MULTIVIEW VISUAL SEGMENTATION FOR MECHANOBIOLOGY

Group 76
Luísa Maria Coimbra Cortes - s232655
Lőrinc Pályi - s241774
Eglantine Madeleine Olympe Anton - s233242
Rong Jet Cheong - s241961

DTU Compute

ABSTRACT

This project explores the application of deep learning to the segmentation of biological cells in brightfield microscopy images without the use of fluorescent markers. Given the challenge of distinguishing cells from background in images with varying focal depths and significant background presence, the focus is on evaluating convolutional neural networks (CNNs) for segmentation tasks in mechanobiology. The dataset consists of 12,793 images paired with binary masks that capture the complexity of cellular morphology. Key aspects such as model architecture, loss functions and optimisation techniques are analysed, with a particular focus on dealing with class imbalance and improving computational efficiency. The results highlight the effectiveness of deep learning in achieving high segmentation accuracy, offering a promising approach for mechanobiology research where traditional fluorescent labelling may not be feasible.

***Index Terms*—** Deep-learning, Cell segmentation, Image Processing, Mechanobiology

1. INTRODUCTION

The quantification of the mechanical forces exerted by biological cells on their substrate is fundamental to the comprehension of cell behaviour. Traditionally, fluorescent markers are employed for the identification of cells and the nanopillars they deform [1] [2]. However, these markers can introduce complexity and are often unsuitable for a multitude of applications. The objective of this project is to investigate the potential of deep learning models, including convolutional neural networks, for accurately segmenting cell locations in brightfield images, thereby facilitating a more widespread application. This project focuses on implementing various deep learning-based image segmentation models and systematically comparing their performance. The objective is to evaluate the effectiveness of different models in segmenting images, aiming to achieve high precision while keeping computational costs low. Through detailed benchmarking, we aim to identify models that balance simplicity and performance, potentially matching or surpassing the capabilities of more complex, generalized models.

2. METHODS

2.1. Data description

The dataset consists of 12,793 brightfield images captured at different focal points and 1,163 corresponding binary masks. All images have the same dimensions (1024x1024 pixels). The images are organized into 7 wells, with each well containing between 50 and 225 locations. Each location includes 11 brightfield images taken at different focal depths. The binary masks, generated using fluorescence data, correspond to the cells' location in the brightfield image sets and are focused on a specific z-plane representing the optimal focus point. Figure 1 represents one sample at different focal points with its corresponding overlaid mask.

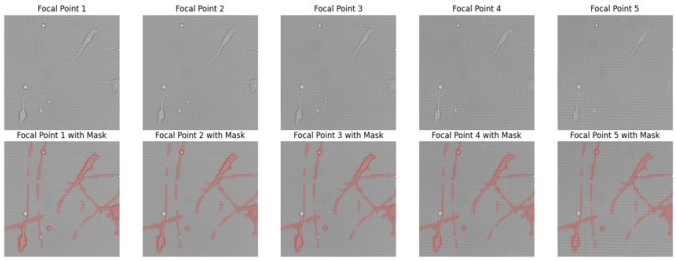
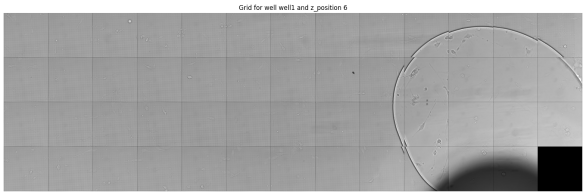


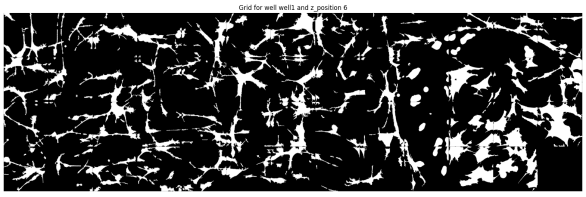
Fig. 1: Top row: In gray-scale, sample 2, on well 1, is represented for 5 focal points (z). Bottom row: the same brightfield pictures are represented with the masks on top in red color.

The dataset is divided into training (well 1), and test subsets (wells 2-7). The data reflect real-world experimental conditions and thus include variability and noise. The grid of the brightfield images and their corresponding ground truth masks for well 1 is presented in Figure 2. From this visualization, it is evident that a single focal point does not capture

all parts of the cells, even to the human eye. In addition, some air bubbles are visible and should not be considered part of the cells. This represents an additional challenge for segmentation.



(a) Brightfield grid for well 1 with the 6th focal point



(b) Mask of the brightfield grid for well 1 with the 6th focal point

Fig. 2: Visualization of well 1

Additionally, it is important to note that the class distribution is unbalanced. This is visible on Figure 2, but is also highlighted in Figure 3, where the cells account for only 10-20% of the full picture. This imbalance poses additional challenges in segmentation.

2.2. Traditional Image Transformation Techniques

A simple image processing approach using image transformations from the cv2 library to generate masks was intially considered. This involved applying diverse transformations, such as morphological opening, blurring or thresholding to identify the regions of interest. A few samples were considered and visually compared to their corresponding ground truth, but no robust evaluations were performed. The idea was to get an overview of the limitations of traditional image transformation techniques, in order to highlight the relevance of neural networks.

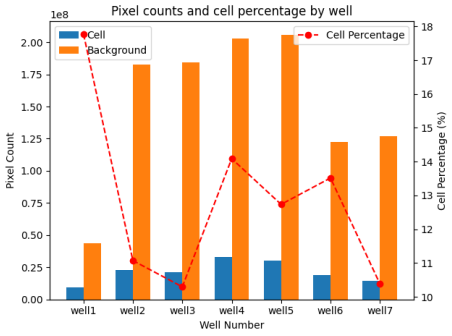


Fig. 3: Class distribution showing the imbalance, where cells account for only 10-20% of the full picture.

2.3. Neural network

2.3.1. Data loading

To make data loading easier, a CSV file was created. Each row represents an image, and the columns contain all useful information about the images, such as well number, location number, file format, path, sample site, z position, corresponding mask path etc... PyTorch was used as the primary framework. The images and masks were resized to 256 × 256 to limit computational cost, and converted to tensor format. A batch size of 16 was used. The training data were shuffled at each epoch, whereas the test data didn't use shuffling to preserve the data order for consistent evaluation.

2.3.2. Architecture

A U-Net architecture was employed. It is a convolutional neural network (CNN) specifically designed for image segmentation. The structure consists of an encoder, that progressively reduces dimensions while extracting feature representations, and a decoder that reconstructs the spatial resolution while concatenating features from corresponding levels in the encoder.

The U-Net model used in this study was designed to segment images with different numbers of input channels (corresponding to the number of focal points included) and to produce a single-channel binary mask. The encoder has three contraction blocks and the decoder has three expansion blocks. Each contraction block consists of two convolutional layers followed by ReLU activation. The first contraction block uses a convolutional kernel size of 7 with a padding of 3, while the subsequent blocks employ a kernel size of 3 with a padding of 1. The encoder increases the number of feature channels progressively to 64, 128, and 256. For the decoder, each expansion block comprises two transposed convolutional layers, followed by ReLU activation. Skip connections are integrated at each stage of the decoder by adding the output from the corresponding encoder layer to the upsampled feature map. The final layer of the decoder is a single transposed convolution that produces the one-channel segmentation map.

2.3.3. Loss function

The Loss function quantifies the difference between the predicted output and the ground truth, guiding the optimization process. The following loss functions were tested to optimize the model's performance:

- **Dice Loss:** Emphasizes the overlap between predicted and ground truth regions

- **Binary Cross-Entropy Loss**: Evaluates pixel-wise classification performance by penalizing incorrect predictions.
- **Combined Loss**: A sum of Dice and Cross-Entropy Loss. It was introduced to combine the strengths of the two approaches.

2.3.4. Optimizers

The optimizer is used to update the network weights during training to minimize the loss function. The following optimizers were considered:

- **ADAM Optimizer**: An adaptive learning rate method.
- **Stochastic Gradient Descent (SGD)**: A method updated iteratively by computing gradients.
- **SGD with Momentum**: An extension of SGD with incorporation of momentum to accelerate convergence and avoid local minima.

Different values of learning rate for each optimizer were experimented to reach faster and more stable convergence.

2.3.5. Feature selection

A feature selection analysis was performed to determine if any focal points in the data were redundant or lacked meaningful contributions. Through an visual analysis of the dataset it was observed that some focal lengths do not contribute much to the shape of the mask, and including them might introduce noise to the model. The aim was to reduce said noise by only choosing the most relevant focal lengths to train on, which was achieved using a simple backward feature selection system.

2.3.6. Overfitting mitigation

Given the limited size of the dataset, overfitting was a major concern. Several strategies were implemented to try to prevent it:

- **Normalization Techniques**: Batch Normalization and Group Normalization were explored.
- **Dropout**: Different probabilities were tested.
- **Data Augmentation**: Random rotation, horizontal and vertical flipping and scaling were applied to artificially increase the diversity of the training data.
- **L2 regularization**: weight decay was used in the Adam optimizer.

2.3.7. Performance metrics

Different metrics were used to assess the performance of the segmentation results. They are computed using True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). The model's performance was evaluated using the following metrics:

- **Intersection over Union (IoU)**: Measures the overlap between predicted and ground truth regions relative to their union

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}$$

- **Dice Score**: Reflects the overlap between predictions and ground truth

$$\text{Dice Score} = \frac{2 \cdot \text{TP}}{2 \cdot \text{TP} + \text{FP} + \text{FN}}$$

- **Rand Index**: Reflects the proportion of true prediction

$$\text{Rand Index} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

The IoU and Dice score were chosen given that the data was imbalanced (more background pixels than foreground pixels). They penalize false positives and false negatives, providing a more meaningful evaluation of segmentation performance than simply the rand index.

2.3.8. General strategy

A baseline U-Net model was trained to establish a reference performance. From there, various hyperparameters were systematically modified to try to improve the performance of the model. The hyperparameters explored included: Learning Rate, Loss Function, Dropout, Normalization Techniques as Group Normalization and Batch Normalization, and Optimizer.

Each configuration was evaluated based on performance metrics on the test set for a quantitative assessment and the model predictions were visually inspected on a few samples for a qualitative analysis of the results.

3. RESULTS

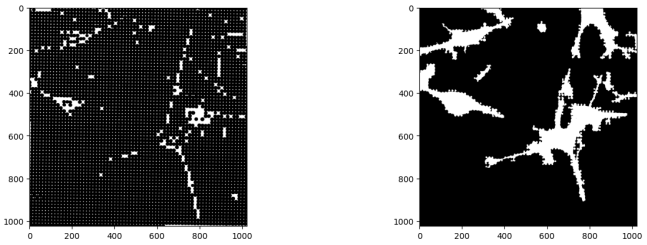
3.1. Traditional Image Transformation Techniques

An example of the images generated by the manual approach can be observed in Figure 4. While the overall cell locations were successfully detected, significant portions of their areas are missing, and the nanopillars are still very visible as background noise in the image.

3.2. Neural network

3.2.1. Baseline

The results of the baseline model are presented in the first row of Table 1. It reached an accuracy IoU of 0.4287, a Dice Score of 0.5772 and an accurcy of 0.8519. The baseline model uses ADAM optimizer with a learning rate of 0.0001, the BCE loss funciton and batch normalization.



(a) Image Transformation Techniques

(b) Ground truth mask.

Fig. 4: Traditional Image Transformation Techniques: : the white zones identify the limits of the cell

3.2.2. Feature selection

The results of the feature selection analysis are presented in Table 2 and Table 3. In the first iteration, excluding focal point 8 yielded the highest score across all performance metrics. In the second iteration, excluding focal point 4 resulted in the highest score for all metrics. However, this process did not provide any reproducible improvements and was very time consuming to run, leading it to be abandoned. All focal points were retained in the up-coming model comparison section.

3.2.3. Models comparison

The U-Net model 1 consistently outperformed other architectures, in terms of IoU (0.5055) and Dice (0.6669), but not for accuracy, where model 2 achives 5% more. Both metric and validation loss plots are presented in Figure 6. After a slight increase during the first epochs, the evaluation metrics converge and stabilize around epoch 15, indicating successful model training and convergence.

The results for one of the samples to which this model was applied are shown in Figure 5.

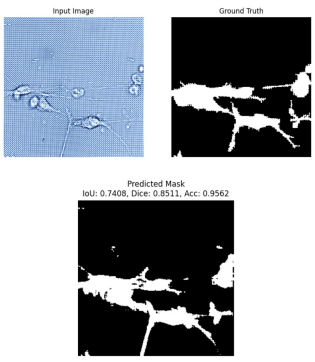


Fig. 5: Comparison of Bright Field Picture, Ground Truth Mask, and Predicted Mask of Model 1 for an example sample

4. DISCUSSION

4.1. Traditional Image Transformation Techniques limitations

The exploration of traditional image transformation techniques, even though their performance were not robustly quantified, highlighted some limitations. They rely on defined transformation and preset parameters (fixed threshold, fixed kernel size etc...), which struggle to generalize to complex and diverse datasets. This justify the interest of more advanced deep learning models.

4.2. Feature selection

The feature selection process did not provide any reproducible improvement. While two focal points were removed from the original dataset without decreasing the overall performance of the model, the process could not be explored further due to time restriction. Furthermore, to ensure consistent results, the best practice will have to be running each test multiple times and averaging the results.

Training the model on the whole original dataset took approximately 12 hours. Reducing the number of channel inputs would have been an interesting way of reducing this time.

4.3. Model comparison

Dropout appeared to play a critical role in preventing overfitting and enhancing performance. Model 1 and 2 (with respectively 0.1 and 0.3 dropout) both achieved better IoU and Dice score compared to the baseline model with 0 dropout. However, excessive dropout, as seen in Model 3 and 4, limits the model's learning.

Reducing the learning rate consistently led to improved performance. This can be attributed to the fact that lower learning rates enable the models to make finer, more precise adjustments to their weights, preventing them from overshooting optimal solutions and improving convergence.

Combining BCE and Dice loss also seems to improve the performance. This is because it combines the advantages of pixel-level classification with spatial overlap, addressing issues of class imbalance and boundary accuracy more effectively than using either loss function alone.

5. CONCLUSION

The U-net architecture achieved good performance for segmenting cell locations in brightfield images. The training time was reasonable, around 12 hours trained in GPU. This validates the potential of CNNs in mechanobiology, offering a non-invasive alternative to fluorescent markers. Future improvements could focus on implementing early stopping techniques, reducing the number of focal points to reduce training time expanding the dataset to enhance generalization.

6. REFERENCES

[1] John L Tan, Joe Tien, Dana M Pirone, Darren S Gray, Kiran Bhadriraju, and Christopher S Chen, "Cells lying on a bed of microneedles: An approach to isolate mechanical force," Tech. Rep.

[2] Olivia Du Roure, Alexandre Saez, Axel Buguin, Robert H Austin, Philippe Chavrier, Pascal Silberzan, and Benoit Ladoux, "Force mapping in epithelial cell migration," Tech. Rep., 2005.

Attachments

Model	IoU	Dice	Accuracy (Rand Index)	Learning Rate	Loss Function	Dropout	GroupNorm	BatchNorm	Optimizer	Channels
Baseline	0.4287	0.5772	0.8519	1.00E-04	BCE	0	None	yes	Adam	All
1	0.5055	0.6669	0.8225	1.00E-03	BCE + Dice	0.1	8	no	Adam	All
2	0.4893	0.6399	0.867	1.00E-04	BCE	0.3	None	yes	Adam	All
3	0.1663	0.277	0.6791	1.00E-04	BCE	0.5	None	yes	Adam	All
4	0.469	0.6215	0.8568	1.00E-04	BCE	0.6	None	yes	Adam	All
5	0.4224	0.5788	0.8287	1.00E-01	BCE	0	None	yes	SGD	All
6	0.423	0.5746	0.8325	1.00E-02	BCE	0	None	yes	SGD	All
7	0.4631	0.6158	0.8463	1.00E-03	BCE	0	None	yes	Adam	All
8	0.4456	0.5972	0.829	1.00E-02	BCE	0	None	yes	SGD w\ momentum	All
9	0.4141	0.5591	0.8498	1.00E-03	BCE	0	None	yes	SGD w\ momentum	All
10	0.4963	0.6433	0.8646	1.00E-04	Dice	0	None	yes	Adam	All

Table 1: Experimental Results

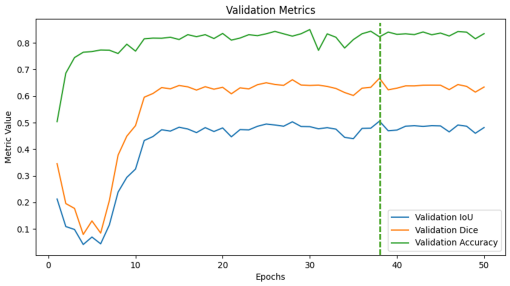
Exclusion	IoU	Dice	Rand
Baseline	0.4287	0.5772	0.8519
1	0.4651	0.6138	0.8492
2	0.4814	0.6357	0.8527
3	0.4998	0.6474	0.8634
4	0.4617	0.6119	0.8559
5	0.4844	0.6361	0.8573
6	0.4990	0.6487	0.8599
7	0.4769	0.6204	0.8592
8	0.5022	0.6498	0.8673
9	0.4881	0.6345	0.8597
10	0.4591	0.6117	0.8360
11	0.4779	0.6425	0.8415

Table 2: Performance metrics for the first iteration of channel selection. Each row corresponds to the exclusion of one focal point compared to the baseline of Table 1

Exclusion	IoU	Dice	Rand
Baseline (8)	0.5022	0.6498	0.8673
8 + 1	0.4907	0.6381	0.8614
8 + 2	0.4706	0.6125	0.8613
8 + 3	0.4713	0.6319	0.8466
8 + 4	0.5037	0.6517	0.8684
8 + 5	0.4830	0.6333	0.8627
8 + 6	0.4761	0.6179	0.8564
8 + 7	0.4534	0.6027	0.8536
8 + 9	0.4763	0.6306	0.8441
8 + 10	0.4914	0.6363	0.8636
8 + 11	0.4870	0.6410	0.8572

Table 3: Performance metrics for the second iteration of channel selection. The baseline corresponds to the exclusion of channel 8, as it provided the best results in Table 2. Each row corresponds to the exclusion of an additional focal point compared to the baseline

(a) Validation loss plot for Model 1. The green-dotted line indicates the best results (epoch 38).



(b) Metrics plot showing IoU, Dice, and accuracy for Model 1 up to epoch 50.

Fig. 6: Performance of U-Net Model 1: Loss and metric plots for validation performance, highlighting the best epoch at 38.