INSTITUTE OF COMPUTER SCIENCES
Master in Artificial Intelligence and Data Science

Universitätsstr. 1 D–40225 Düsseldorf

Heinrich Heine Universität Düsseldorf

AI-based fluorescent labeling for cell line development

Hanna Pankova

Master thesis

Date of issue: 01. April 2022
Date of submission: 29. August 2022
Reviewers: Prof. Dr. Markus Kollmann
Dr. Wolfgang Halter

Erklärung

Hiermit versichere ich, dass ich diese Master thesis selbstständig verfasst habe. Ich habe dazu keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Düsseldorf, den 29. August 2022

Hanna Pankova

Abstract

Cell line development is an expensive and time-consuming process, however that is the most modern approach for producing the proteins needed in pharmaceuticals.

Contents

1 Introduction **1**
1.1 Motivation 1
1.2 Notation 1

2 Background **2**
2.1 Biology 2
2.2 Imaging 3
2.3 ML 4

3 Prediction of cell organelles **5**
3.1 Overfitting 5

4 Crops combination **9**

5 Predicting cell organelles **10**
5.1 Nuclei 10

6 Downstream metrics **11**
6.1 Nuclei segmentation 11

7 Summary **17**

List of Figures **18**

List of Tables **18**

1 Introduction

1.1 Motivation

1.2 Notation

2 Background

2.1 Biology

Cell line development is a process of generating single cell-derived clones that produce high and consistent levels of target therapeutic protein. (pharma.lonza.com/offerings/mammalian/cell-line-development)

Differential interference contrast (DIC) microscopy is an optical microscopy technique used to enhance the contrast in unstained, transparent samples (wikipedia).

Proteins are large biomolecules and macromolecules that comprise one or more long chains of amino acid residues (https://en.wikipedia.org/wiki/Protein).

Fluorescent labelling is the process of covalently binding fluorescent dyes to biomolecules such as nucleic acids or proteins so that they can be visualized by fluorescence imaging (https://www.nature.com/subjects/fluorescent-labelling). A fluorophore is a chemical compound that can reemit light at a certain wavelength.These compounds are a critical tool in biology because they allow experimentalists toimage particular components of a given cell in detail. (O'reilly life sciences p113)

Cell line development (CLD) is the process by which the cellular machinery is co-opted to manufacture therapeutic biologics or other proteins of interest. One can use different expression systems for cell line development: bacterial, plant-based, yeast, mammalian. (copy paste from https://www.beckman.de/resources/product-applications/lead-optimization/cell-line-development) Chinese hamster ovary (CHO) cells are the most popular mammalian cells used for protein production. (doi:10.1016/B978-0-08-100623-8.00007-4)

First step of CLD is the introduction of the gene of interest (GOI or a DNA vector) to CHO cells. This process is called a transfection. It is important to transfect a GOI into an optimal site of genome to secure a high protein expression over time during protein production, however pratically transfection mostly results in a vector being inserted into a random site within a host cell genome. In case the gene was transfected in the inactive site of genome (and the majority of genome is not transriptionally active) the cell will likely not express the gene. (doi:10.1016/B978-0-08-100623-8.00007-4) (doi:10.1016/j.coche.2018.08.002)

The second step is the selection of cell pools that have successful and stable gene integrations. The reasond why not all of them are suitable for cloning is the following: during the transfection only 80% of cells recieve the vector of GOI (doi:10.1016/B978-0-08-100623-8.00007-4), only the small percent of which actually integrate a vector into the genome and, as mentioned above, only a fraction of those cells are able to stabily express a protein. (Reference needed). Such selection could be done with bulk sorting algorithm. (doi:10.1016/B978-0-08-100623-8.00007-4)

The third step in CLD is to clone the cells. The chosen stable pools of cells are phenotypically and genetically diverse - meaning they have different growth rates,

metabolic profile and etc. This is not ideal for industrial production - all the cells used for protein production should be derived from the same clone ([25] here doi:10.1016/B978-0-08-100623-8.00007-4). In order to choose single best cells for further cloning one asseses several parameters like cell size, granularity, cell surface protein expression and etc. This can be done with Fluorescent Activated Cell Sorting (FACS) technology. (https://doi.org/10.1517/14712598.4.11.1821). Unfortunately fluorescence labeling is expensive and may ruin the cell due to its phototoxicity (https://doi.org/10.1371/journal.pone.0007497). There is a limited number of available fluorescent channels in microscopes as well as such labels can also be inconsistent, depend a lot on reagent quality, and require many hours of lab work. Therefore there exists a need for flurescent labeling in silico - without intervening into the cell.

Once the cells are cloned, phenotypical and genetical heterogeneity is reduced, the next step is to charaterize clonally-derived cells based on the following criteria: cell size, growth rate, protein quality, titer, metabolities and etc. With this one can estimate clones productivity and titer. Such observations may take up to 90 days after which one can determine which cells are stable and therefore suitable for production. This is the last step of CLD process and consumes a lot of time and maintaining costs for feeding and cloning the cells. Predicting the stability of the cells directly from DIC images would reduce this time significantly allowing to escape this process completely.

However there are also some disadtantages of this approach. First, it can be less accurate than skilled cells staining perfomed manually. Extreme or unsual clones and phenotypes might be challenging if they were not used in the training set of images.

2.2 Imaging

The microscope used in the experiments takes photos of the well plate in random locations. The reason for that hides in the focusing problem, to get a reasonably good photo without blur it has to focus on a specific location, this is done there automatically, therefore the location of the focus is almost random (see Figure 1).

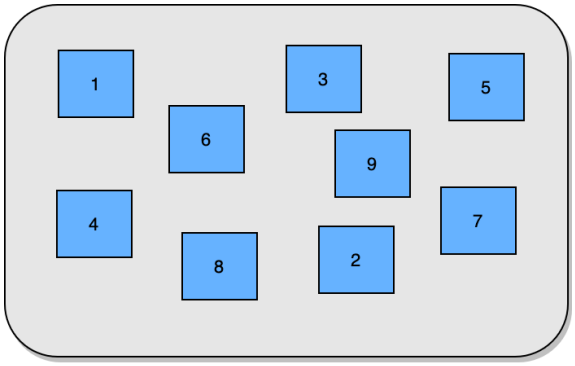


Figure 1: Way in which photos of the well-plate were taken

Although it might be problematic in the following sense: photos takes by the microscope in such manner do not gurantee that the focus will land in distinct spots all the time. This means that some cells taken during one of the photos might appear in the later ones. Since the photos have a high-resolution the crops are first performed and it might happen that same cell might appear in several crops. Afterwards, when crops are split between train, test and validation datasets it might happen that the same cell will once land in the train set and another time in the validation set, which will lead to a not completely fair and representative validaiton loss during training.

2.3 ML

Background on Unet and ML in general

Convoutional neural network is a neural network that is based on convolutional layers. It is a powerful tool for image processing and is used in medical imaging. Convolution is a linear operation used in convolutional layers that can be performed by applying a kernel (a 2d matrix) across a bigger input matrix called tensor, which can be 3d. Element-wise product between them is calculated and summed, this value will be an element of the output 2d matrix. Kernel slides accross all locations of the input tensor. In case if several different kernels were used then a 3d tensor will be created.

Main advantage of convolutional neural networks if weight sharing. Kernel has learnable weights however these weights are shared across all locations of the kernel on the input tensor, thic strongly reduces the number of parameters needed.

CNNs also uses non-linearities like RELU, ELU, Tahn, Sigmoids and etc. They are also often combined with max pooling layers and dropouts to escape overfitting.

Overfitting is one of the most often problems in deep learning that prevents model to generalize well for unseen data. This can happen when the model is too big for the amount of training data given, it was not regularized well or there is just not enough data for training.

U-Net architecture is widely used for segmentation purposes. It is a convolutional neural network with the following architecture: [img] . It first performs image downsampling and upsampling afterwards. [https://arxiv.org/pdf/1505.04597.pdf] The following architecture have been used for nuclei prediction.

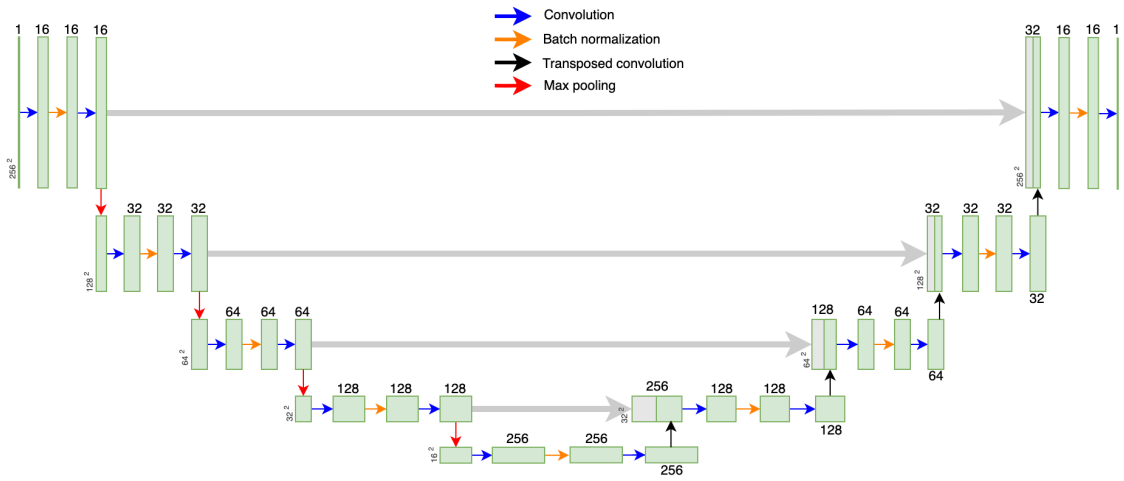


Figure 2: Unet

3 Prediction of cell organelles

3.1 Overfitting

Definition 3.1 (Overfitting)**.** "Hypothesis overfits the training samples if some other hypothesis that fits the training samples less well actually performs better over the entire distribution of instances" (p67 Mitchell Machine Learning 1997).

Overfiting prevents model to generalize well on the unseen data and in order to avoid fitting to closely to the training dataset one has several options:

(Bishop book)

3.1.1 Early-stopping

Overfitting effect occurs at later epochs. This doesn't happen during early epochs as with the correct weights initialization the weights of the model are quite small and random and therefore the best decision surface would be a smooth one. But in the later epochs the difference in values of the weights grows and they become not similar anymore which means also that the decision surface becomes more complex and will be ble to fit not only the training data itself, but also its noise. (p111 Mitchell Machine Learning 1997). And that is why stopping before the model became too complex for the given data may mitigate this problem.

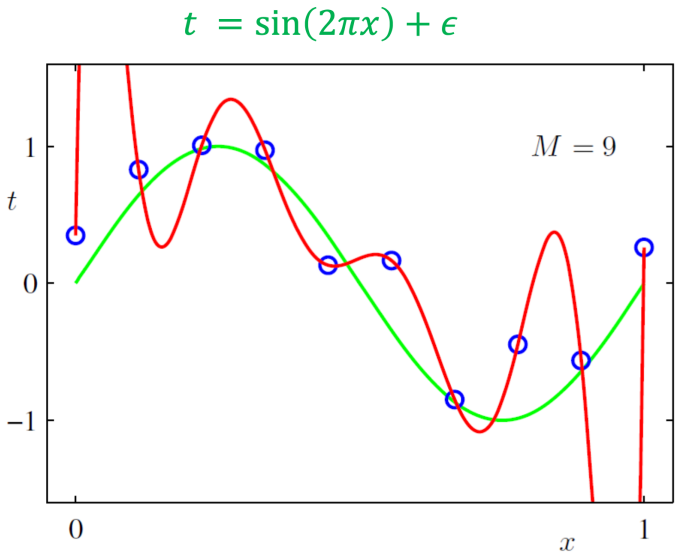


Figure 3: Overfitting

3.1.2 Regularization

The complexity of the model grows with the number of features it uses, sometimes the model may pay attention to the features that are not important to the outcome, or even considers a noise to be a feature. To prevent this one should decrease the weights associated with useless features, however we cannot know ahead which of them should be ignored, therefore one may limit them all. (doi:10.1088/1742-6596/1168/2/022022) In order to do that, a penalty term in loss function is added:

$$\tilde{L}(\theta, X, y) = L(\theta, X, y) + \lambda R(\theta) \tag{1}$$

for some $\lambda > 0$. This is called a *soft-constrait* optimization. When $R(\theta)$ is of the form $R(\theta) = ||\theta||_2^2 = \sqrt{\sum_i \theta_i^2}$ this is called *L2-regularization* and when it is of form $R(\theta) = |\theta||_1 = \sum_i |\theta_i|$ this is called *L1-regularization*. *L2-regularization* used in combination with backpropagation is equivalent to weight decay. Weight decay is defined by Hanson and Pratt (1988) as follows:

$$\theta_{t+1} = (1 - \lambda)\theta_t - \alpha \frac{\partial L}{\partial \theta_t} \tag{2}$$

where α is a learning rate. Weight decay successfully affects more those weights the gradient change along which is smaller (Goodfellow Deep learning p229). *L1-regularization* induces sparsity of the weights by assining some of them to zero, this could be also considered as feature selection approach.

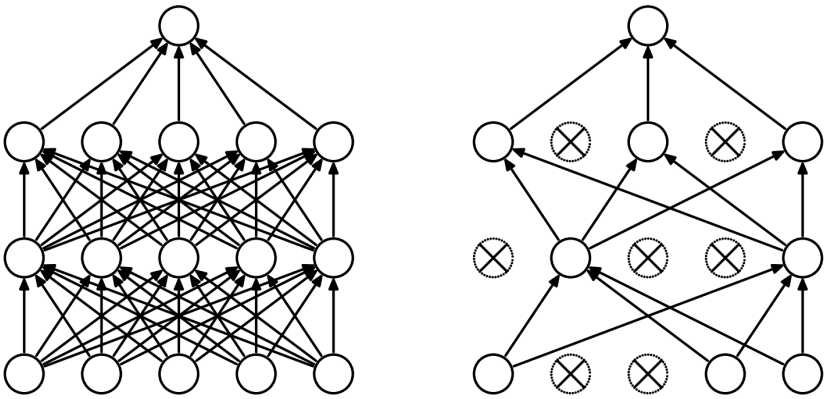
Regularization techniques like BatchNorm and Dropout could also be applied. Batch-

Norm is defined by :

$$
\begin{aligned}
y_i &= \gamma \frac{x_i - \mu_B}{\sigma_B^2 + \epsilon} + \beta \\
\sigma_B^2 &= \frac{1}{m} \sum_i^m (x_i - \mu_B)^2 \\
\mu_B &= \frac{1}{m} \sum_i^m x_i
\end{aligned}
\tag{3}
$$

where m is a batch size. Ioffe and Szegedy, 2015

Dropout is a technique that randomly sets some of the weights to zero. (Srivastava, Hinton 2014).



(a) Standard Neural Net (b) After applying dropout.

Figure 4: Dropout

From the first training one can clearly see that overfit happens around epoch 30. Although one could just pick out one of epochs before the 30th one (before overfit has happened), applying more regularizion to the model that has been using dropout only would be a good idea. Early stopping in combination with weight decay, BatchNorm were used to regularize the model that was overfitting too quickly with dropout only. $BatchNorm$ layers have been added after the first Convolution layer in each ConvBlock and $TransposedConvBlock$. The results of training the regularized network is presented in Figure 6.

3.1.3 Network reduction

Since learning a too complex and noise-fitting decision surface might be an often cause of an overfit, another way to mitigate it would to be reduce the space of the possible decision surfaces and therefore make the surface simpler so that it cannot fit into the noise from the data. By changing the number of adaptive parameters in the network, the complexity can be varied. (cite Page 332, Neural Networks for Pattern Recognition, 1995.)

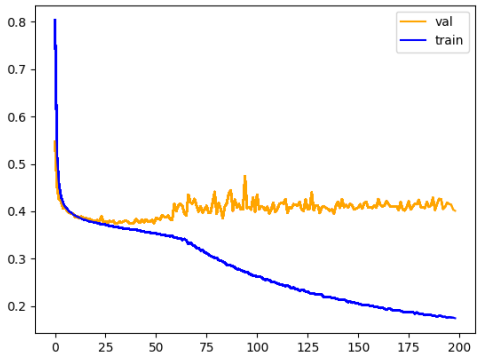


Figure 5: Not regularized

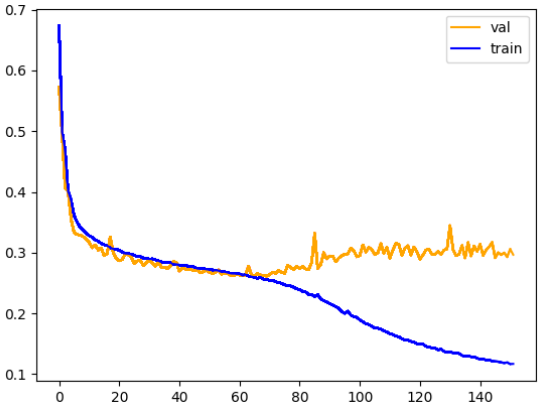


Figure 6: Regularized

3.1.4 Expansion of the training data

To well-tune the hyperparameters the model needs to have a sufficient amount of quality samples. An expanded dataset can improve the quality of the predictions, (cite doi:10.1088/1742-6596/1168/2/022022) however only when the model has already performed well on the initial dataset. If the model performing bad initially, adding more data will not solve the problem. Here having TODO n samples of data the model was trained on TODO samples only to find the best structure and regularization first, afterwards the model was retrained using more data and the quality imroved from, to TODO.

4 Crops combination

Since the neural network has been constucted in a way that the input size is 256 by 256 pixels, one cannot feed a bigger image in there. Therefore a high-resolution image has to first split into several crops which are then get recombined. There are several ways of how one can split the image, the easiest approach would be to use a sliding window. With the step size of s and the window size of w the sliding window is defined as:

When step size s is equal to window size w, there is no overlap between the windows. One very important insight from prediction results if that the model is less accurate on the borders of the image, rather than of cropping itself. Images are cropped consequently and most the times there are cells on the borders of the crops that get sliced and it might be impossible the make a good prediction for them just due to the lack of the information. Therefore the step size has to be smaller than the window size, so that the windows are overlapping and for each prediction we use only the image center and are allowed to ingore predictions on the border.

Such approach would help to reduce the effect of the grid visible on the image composed of many small crops, which one can see in the Figure 7 to almost non-visible borders as in the Figure 8. This would of course take longer time in predictions, however as the speed is less crucial in comparison to the accuracy of the predictions.

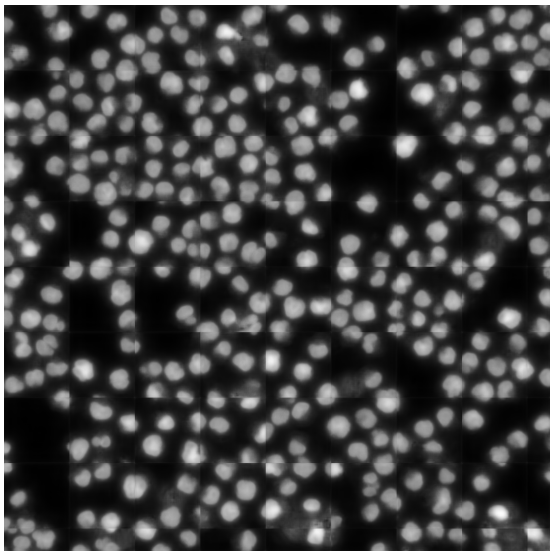


Figure 7: No overlap

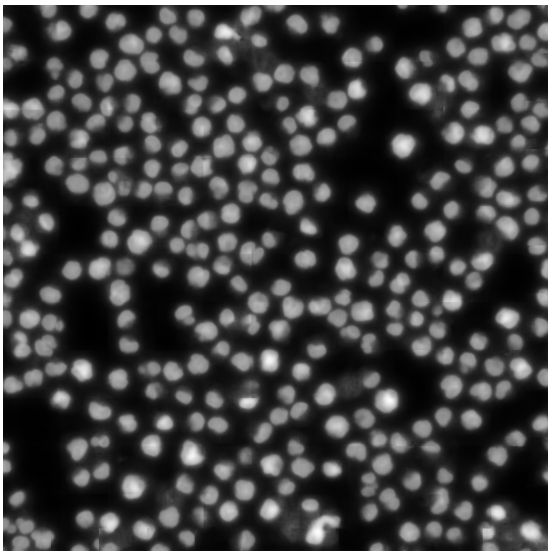


Figure 8: 30 pixels overlap

5 Predicting cell organelles

5.1 Nuclei

In this subsection the results of the nuclei predictions will be presented. You can see examples of predictions presented in Figures TODO and TODO. There are TODO visible problems with the predictions:

- The form of the nucleus is captured, but the texture inside is not
- Blurry border aorund the nuclei
- Problems with the predictions on the borders of the crop

Predictions of the border of the crops are quite challenging for the model as there is not enough of the information due to the cropped parts of the cell. However this can be easily solved with using overlaps while cropping the image to avoid the use of the pixels predicted on the border. There more information on this in Chapter TODO (Crops combination).

Explanation for the lack of details inside TODO

The nucleus is relatively a simple task for predictions as it might be relatively easier to spot the nuclei manually as well. They are relatively big with respect to the cell as well. More challenging task would be a prediction of Golgi Apparatus.

6 Downstream metrics

The evaluation of the model in terms of the loss is not quite objective for the current problem. Even when one model might have a smaller loss, it doesn't nessecarily will perform better than another model. (cite Cohen) Therefore the evaluation of the model is done in terms of the metrics that are defined as follows. The most used ones by the clients were

- Number of nuclei
- The relatuve area of the nuclei
- Total intensity
- Mean intensity

6.1 Nuclei segmentation

6.1.1 Challenge

To properly evaluate these metrics on model predictions, post-processing first for segmenting the nuclei is needed. This is not a very straight-forward task to do as there are different edge cases where the nuclei are difficult to segment due to the variety of different factors.

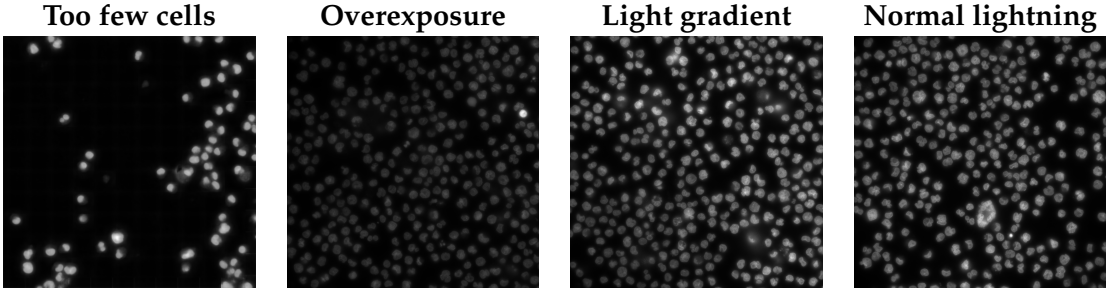


Figure 9: Different lightning conditions

For example, different brightness of the images that comes from different exposure during the photo taking process might make the nuclei segmentation more challenging. The same goes for the different lighting conditions. Different lighning conditions are presented in Figure 9. The following inconsistenties in lightning conditions are presented (from left to right): image contains too few cells, which leads to background being much darker than usually; overexposure of one cell, which leads to difficulties of segmenting the rest of the cells as they are hard distinguishable from the background; lighting gradient from darker (left bottom corner) to brighter (upper right corner) region; normal lighting conditions.

Another challenge for segmentation bring nuclei that are very close to each other. This might happen sometimes because some of the cells are currently in the process of the division. Also when some have already fully divided, they might still be located close to one another. The example of such situation is presented in Figure 10.

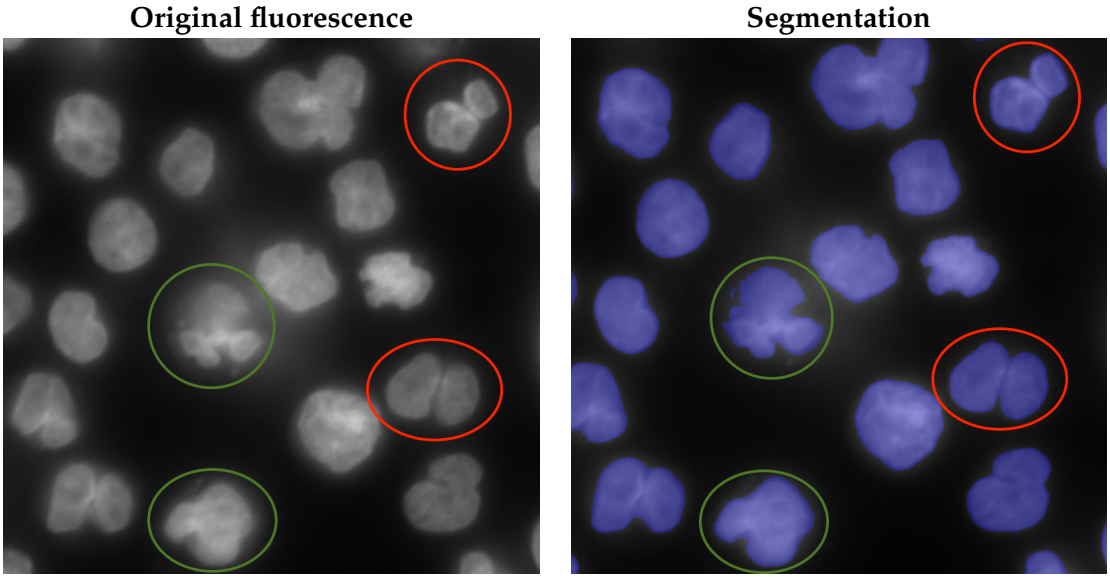


Figure 10: Closely located cells

Here cells, that are not yet fully divided are highlighted with the green cirles, and ones, that are fully divided, but just located too close to one another are highlighted with red circles. You can see that the segmentation algorithm (described in TODO) recognises both such cases as one cell.

6.1.2 Algorithm

Global thresholding is a an algorithm that simply choses one threshold T for the whole histogram of the image. All pixels that are smaller than this threshold $x_i < T$ are assigned to be of class 0 (background) and all pixels that are larger than this threshold $x_i > T$ are assigned to be of class 1 (foreground). To find a threshold automatically Gonzalez et al. (cite Digital Image Processing (2nd Edition)) proposed the following algorithm:

Algorithm 1 Global thresholding

1. Select an initial estimate for T.
2. Segment the image using T. This will produce 2 groups of pixels G_1 (all pixels $x_i > T$) and G_2 (all pixels $x_i < T$).
3. Computer the average gray values μ_1 and μ_2 for the pixels in regions G_1 and G_2.
4. Compute a new threshold value $T' = \frac{\mu_1 + \mu_2}{2}$
5. Repeat steps 2-4 until difference in the change of value T is smaller that a predefined parameter.

There are different ways of how one can define such initial threshold. There is also no single best solution for all of the cases. For example, when one has an assumption that the foreground occupies approximately the same area as the background, than initial threshold T should be chosen to be an average gray level. In this case global thresholding did not perfom well due to different intensities in different regions of the images (non-uniform illumination). Distribution of the cells also varies through the dataset and some images contain more cells, while other contain only few.

In order to segment the background from the foreground (nuclei), the following function was used:

```
skimage.filters.local_threshold(img, block_size=7,
   method='gaussian', offset=0)
```

This is where basic adaptive thresholding or local thresholding comes in handy. The advantage of this method hides in the fact, that it doesn't compute a threshold based on the full histogram of the image, but uses parts of it to compute different thresholds for different subregions of the image. This method is also known as adaptive or dynamic thresholding. The threshold value is the weighted mean for the local neighborhood of a pixel subtracted by a constant. (cite Digital Image Processing (2nd Edition)). With the image size of 2136x2136, the local neighborhood or a *block size* was chosen equal to 111 by experimenting with different values. The default method used on for local thresholding is *gaussian*. *offset* valut is a constant that will be subtracted from weighted mean of neighborhood during the calculation of the local threshold, by default this value is 0. (cite skimage)

Let z be a random variable that quantifies a gray-level value of the pixel, then the histogram of the image is a probability density function (PDF) $p(z)$. Since we assume that the image contains a background and a foreground, then this PDF is a mixture of two densities $p_1(z)$ and $p_@(z)$ weighted by the relative areas of these two classes (their number of pixels) P_1 and P_2. Then

$$p(z) = P_1 p_1(z) + P_2 p_2(z) \tag{4}$$

By assuming Gauassian model for both $p_1(z)$ and $p_2(z)$, one gets a Gaussian Mixture Model (GMM). Here since we assume that each pixel can be asssigned to background of foreground only, we have $P_1 + P_2 = 1$.

Probability to falsely classify an background pixel as object then is:

$$E_1(T) = \int_{-\infty}^{T} p_2(z)\, dz \tag{5}$$

And probability to falsely classify an object pixel as background then is:

$$E_2(T) = \int_{T}^{+\infty} p_1(z)\, dz \tag{6}$$

The overall error is:

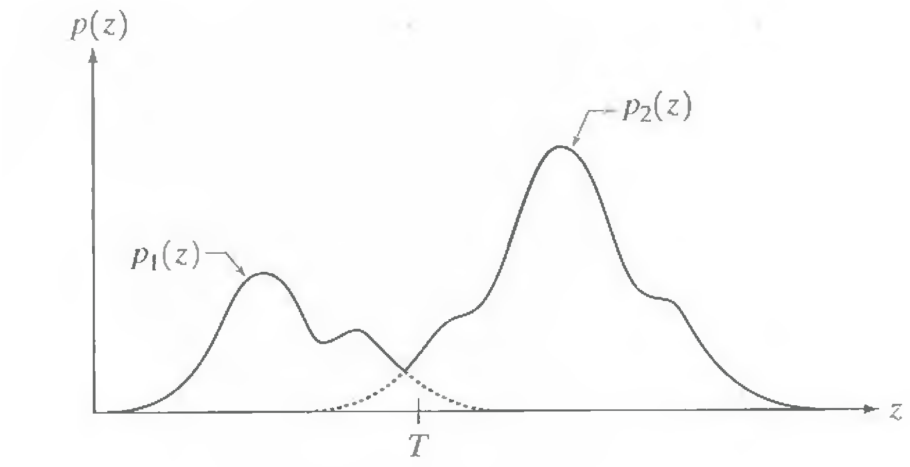


Figure 11: Histogram as a probability density function

$$E(T) = P_1 E_1(T) + P_2 E_2(T) \tag{7}$$

By differentiating $E(T)$ wrt to T and equating the result to zero the optmal equation will be:

$$P_1 p_1(T) = P_2 p_2(T) \tag{8}$$

Since Gaussian distributions have been assumed then:

$$p(z) = \frac{P_1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(z-\mu_1)^2}{2\sigma_1^2}} + \frac{P_2}{\sqrt{2\pi}\sigma_2} e^{-\frac{(z-\mu_2)^2}{2\sigma_2^2}} \tag{9}$$

With μ_i and σ_i^2 for $i \in \{1, 2\}$ are the mean and variance of the Gaussian distribution $p_i(z)$. This results in the following solution for T:

$$AT^2 + BT + C = 0 \tag{10}$$

where

$$\begin{aligned} A &= \sigma_1^2 + \sigma_2^2 \\ B &= 2(\mu_1\sigma_1^2 - \mu_2\sigma_2^2) \\ C &= \sigma_1^2\mu_2^2 - \sigma_2^2\mu_1^2 + 2\sigma_1^2 2\sigma_2^2 ln\left(\frac{\sigma_2 P_1}{\sigma_1 P_2}\right) \end{aligned} \tag{11}$$

To escape two optimal solutions of the quadratic equation it may be assumed that $\sigma_1 = \sigma_2 = \sigma$ and then:

Local Threshold	Global Threshold
0.3 sec	17 sec

Table 1: Threshold timing

$$T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} ln\left(\frac{P_2}{P_1}\right) \tag{12}$$

Such threshold search is then applied to all of the subregions of the image with overlaps. Threshold are calculated only for the regions that contain two peaks and interpolated to the other pixels from the regions that do not contain clear two peaks in their histograms. If the subregions doesn't contain two peaks, it simply means that there is no foreground or background object on it.

Of course local thresholding approach takes a longer time: TODO table with timing. Therefore there as an alternative one can use *minimum thresholding*. This is a global thresholding approach which performs visually a bit worse that a local threshold, however it is much faster (see Table 1).

Comparison of the prdections in difficult lightning conditions of minimum thresholding and local adaptive thresholding are presented in the Figure TODO. In extreme cases of difficult lighting conditions the local adaptive thresholding is much better than the minimum thresholding, however on the images of the better quality TODO (see Figure 13) minimum thresholding might successfully sibstitute the local adaptive thresholding when the time of processing is crucial.

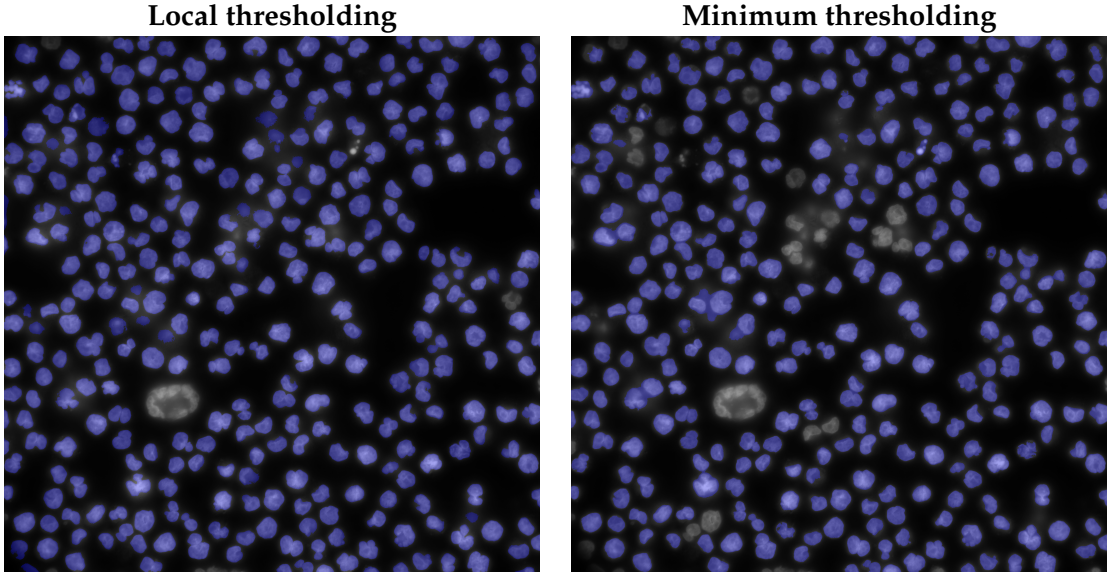
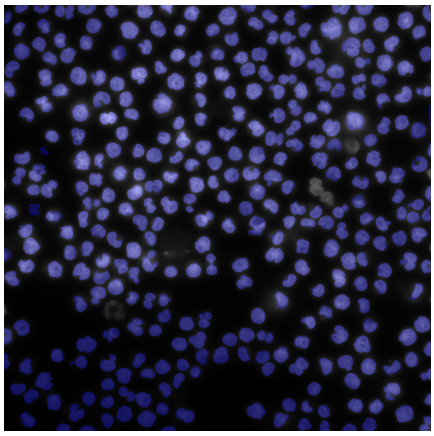
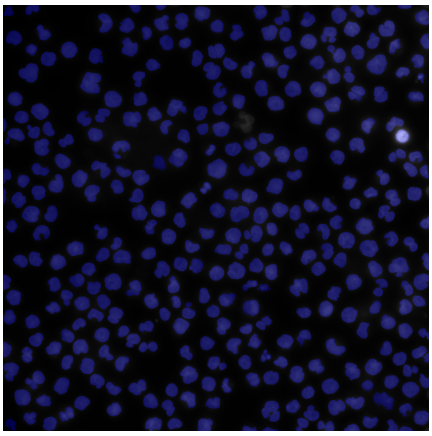


Figure 13: Local vs. Global thresholding (normal conditions)

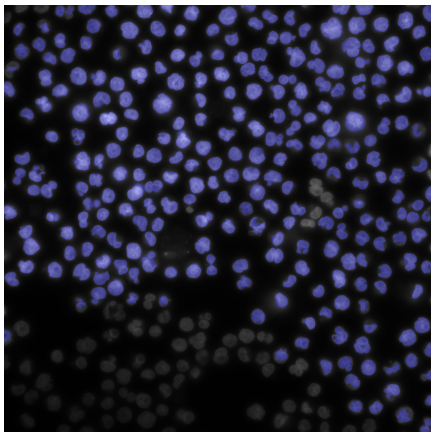
According to skimage documentation, minimum thresholding works in the following way: (cite https://doi.org/10.1111/j.1749-6632.1965.tb11715.x) it assumes that the his-



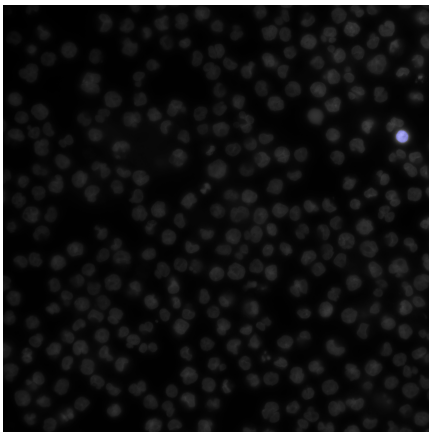
(a) Local



(b) Local



(c) Global, minimum



(d) Global, minimum

Figure 12: Local vs. Global thresholding

togram of the image is bimodal, meaning that it has two clearly defined peaks, then it iteratively smoothes the histogram using a running average of size $k = 3$ (see Equation 13) until only 2 local maximas are left. Afterwards, the lowest point between these two peaks is found and assigned to be a threshold value.

$$a_k = \frac{1}{k} \sum_{i=n-k+1}^{n} p_i \tag{13}$$

Then the threshold is taken as the minimum between the two local maximas.

$$x_k \leq T \leq y_k \tag{14}$$

TODO arrange this equation better

That is why also images which histograms have very unequal peaks or a broad and flat valley will be unsuitable for this method. (cite skimage)

7 Summary

List of Figures

1 Way in which photos of the well-plate were taken 3
2 Unet 5
3 Overfitting 6
4 Dropout 7
5 Not regularized 8
6 Regularized 8
7 No overlap 9
8 30 pixels overlap 9
9 Different lightning conditions 11
10 Closely located cells 12
11 Histogram as a probability density function 14
13 Local vs. Global thresholding (normal conditions) 15
12 Local vs. Global thresholding 16

List of Tables

1 Threshold timing 15