally, another dataset with one nucleus per image and patch size 80×80 was necessary to train the U-Net of the proposed approach. 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.



(a) **(b)**

Figure 4: Example of a DAPI image and corresponding FUCCI image. a) DAPI image. b) FUCCI image.

Additionally, a dataset of gastric cancer cells stained with DAPI and with cyclin B1 was analysed. The cyclin B1 is a marker for cells in G2/M phases. An example of an image from this new cell line is shown in fig. 5. In these images the nuclei in G2 phase were annotated manually by the biologists based on the expression of the cyclin B1 (indicated with red arrows in fig. 5).

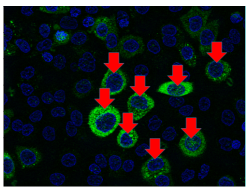


Figure 5: Example of an image from the gastric cancer cell line.

4.2. Data Preprocessing

Both DAPI images and FUCCI images have values ranging from 0 to 255. However, a lot of factors influence the image quality, such as the illumination conditions and image acquisition parameters. For instance, one image can have values ranging from 0 to 255 and another image from 0 to 100. In order to have all inputs in a comparable range and with small values, a simple normalization step was performed. For each image its values were divided by 255.

In the context of this work, the DAPI images acquired in different experiments were similar, that is, there weren't significant differences. However, for the FUCCI images there were clearly some of them in which the background was greenish. In order to make the intensity between all the images comparable, a normalization step was performed for the FUCCI images. In this step, for each image, for each color channel, the mean background intensity was subtracted from the image.

4.3. Computational Environment

All experiments were carried out on a NVIDIA GPU GTX 1050 (4 GB) and in Python 3.6. Additionally, all deep learning implementations are based on open-source deep learning libraries Tensorflow and Keras [29, 30].

4.4. Cell Nuclei Segmentation

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 [31]. In order to train the Fast YOLO on the dataset used in this work, XML files based on Pascal VOC format needed to be generated. These files were generated from the ground truth data using skimage tool regionprops and lxml.etree module. The network was adapted for the problem studied in this work, the number of classes in this problem is one, therefore the number of filters of the last layer was changed to 30, according to the formula $5\times(classes+5)$ [31]. The U-Net was implemented using Keras [30] with Tensorflow backend [29].

The performance of the proposed approach was measured and compared with the performance of four approaches: Yen's thresholding plus watershed [2, 32], Original U-Net [12], similar approach to the winning solution of Kaggle 2018 [16] and Mask R-CNN [19]. To simplify these models will be denoted 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, 13 models were trained with 120 images each and their performance was tested on 10 images. In this case it is equivalent to a leave-one-experiment-out cross-validation. This was performed 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 (the concept of F1 score is explained in subsection 4.6).

4.5. Cell Cycle Staging

To train the supervised classifier input features and ground truth labels are needed. The input features are extracted from the DAPI images. Regarding the ground truth labels there are many ways to generate them. In this work these labels were generated based on the FUCCI technology. FUCCI is a fluorescent protein based sensor where the green and red fluorescent proteins are fused to two cell cycle regulators, the geminin and the cdt1, respectively [33]. Geminin is degradated in G1 phase and cdt1 is degradated in S/G2/M phases. Therefore, FUCCI labels the nuclei. Those stained with red are in the G1 phase, while the ones stained with green are in S/G2/M phases [33]. Thus, the ground truth label for each nucleus is created based on its color (green or red) in the FUCCI image. In this work an automatic way to generate these ground truth labels is presented. The overall pipeline to create ground truth labels from the FUCCI images is shown in fig. 6.

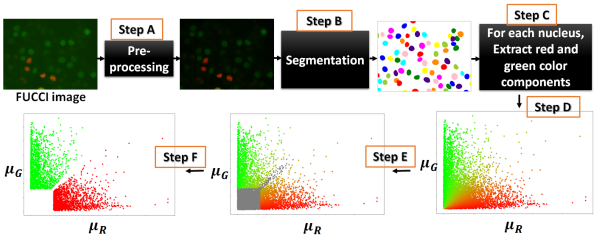


Figure 6: Overall pipeline to create the ground truth labels.

First, FUCCI images were normalized (step A in fig. 6). After this step, segmentation of FUCCI images was performed (step B in fig. 6). This step was performed using the deep learning based

approach proposed in this work. After obtaining the segmentation mask, features can be extracted for each nucleus (step C in fig. 6). The features that were extracted are shown in table 1, where b_p represents the n x m binary mask for each nucleus, where 0 denotes background and 1 denotes foreground/nucleus. And F_p represents the corresponding RGB nucleus in the FUCCI image.

Table 1: Feature Extraction from the FUCCI images.

Total Red Intensity	$T_R = \sum_{i=1}^{n}\sum_{j=1}^{m} b\_p(i,j)F\_p(i,j,0)$ ($F\_p(i,j,0)$ denotes the red channel)	(4)
Total Green Intensity	$T_G = \sum_{i=1}^{n}\sum_{j=1}^{m} b\_p(i,j)F\_p(i,j,1)$ ($F\_p(i,j,1)$ denotes the green channel)	(5)

Based on the features shown in table 1, other quantities were calculated:

Table 2: Quantities calculated from the features shown in table 1.

Mean Green Intensity (μ_G)	$\mu_G = \frac{T_G}{A}$	(6)
Mean Red Intensity (μ_R)	$\mu_R = \frac{T_R}{A}$	(7)
Normalized Red Intensity ($\hat{R}$)	$\hat{R} = \frac{\mu_R}{\sqrt{{\mu_R}^2+{\mu_G}^2}}$	(8)
Normalized Green Intensity ($\hat{G}$)	$\hat{G} = \frac{\mu_G}{\sqrt{{\mu_R}^2+{\mu_G}^2}}$	(9)

where $A = \sum_{i=1}^{n}\sum_{j=1}^{m} b\_p(i,j)$.

Afterwards, the nuclei from all images were represented in the 2D space (μ_R,μ_G) (step D in fig. 6). Additionally, each point was colored with the following RGB intensity: ($\hat{R}$,$\hat{G}$,0). Step E in fig. 6 consists in the removal of nuclei that are colorless (both early G1 and G0 cells are colorless, therefore it is difficult to discriminate visually these cells [7]). Additionally, outliers were removed and nuclei that are in the transition between G1 and S, which are more difficult to evaluate, were removed as well. The nuclei that were removed from further analysis are colored with gray (see plot in the middle in fig. 6). Finally, step F, in fig. 6, labels each nucleus according to the information provided by the FUCCI technology. That is, it labels a nucleus as G1 if $\mu_R > \mu_G$ and as S/G2 otherwise. The last plot (plot on left in fig. 6) shows the nuclei in the 2D space (μ_R,μ_G), where the each point is colored according to its label. That is, nuclei labeled as G1 are colored with red, and nuclei labeled as S/G2 are colored with green.

By following the steps shown in fig. 6 the ground truth labels for 3553 nuclei were obtained. These ground truth labels were generated based on the FUCCI molecular classification system.

Moreover, in order to train the classifier, input features are also needed. These features are extracted from the DAPI images. Therefore, DAPI images segmentation needs to be performed. In this case, the segmentation masks obtained for the FUCCI images were used to extract features from the DAPI images. This is due to the fact that the nuclei shapes and positions are the same in both DAPI and FUCCI images. The feature that was extracted from the DAPI images is the total DAPI intensity (eq. 10).

$$T_D = \sum_{i=1}^{n}\sum_{j=1}^{m} b\_p(i,j) \times D\_p(i,j) \tag{10}$$

where b_p is the nucleus's binary patch and D_p is the corresponding patch in the DAPI image.

Additionally, the following quantities were calculated for each nucleus: [2,3]

$$\hat{A} = \frac{A - \mu_A}{\sigma_A} \tag{11}$$

$$\hat{I_D} = \frac{T_D - \mu_{T_D}}{\sigma_{T_D}} \tag{12}$$

where $\hat{A}$ and $\hat{I}_D$ denote the normalized nucleus area and normalized DAPI intensity, respectively, (both normalized per image).

Finally, the SVM was trained and its performance evaluated in two input spaces:

- SVM in the non-normalized input space (proposed in [1]), i.e., in the 2D input space (A, T_D), this is denoted as SVM_A.
- SVM in the normalized input space (proposed in this work), i.e., in the 2D input space ($\hat{A}$, $\hat{I}_D$), this is denoted as SVM_B.

A five-fold cross-validation was performed for both SVM_A and SVM_B. Furthermore, for SVM_B a leave-one-experiment-out cross-validation was performed.

The overall methodology of the proposed approach is summarized in fig. 7. First the FUCCI images are preprocessed and cell nuclei segmentation is performed based on deep learning. Thereafter, on one hand, features are extracted from the FUCCI images to generate the ground truth labels to train the classifier. On the other hand, based on the segmentation masks generated for the FUCCI images, features are extracted from the corresponding DAPI images (total intensity, T_D) and from the corresponding masks (area, A). These features and the corresponding ground truth labels are used to train the SVM_A. Moreover, for each nucleus, normalized area ($\hat{A}$) and normalized intensity ($\hat{I}_D$) are calculated from the features extracted previously. These normalized features and the corresponding ground truth labels are used to train the SVM_B.

[2] μ_A and σ_A denote the mean and the standard deviation of the area, respectively, of all nuclei belonging to a given image.

[3] μ_{T_D} and σ_{T_D} denote the mean and the standard deviation of the total DAPI intensity, respectively, of all nuclei belonging to a given image.

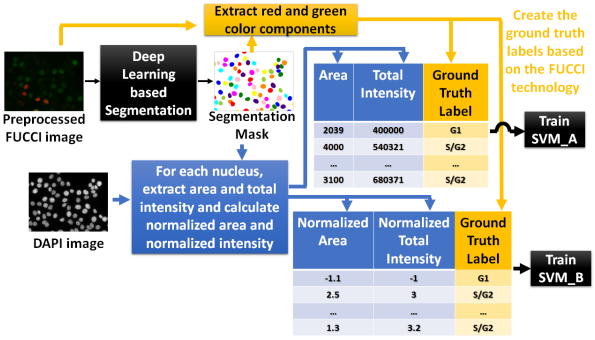


Figure 7: Overall pipeline to train classifiers for cell cycle staging.

4.6. Evaluation Criteria

To test the performance of the proposed nuclei segmentation model, the F1 score was calculated (see eq. 16) at different thresholds of the intersection over union (IoU). The IoU between two objects is given by:

$$IoU = \frac{Area\ of\ overlap}{Area\ of\ union} \tag{13}$$

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. 13) between object i in the ground truth and object j in the prediction mask. 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. 14) and recall (eq. 15). Where TP, FP and FN stand for true positives, false positives and false negatives, respectively. On 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 with all nuclei in the ground truth 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).

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

$$Recall = \frac{TP}{FN + TP} \tag{15}$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{16}$$

To test the performance of the cell cycle staging classifier, the classification_report tool from sklearn.metrics was used. This tool gives the precision (eq. 14) , recall (eq. 15) and F1 score (eq. 16) for each class in the classification problem. It outputs the matrix represented in table 3. The second, third and fourth row represent the precision, recall and F1 score when class G1 is considered positive, when class S/G2 is considered positive and the average between both classes, respectively.

Table 3: Classification results obtained by using the classification_report tool from sklearn.metrics.

	Precision	Recall	F1-Score
G1	G1_Precision	G1_Recall	G1_F1_Score
S/G2	S/G2_Precision	S/G2_Recall	S/G2_F1_Score
Average	Average_Precision	Average_Recall	Average_F1_Score

5. Results and Discussion

This section presents the results and discussion regarding cell nuclei segmentation and cell cycle staging.

5.1. Cell Nuclei Segmentation

First, results regarding training and testing models with DAPI images are presented. Figure 8 shows a visual comparison between different models regarding nuclei segmentation performed for the image shown in fig. 8a.

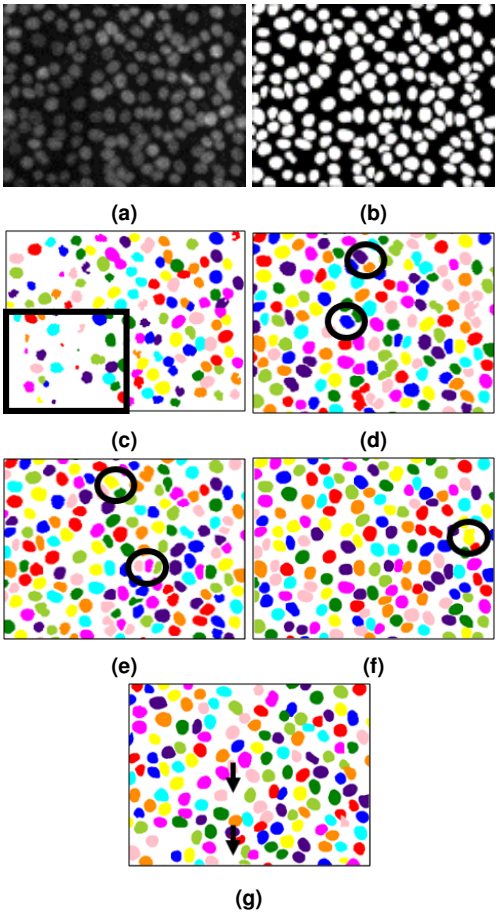


Figure 8: Nuclei segmentation results obtained by applying different methods. a) Input image. b) Corresponding ground truth mask. c), d), e), f) and g) Segmentation masks obtained by applying Yen + watershed, original U-Net, Kaggle_2018, Mask R-CNN, and the proposed approach, respectively.

Figure 8c shows the segmentation mask ob-

tained with Yen + watershed. In the input image (fig. 8a) there is high intensity variation, therefore the classical method struggles in detecting nuclei located in the lower left corner, indicated with a black square.

Results regarding original U-Net (fig. 8d) and Kaggle_2018 (fig. 8e) show that although these segmentation masks are better than the one obtained with Yen + watershed, in some cases there are gaps between close nuclei (highlighted with black circles). On one hand, this can be explained by the fact the ground truth data also has some gaps between touching nuclei. On the other hand, U-Net is designed for semantic segmentation. Thus, it makes the classification at a pixel level, that's why the nuclei boundaries obtained with the original U-Net and Kaggle_2018 aren't that accurate.

By comparing the segmentation masks obtained with Mask R-CNN (fig. 8f) and with the proposed approach (fig. 8g), it is observed that the proposed approach fails to identify some of the occluded or very close nuclei (indicated by black arrows in fig. 8g). This is due to the detection performance of Fast YOLO which is worse than that of the Mask R-CNN. Moreover, the Mask R-CNN presents a merge (highlighted with a black circle). Despite these missdetections and merge, Mask R-CNN and the proposed approach are the models that present the best results, because these models are designed for instance segmentation.

Figure 9 shows a plot of average F1 score across increasing thresholds of IoU. By comparing the deep learning approaches with the traditional method (Yen + watershed), it can be concluded that deep learning models significantly outperform this classical method. Additionally, for $IoU < 0.75$ the performance of the proposed approach is high (F1 score above 0.9), similar to the performance of Mask R-CNN and better than that of all of the other methods.

The accentuated decrease of the F1 score, at $IoU \approx 0.80$, can be explained by the presence of inaccurate boundaries on the ground truth. For example, since the ground truth masks are binary, in order to separate touching nuclei and to solve the problem as an instance segmentation problem, lines were drawn between touching nuclei and the pixels contained in these lines were considered as belonging to the background, (as observed in fig. 8b).

Regarding computational efficiency the training time and the test time required by all the methods were compared. Training time corresponds to the time a model needs to learn a given task, in this case, the task of nuclei instance segmentation. By observing fig. 10a, it can concluded that Mask R-CNN requires significantly more time to learn the

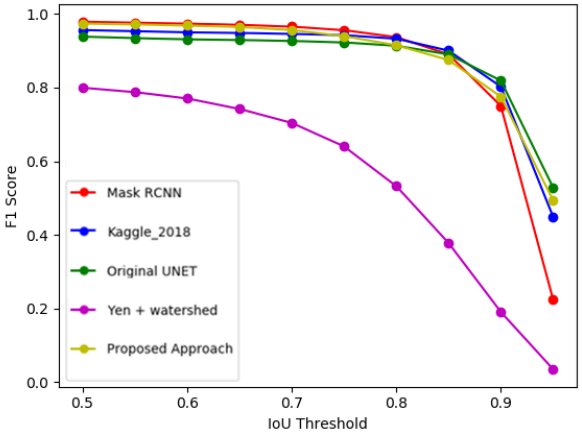


Figure 9: Average F1 score vs IoU threshold, comparison between different models: Yen + watershed (purple), original U-Net (green), Kaggle_2018 (blue), Mask R-CNN (red), proposed approach (yellow).

task of nuclei segmentation (about 1420 minutes), in comparison with the other methods. Although the proposed approach requires more time to train (450 minutes) than the Original U-Net (14 minutes) and Kaggle_2018 (100 minutes), it also provides better segmentation masks as illustrated in fig. 8.

On the other hand, regarding test time, which is the time required for a model to give a segmentation prediction for an image, the results are presented in fig. 10b. These results show that Mask R-CNN is the model that presents the highest test time (15.1 s). The proposed 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 the proposed approach (1.6 s), however Yen + watershed presents the worst performance, as observed in fig. 9.

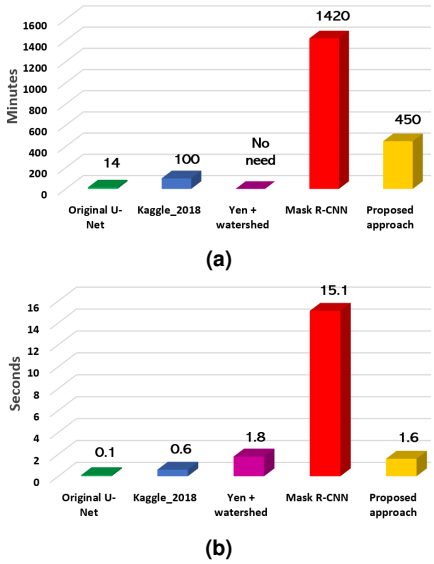


Figure 10: a) Training time (in minutes) associated with each model. **b)** Mean test time per image (in seconds) for each model, for images of size 1388×1040.

To sum up, by training and testing the models in DAPI images it was shown that, when compared to the state-of-the-art method Mask R-CNN, the proposed approach presents a small lost in performance, however it is much faster and computationally more efficient. After showing this, the main goal was to show that the proposed approach for cell nuclei segmentation can be applied for the segmentation of images of cells stained with other dyes. In this work, the FUCCI images were used. Moreover, the segmentation of FUCCI images is really important in this work since the ground truth labels to train the cell cycle staging classifiers will be generated based on the information extracted from the FUCCI images.

Below results regarding segmentation of FUCCI images are presented. Figure 11 shows the F1 Score at different thresholds of the IoU. It can be concluded again that deep learning based approaches perform better than the traditional method. Additionally, the best methods for nuclei segmentation are the Mask R-CNN and the proposed approach, both present a good F1 score (above 0.8) and similar performance for thresholds below 0.75.

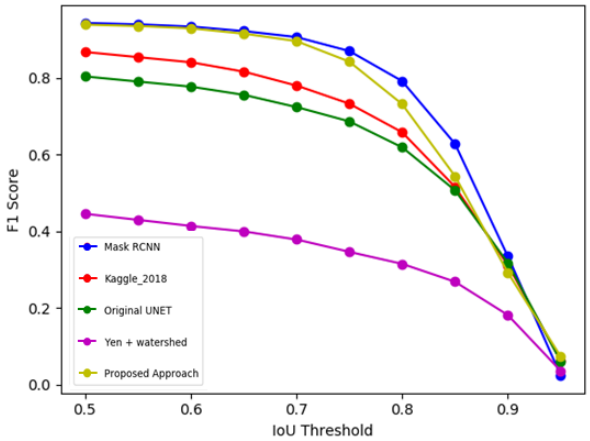


Figure 11: Average F1 Score vs IoU threshold, comparison between different models: Yen + watershed (purple), original U-Net (green), Kaggle_2018 (red), Mask R-CNN (blue) and proposed approach (yellow).

The results in fig. 11 are for the models that were trained and tested with the FUCCI images and with the ground truth masks generated from the DAPI images. Although the nuclei positions and shapes are the same in FUCCI and DAPI images, there are some nuclei in FUCCI images that are colorless (those in G0 and early G1 phases [7]), that is, their green and red intensities are very low. Therefore, the segmentation models may not detect these nuclei that are present in the ground truth masks. This can explain why the results aren't that good as the ones shown in figure 9.

To sum up, it was shown that the proposed method can be used for nuclei segmentation in images of cells stained with different dyes. Moreover, the proposed approach achieves results that are comparable to the state-of-the-art method for instance segmentation (Mask R-CNN). Finally, after an accurate segmentation of the DAPI and FUCCI images the goal was to study the cell cycle.

5.2. Cell Cycle Staging

In this subsection results regarding cell cycle staging are presented.

Comparison with the Labels Given by the Biologist

The ground truth labels generated in this work were compared with the labels given by the biologist (in [1]). In this comparison only the nuclei that were labeled by both were considered, that is, 2681 nuclei. It was observed that only 21 nuclei were labeled differently by both methods. For 2660 nuclei the labels obtained in this work and the ones given by the biologist coincided. To sum up, an automatic method for nuclei labeling based on the FUCCI system was presented. This method can efficiently and accurately replace the intensive work of the biologist.

Nuclei Classification

In this subsection results regarding five-fold cross-validation performed for both SVM_A and SVM_B are presented. Tables 4 and 5 show the classification results obtained with SVM_A and SVM_B respectively. In each table, the results correspond to the mean and standard deviation over the five models obtained by performing five-fold cross-validation.

By comparing the two approaches based on the SVM it is clearly observed that the one that presents better results is the SVM_B. This means that the normalization of the area and intensity, per image, helps in separation between classes G1 and S/G2. These results show that what matters is the relative area and DAPI intensity, since nuclei in phases S/G2 are bigger and present higher DAPI intensity when compared to nuclei in phase G1.

Table 4: Classification results for SVM_A.

	Precision	Recall	F1-Score
G1	0.698 ± 0.019	0.900 ± 0.020	0.786 ± 0.012
S/G2	0.622 ± 0.061	0.294 ± 0.018	0.399 ± 0.024
Avg	0.660 ± 0.032	0.597 ± 0.014	0.592 ± 0.013

Table 5: Classification results for SVM_B.

	Precision	Recall	F1-Score
G1	0.904 ± 0.021	0.926 ± 0.015	0.915 ± 0.006
S/G2	0.858 ± 0.025	0.822 ± 0.040	0.839 ± 0.019
Avg	0.881 ± 0.016	0.874 ± 0.022	0.877 ± 0.010

Ferro et al. [1] applied the modified k-means clustering algorithm per image, whereas in this work the features from all nuclei from all images

were used to train the SVM_A and SVM_B. In [1] a non-normalized input space (area, total intensity) was used, and since there may be variations in the acquisition parameters between different images, the k-means algorithm was applied per image. In this work, on one hand, the SVM_A was trained in the non-normalized input space (area, total intensity) with features from different images and it was verified that the results aren't that good (F1 scores 0.786 ± 0.012 and 0.399 ± 0.024 for class G1 and class S/G2, respectively). These results are expectable because, for instance, the intensities between different images may not be comparable. On the other hand, the SVM_B was trained and tested in the normalized input space with features from different images and provided good results (F1 scores 0.915 ± 0.006 and 0.839 ± 0.019 for class G1 and class S/G2, respectively). These results are better than those obtained with SVM_A, because the F1 score values obtained are higher and the standard deviation across the five folds is lower, that is, the diversity in different folds is lower. This is due to the fact that SVM_B takes into account the relative area and intensity for each nucleus.

To sum up, the SVM_B presented in this work for cell cycle staging is better than the approach presented in [1]. While the method proposed in [1] needs to be applied image by image, which may be computationally expensive, the proposed SVM_B can take as input nuclei features from all images to perform cell cycle staging.

Furthermore, for the SVM_B a leave-one-experiment-out cross-validation was performed. It was observed that the F1 score varies within the set of images and for 21 images out of 130 images the classifier achieved a F1 score of 100 %.

Testing the Classifier on a Different Cell Line

To exclude cell line dependence, the interphase cell cycle classifier was further tested in images from cells belonging to another cell line different from the one that was used to train the system. In particular, the performance of the trained classifier was tested in images of gastric cancer cells stained with DAPI and labeled with a molecular marker for cells in stage G2. The classification results are shown in table 6. SVM_A classified correctly 12 nuclei (recall of 16 %), and SVM_B classified correctly 68 nuclei (recall of 93 %).

Table 6: Classification results for SVM_A and SVM_B in a different cell line.

	TP	FN	Recall
SVM_A	12	61	16 %
SVM_B	68	5	93 %

To sum up, these results show that SVM_B presents good results in this new cell line, emphasizing once again the importance of area and intensity normalization. Due to different acquisition parameters and different size of the cells belonging to different cell lines one can't rely on area and total intensity for cell cycle staging. These results show that the important features for cell cycle staging are the relative area and relative intensity.

6. Conclusions and Future Work

This work addresses the important problem of nuclei segmentation for high throughput applications. A new approach for cell nuclei instance segmentation is presented, it 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 about $9\times$ on the segmentation time was obtained.

Furthermore, in this work the cell cycle was studied, which is a fundamental mechanism of all living organisms and plays an important role in several human diseases. A new supervised approach was presented for cell cycle staging from images of cells stained with DAPI, a commonly used fluorescent dye. The proposed approach can be applied for the cell cycle staging of adherent cells without the need of destroying their natural architecture. Furthermore, it presents good results and a good generalization capability in the normalized input space showing that it is always important to take into account the information regarding the population of cells.

To sum up, the methods proposed in this work will be of great importance for the research line focused on the diagnosis of hereditary gastric cancer from fluorescence microscopy images. Thus, in the future, morphological and textural features will be extracted from the segmented nuclei for diagnosis of pathogenic mutations associated with cancer. Additionally, the SVM trained in the normalized space will be used for cell cycle staging of cells with wild-type proteins and cells with mutations in proteins associated with cancer, to study if there is dysregulation of the cell cycle in cancerous cells.

Acknowledgement This document was written and made publicly available as an institutional academic requirement and as a part of the evaluation of the MSc thesis in Biomedical Engineering of the author at Instituto Superior Técnico. The work described herein was performed at the Institute for Systems and Robotics of Instituto Superior Técnico (Lisbon, Portugal), during the period February-October 2019, under the supervision of Prof. João Sanches, Prof. Margarida Silveira, Doctor Raquel Seruca and Doctor Sofia Fernandes, and within the frame of the FCT (Fundação para a Ciência e Tecnologia) through the projects TRACE (PTDC/BBB-IMG/0283/2014) and reference UID/EEA/50009/2019.

References

[1] Anabela Ferro, Tânia Mestre, Patrícia Carneiro, Ivan Sahumbaiev, Raquel Seruca, and João M Sanches. Blue intensity matters for cell cycle profiling in fluorescence DAPI-stained images. *Laboratory Investigation*, 97(5):615, 2017.

[2] Humayun Irshad, Antoine Veillard, Ludovic Roux, and Daniel Racoceanu. Methods for nuclei detection, segmentation, and classification in digital histopathology: a review-current status and future potential. *IEEE reviews in biomedical engineering*, 7:97–114, 2014.

[3] Fuyong Xing and Lin Yang. Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: a comprehensive review. *IEEE reviews in biomedical engineering*, 9:234–263, 2016.

[4] Vassilis Roukos, Gianluca Pegoraro, Ty C Voss, and Tom Misteli. Cell cycle staging of individual cells by fluorescence microscopy. *Nature protocols*, 10(2):334, 2015.

[5] Tânia Mestre, Joana Figueiredo, Ana Sofia Ribeiro, Joana Paredes, Raquel Seruca, and João Miguel Sanches. Quantification of topological features in cell meshes to explore e-cadherin dysfunction. *Scientific reports*, 6:25101, 2016.

[6] Bhupendra S Deshmukh and Vijay H Mankar. Segmentation of microscopic images: A survey. In *2014 International Conference on Electronic Systems, Signal Processing and Computing Technologies*, pages 362–364. IEEE, 2014.

[7] Raquel Seruca, Jasjit S Suri, and J Miquel Sanches. *Fluorescence imaging and biological quantification*. CRC Press, 2017.

[8] Geoffrey M Cooper, Robert E Hausman, and Robert E Hausman. *The cell: a molecular approach*, volume 10. ASM press Washington, DC, 2000.

[9] Hemaxi Narotamo, J Miguel Sanches, and Margarida Silveira. Segmentation of cell nuclei in fluorescence microscopy images using deep learning. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 53–64. Springer, 2019.

[10] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[13] Juan C Caicedo, Jonathan Roth, Allen Goodman, Tim Becker, Kyle W Karhohs, Matthieu Broisin, Molnar Csaba, Claire McQuin, Shantanu Singh, Fabian Theis, et al. Evaluation of deep learning strategies for nucleus segmentation in fluorescence images. *BioRxiv*, page 335216, 2019.

[14] Yuxin Cui, Guiying Zhang, Zhonghao Liu, Zheng Xiong, and Jianjun Hu. A deep learning algorithm for one-step contour aware nuclei segmentation of histopathological images. *arXiv preprint arXiv:1803.02786*, 2018.

[15] Find the nuclei in divergent images to advance medical discovery. `https://www.kaggle.com/c/data-science-bowl-2018/discussion`. Accessed: 2019-02-20.

[16] [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: 2019-02-20.

[17] Fidel A Guerrero-Pena, Pedro D Marrero Fernandez, Tsang Ing Ren, Mary Yui, Ellen Rothenberg, and Alexandre Cunha. Multiclass weighted loss for instance segmentation of cluttered cells. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2451–2455. IEEE, 2018.

[18] Aarno Oskar Vuola, Saad Ullah Akram, and Juho Kannala. Mask-RCNN and U-net ensembled for nuclei segmentation. *arXiv preprint arXiv:1901.10170*, 2019.

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[20] Jeremiah W Johnson. Adapting Mask-RCNN for automatic nucleus segmentation. *arXiv preprint arXiv:1805.00500*, 2018.

[21] Xiaowei Chen, Xiaobo Zhou, and Stephen TC Wong. Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy. *IEEE Transactions on Biomedical Engineering*, 53(4):762–766, 2006.

[22] Jun Yan, Xiaobo Zhou, Qiong Yang, Ning Liu, Qiansheng Cheng, and Stephen TC Wong. An effective system for optical microscopy cell image segmentation, tracking and cell phase identification. In *2006 International Conference on Image Processing*, pages 1917–1920. IEEE, 2006.

[23] Meng Wang, Xiaobo Zhou, Fuhai Li, Jeremy Huckins, Randall W King, and Stephen TC Wong. Novel cell segmentation and online SVM for cell cycle phase identification in automated microscopy. *Bioinformatics*, 24(1):94–101, 2007.

[24] Thomas Blasi, Holger Hennig, Huw D Summers, Fabian J Theis, Joana Cerveira, James O Patterson, Derek Davies, Andrew Filby, Anne E Carpenter, and Paul Rees. Label-free cell cycle analysis for high-throughput imaging flow cytometry. *Nature communications*, 7:10256, 2016.

[25] Philipp Eulenberg, Niklas Köhler, Thomas Blasi, Andrew Filby, Anne E Carpenter, Paul Rees, Fabian J Theis, and F Alexander Wolf. Reconstructing cell cycle and disease progression using deep learning. *Nature communications*, 8(1):463, 2017.

[26] WL Godfrey, DM Hill, JA Kilgore, GM Buller, JA Bradford, DR Gray, I Clements, K Oakleaf, JJ Salisbury, MJ Ignatius, et al. Complementarity of flow cytometry and fluorescence microscopy. *Microscopy and Microanalysis*, 11(S02):246–247, 2005.

[27] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[28] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[29] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[30] François Chollet et al. Keras. `https://keras.io`, 2015.

[31] Darkflow. `https://github.com/thtrieu/darkflow`, 2018.

[32] Jui-Cheng Yen, Fu-Juay Chang, and Shyang Chang. A new criterion for automatic multilevel thresholding. *IEEE Transactions on Image Processing*, 4(3):370–378, 1995.

[33] Asako Sakaue-Sawano, Hiroshi Kurokawa, Toshifumi Morimura, Aki Hanyu, Hiroshi Hama, Hatsuki Osawa, Saori Kashiwagi, Kiyoko Fukami, Takaki Miyata, Hiroyuki Miyoshi, et al. Visualizing spatiotemporal dynamics of multicellular cell-cycle progression. *Cell*, 132(3):487–498, 2008.