École Polytechnique

Segmentation and Statistical Analysis of Cellular Images using Deep-Learning

by Yiming Wei

Lab Rotation Report

Prof. Abdul Barakat
Report Advisor

Manuel Carrasco Yagüe
Report Supervisor

Bd des Maréchaux, 91120 Palaiseau

June 20, 2023

To Nature, for Its variety of wonders, beautiful inspirations, and essential perseverance;
to Chaos, for creating the opportunities;
and to Time, for allowing everything to exist.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Abdul Barakat. Your leadership, guidance, and continued support throughout this internship have been invaluable. Your professional acumen and ability to create a conducive environment for learning are truly inspirational. The knowledge I have gained under your supervision will undoubtedly serve as the bedrock of my future endeavors.

I would also like to express my profound thanks to Manuel Carrasco Yagüe, whose tireless guidance has helped shape my coding skills. His hands-on approach in showing me how to construct an efficient working environment, use GitLab, and write clean, efficient code has significantly impacted my understanding and proficiency in the field of computer science. Manuel, you went above and beyond the duties of a mentor, showing me the nuts and bolts of our profession, and for that, I am sincerely grateful.

The experience and skills acquired during this two-month internship are a testament to the quality of your supervision and mentoring. Your dedication to my professional development has had a profound effect on me, and for that, I am eternally grateful. You both have left an indelible mark on my academic and professional journey, and for that, I am forever grateful.

I am optimistic that the experience and knowledge acquired during my internship under your guidance will be instrumental in my future career. I look forward to building on this foundation and hopefully making you proud of your investment in me.

Thank you, Professor Abdul Barakat and Manuel Carrasco Yagüe, for everything you've done during my internship period.

Palaiseau, June 20, 2023 Yiming Wei

Abstract

This project delves into the transformative role of deep learning in cellular image analysis, with a particular emphasis on its application in the field of bio-impedance spectroscopy, also known as electro-chemical impedance spectroscopy(ECIS). This technology, which allows for continuous, real-time monitoring of cells, serves as the backdrop for the study.

The cells used in this study are breast epithelial cells MCF10A and their genetically mutated counterparts MCF10A_Braf and MCF10A_Rac. The selection of these cells enables researchers to gain critical insights into breast cancer progression. The research investigates the segmentation and statistical analysis of cellular images, a critical aspect of biomedical engineering. The focus is on understanding the impact of preprocessing and the quantity of training images on the accuracy of the deep learning model. This exploration is crucial in the context of the often limited availability of annotated biomedical images, and the need for models that can learn representative features from relatively small datasets.

In addition to the deep learning aspect, the project also explores the temporal dynamics of cell area and cell numbers. The changes in these parameters are fitted with appropriate mathematical functions, providing a quantitative understanding of cellular behavior over time.

The outcome of this research promises to offer significant insights into the behavior of breast epithelial cells and their mutated counterparts. As a result, it may facilitate the development of new strategies for breast cancer detection.

Résumé

Ce projet étudie le rôle transformateur de l'apprentissage profond dans l'analyse d'images cellulaires, en mettant particulièrement l'accent sur son application dans le domaine de la spectroscopie de bio-impédance, également connue sous le nom de spectroscopie d'impédance électrochimique(ECIS). Cette technologie, qui permet une surveillance continue et en temps réel des cellules, sert de toile de fond à l'étude.

Les cellules utilisées dans cette étude sont les cellules épithéliales mammaires MCF10A et leurs homologues génétiquement mutées MCF10A_Braf et MCF10A_Rac. La sélection de ces cellules permet aux chercheurs d'acquérir des connaissances essentielles sur la progression du cancer du sein. La recherche porte sur la segmentation et l'analyse statistique des images cellulaires, un aspect essentiel de l'ingénierie biomédicale. L'accent est mis sur la compréhension de l'impact du prétraitement et de la quantité d'images d'entraînement sur la précision du modèle d'apprentissage profond. Cette exploration est cruciale dans le contexte de la disponibilité souvent limitée d'images biomédicales annotées et du besoin de modèles capables d'apprendre des caractéristiques représentatives à partir d'ensembles de données relativement petits.

Outre l'aspect apprentissage profond, le projet explore également la dynamique temporelle de la surface et du nombre de cellules. Les changements de ces paramètres sont ajustés à l'aide de fonctions mathématiques appropriées, ce qui permet d'obtenir une compréhension quantitative du comportement cellulaire au fil du temps.

Le résultat de cette recherche promet d'offrir des perspectives significatives sur le comportement des cellules épithéliales du sein et de leurs homologues mutés. Ils pourraient ainsi faciliter le développement de nouvelles stratégies de détection du cancer du sein.

Contents

Chapter 1

Introduction

1.1 Laboratory Rotations

This report is the culmination of my two-month internship journey in the field of Biomedical Engineering (BME), a period of immersive learning and practical application. As part of the program's unique design, students are required to complete rotations in at least two different BME laboratories, spending four weeks in each. This immersive process allowed me to shadow doctoral students or postdoctoral fellows and contribute to research projects of limited scope.

1.2 The Project

The project I am associated with is using impedance sensors to map the spatio-temporal dynamics of different cell types.

Impedance-based sensing of cells, also known as electro-chemical impedance spectroscopy (ECIS) or impedance cytometry, is a technology that has been used since at least the late 20th century. It's a noninvasive method that allows the continuous, real-time monitoring of cells.

One of the earliest instances of this technology was developed in the 1980s by Giaever and Keese[1], who discovered that small electric currents could be used to monitor the behavior of cells in culture. They observed that the impedance (resistance to electrical current) across a cell layer changed as cells moved and changed shape. This discovery led to the development of the ECIS technology, which is now widely used in biological and medical research for a variety of applications, such as studying cell adhesion, proliferation, migration, and other behaviors.

In this method, cells are cultured on an electrode array, and a small alternating current is passed through the array. The impedance of the cell layer is measured at different frequencies(that is why it is called spectroscopy), providing information about the cells' behavior(see figure:1.1). Since different types of cells and different cell states (e.g., living vs. dead, healthy vs. diseased) can have different impedance

characteristics, this technology can also be used for cell identification and characterization.

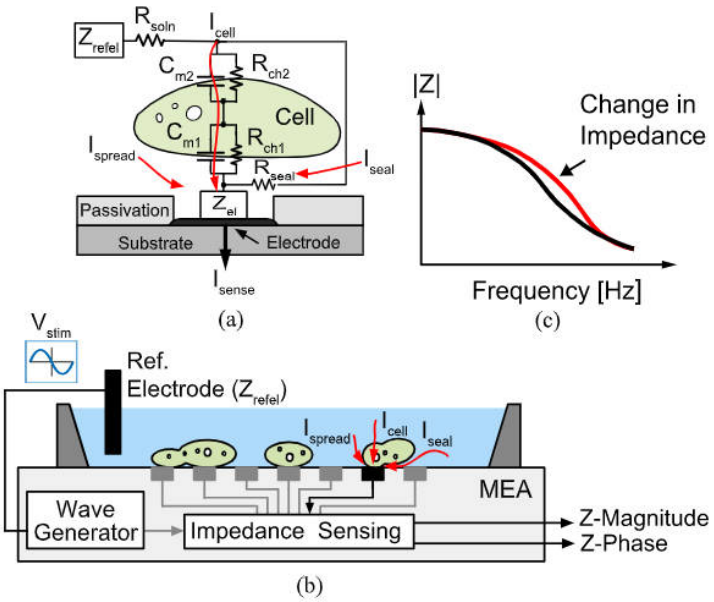

Figure 1.1: (a) Cell impedance model, (b) principle of EIS measurements and setup, (c) EIS response graph[2].

Recently, Pierluca Messina did a study that uses a machine learning system and an EIS sensor to identify different types of blood clots[3]. The system is trained to recognize four types of samples: Blood, and three types of clots named White, Mixed, and Red. The researchers tested the system using samples from different donors. The results showed that the system can accurately identify different types of blood clots, which could be useful in treating strokes.

Therefore, Manuel wants to use impedance sensors to map the spatio-temporal dynamics of different cell types(like figure:1.2), which in turn can be classified using machine learning. Also my goal is to use cellpose to segment the cell images and then count the number and area of cells, which can be combined with the obtained impedance information to better understand the cell changes.

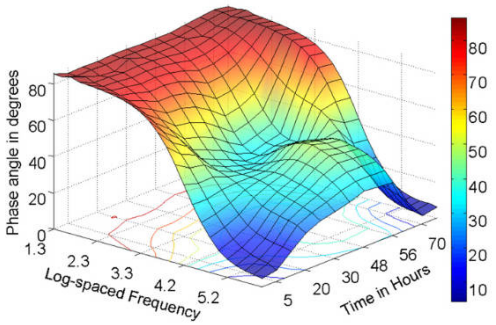

Figure 1.2: Changes in the phase angle of electrodes with the progression of culture time[4].

1.3 Cellpose

I mainly use Cellpose to segment the cells. Cellpose is a powerful tool designed for biological segmentation in cellular imaging[5]. It is an algorithm that can be applied broadly across different types of cellular images due to its generalist nature(see figure:1.3). The need for such a tool arises from the vast diversity of biological images of cells, which can vary based on factors such as microscopy techniques, tissue types, cell lines, and fluorescence labeling. As new advances in biology and microscopy continue to expand the diversity of cells and signals that can be monitored, the challenge for automated segmentation approaches increases.

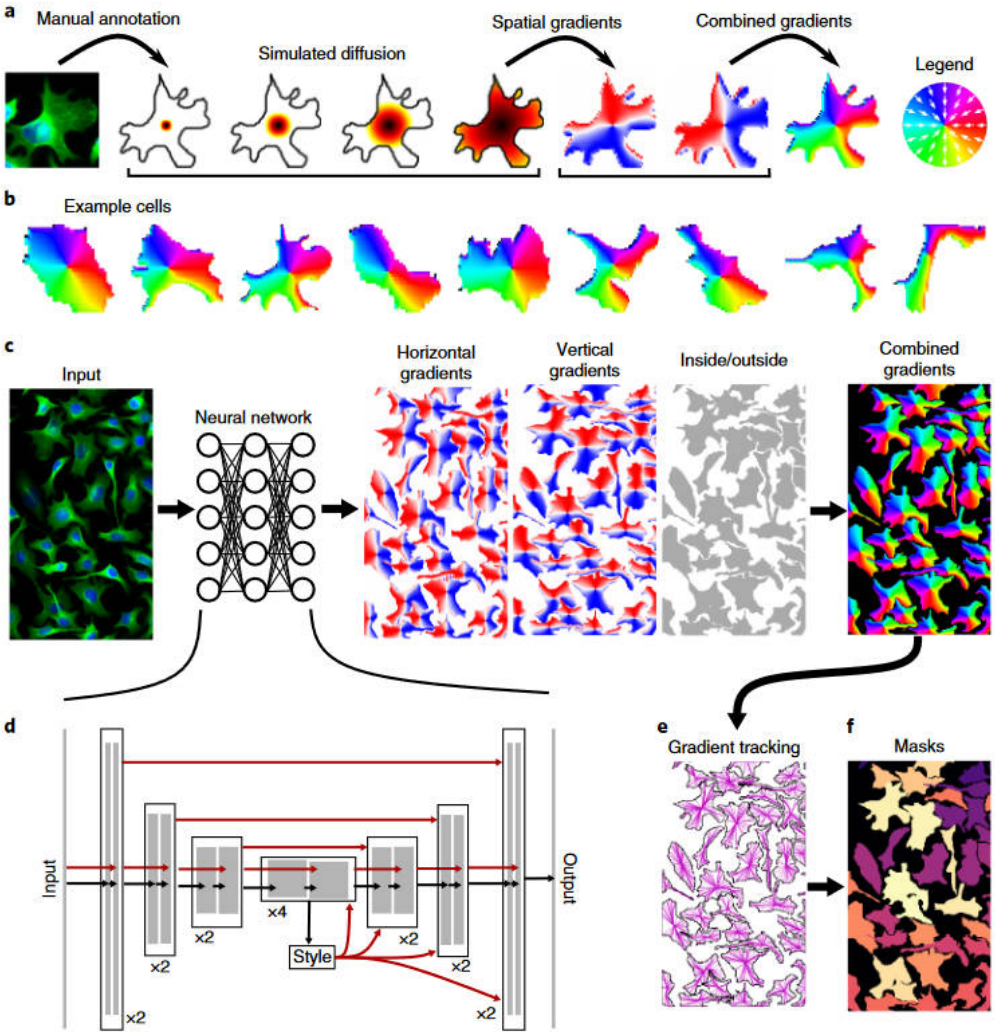

Figure 1.3: Model architecture[5].

Pretrained neural network models like Cellpose can provide good results for many image types right out of the box. However, these models often don't allow users to adapt the segmentation style to their specific needs and can perform suboptimally for test images that are very different from the training images.

To tackle this issue, Cellpose 2.0 was introduced[6]. It includes an ensemble of diverse pretrained models as well as a human-in-the-loop pipeline for rapid prototyping of new custom models. These models,

pretrained on the Cellpose dataset, can be fine-tuned with only 500-1,000 user-annotated regions of interest (ROI) to perform nearly as well as models trained on entire datasets with up to 200,000 ROI. This human-in-the-loop approach further reduces the required user annotation to 100-200 ROI, while still maintaining high-quality segmentations. To facilitate the adoption of Cellpose 2.0, the package provides software tools such as an annotation graphical user interface, a model zoo, and a human-in-the-loop pipeline(see figure:1.4).

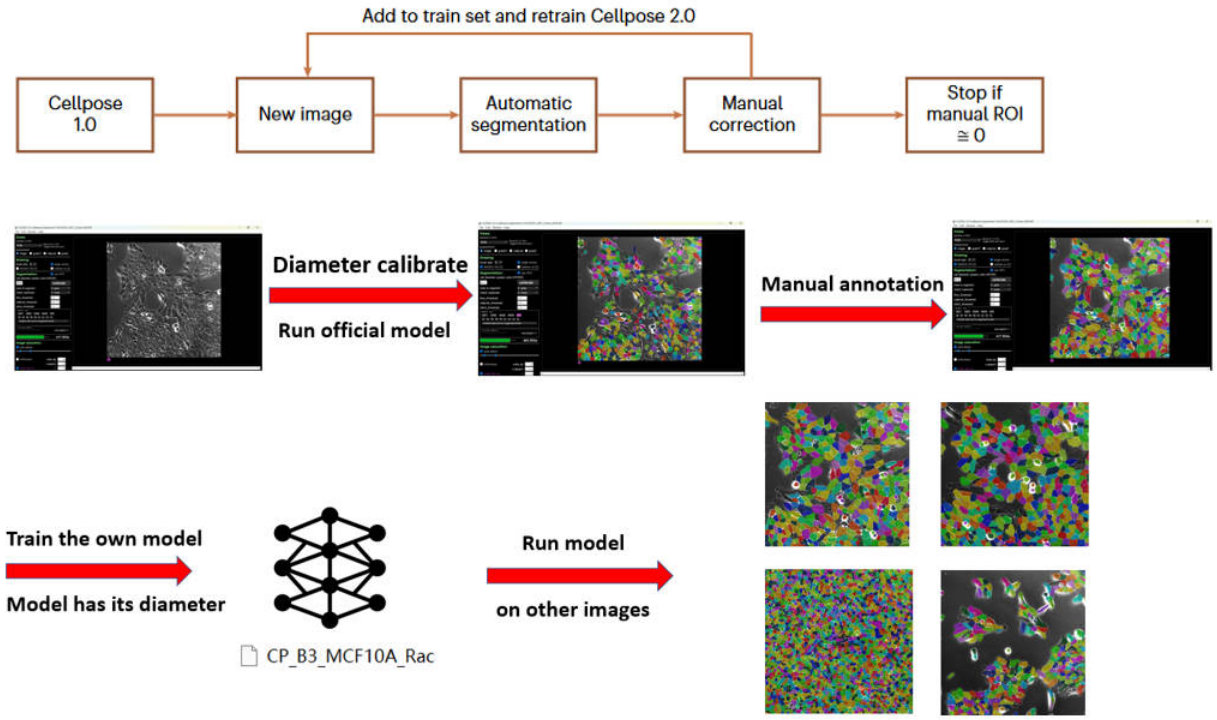

Figure 1.4: A human-in-the-loop approach for training specialized Cellpose models[6].

Chapter 2

Dataset and Preprocessing

2.1 Dataset

The MCF10A cell line is a non-tumorigenic human mammary epithelial cell line extensively used as a model system in breast cancer research[7]. These cells are genetically stable and exhibit features that mimic the normal cellular processes, including cell cycle regulation, apoptosis, and differentiation. The MCF10A_Braf and MCF10A_Rac cell lines are genetically modified versions of the parent MCF10A cells. In these cell lines, the Braf and Rac proteins, respectively, have been overexpressed.

The Braf protein is a part of the RAF kinase family, which plays a role in regulating cell division and differentiation. Mutations in this gene are associated with various types of cancer, including melanoma, colorectal cancer, non-small cell lung cancer, and others. Overexpressing this protein in a cell line like MCF10A could be useful for studying the effects of such mutations.

Rac is a member of the Rho family of small GTPases, which are involved in a wide variety of cellular processes, including cell growth, cytoskeletal reorganization, cell movement, and cell division. Overexpressing this protein could be useful for studying its role in these processes and how disruptions might contribute to diseases like cancer.

The dataset consists of bright-field microscopy images taken from the cultures of three cell lines: MCF10A, MCF10A_Braf, and MCF10A_Rac(see figure:2.1). These cell lines were maintained under the same culture conditions for a period of approximately 70 hours. Throughout this duration, an image of each culture was captured hourly, generating a comprehensive and time-resolved visual record of cellular activity and changes.

The imaging was conducted across five distinct areas within each cell culture dish(see figure:2.1), representing five replicate samples for each cell line at each time point. This was done to ensure a broader perspective of the culture dynamics, capturing any potential heterogeneity in cell behavior across different spatial regions of the culture medium.

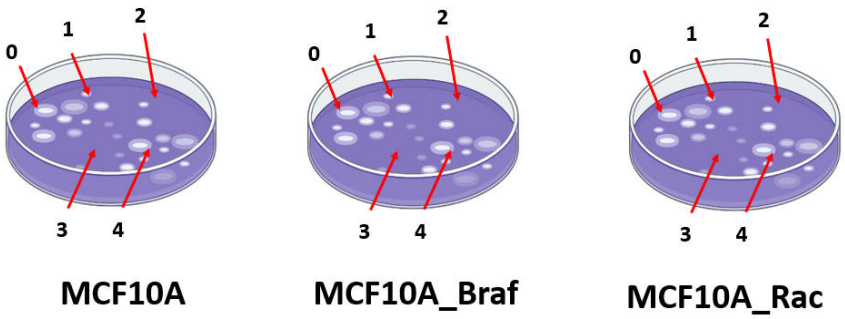

Figure 2.1: Schematic diagram of cell culture medium.

Here cell growth in a culture medium typically goes through three main stages[8]:

1) Lag Phase: Cells adjust to the new environment and prepare to grow but do not divide yet.

2) Log Phase: Cells start to grow rapidly and divide, increasing in number quickly.

3) Stationary Phase: Growth slows down because resources are getting used up and waste is building up. The number of cells stays steady because the rate of cell division equals the rate of cell death.

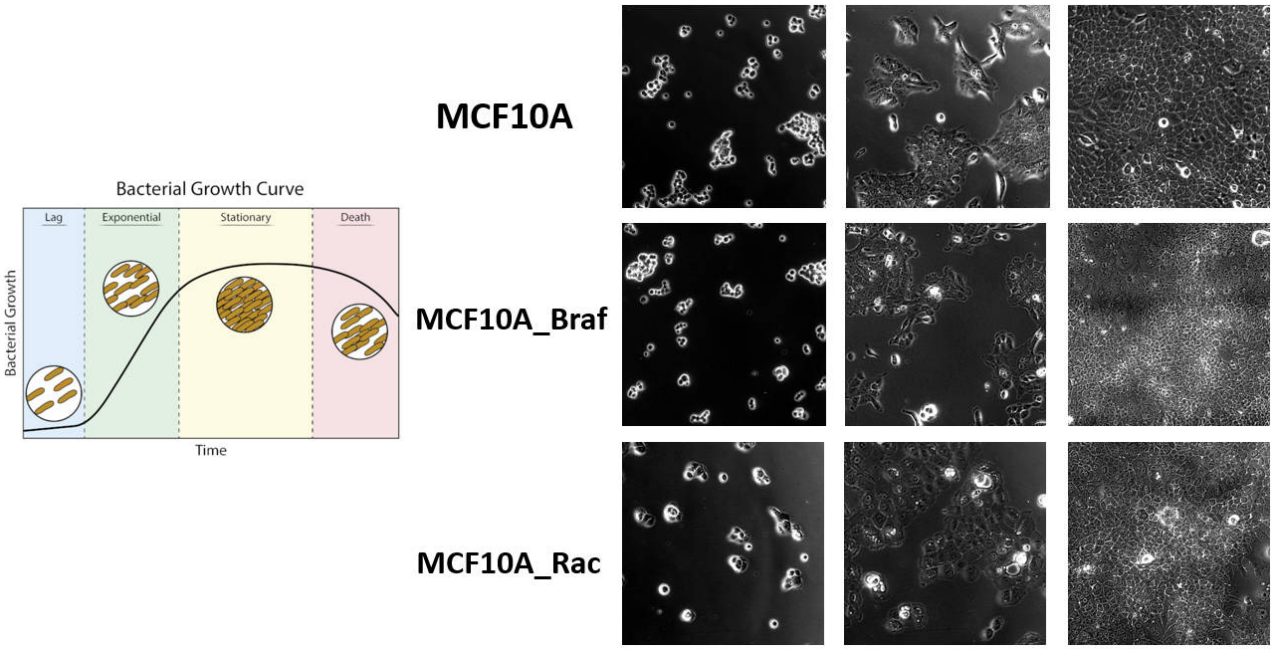

Figure 2.2: Diagram of cell growth.

Each image in the dataset represents a unique combination of cell line, time point, and spatial location within the dish. As a result, this dataset provides a rich source of visual data for the exploration of cell growth, division, and morphological changes under the influence of Braf and Rac overexpression. This in-depth data collection strategy allows for a detailed analysis of cellular dynamics over time and space, offering potential insights into the influence of these proteins on cellular behavior.

2.2 Preprocessing

2.2.1 Method

The preprocessing of the images involved the application of Contrast Limited Adaptive Histogram Equalization (CLAHE) and a Sigmoid function[9].

CLAHE

CLAHE is an algorithm that is used to enhance contrast in images. Unlike standard histogram equalization, which applies a global contrast transformation based on the entire image histogram, CLAHE operates on small, localized regions of the image(figure:2.1). This ensures that contrast enhancement is adaptive and takes local image characteristics into account.

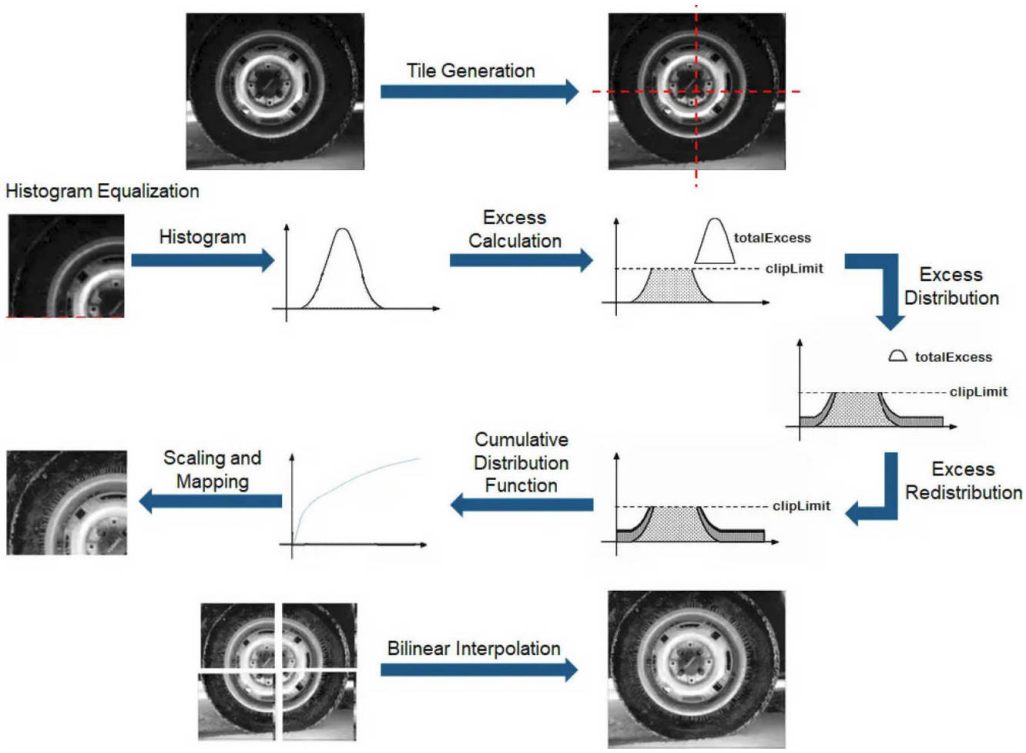

Figure 2.3: CLAHE algorithm and noise amplification control - source: Mathwork.

The algorithm involves redistributing pixel intensities across each tile, enhancing the contrast within that region. However, in order to prevent the over-amplification of noise, the redistribution of intensities is limited (contrast-limited). The amount of contrast enhancement in each tile can be controlled with a contrast limiting parameter.

This technique has the benefit of enhancing the visibility of features in regions that are darker or lighter than the average, making it a valuable tool for the analysis of biological images.

Sigmoid Function

After applying CLAHE, we applied a Sigmoid function to the images. The Sigmoid function, often used in machine learning and deep learning, is a mathematical function that outputs a value between 0 and 1.

When applied to image processing, the Sigmoid function can help in adjusting the contrast and brightness of the image. Each pixel intensity in the image is mapped using the Sigmoid function, effectively spreading out or compressing pixel intensity values depending on the sigmoid curve parameters.

The use of a Sigmoid function can thus enhance contrast and improve visibility of details in the images, especially in cases where the contrast variation is subtle.

2.2.2 Interactive Image Display with ipywidgets

To visually evaluate the effects of CLAHE and Sigmoid function on the images and their histograms, we implemented interactive adjustment buttons using ipywidgets, a Python library that provides interactive widgets for the Jupyter notebook.

In our case, we used ipywidgets to create sliders for adjusting the parameters of the CLAHE and Sigmoid functions. This allowed us to dynamically adjust these parameters and immediately visualize the resulting changes in the processed images and their corresponding histograms(figure:2.4).You can see the increased contrast of the pre-processed image. This interactive tool provided a convenient means for real-time assessment and optimization of the image preprocessing steps.

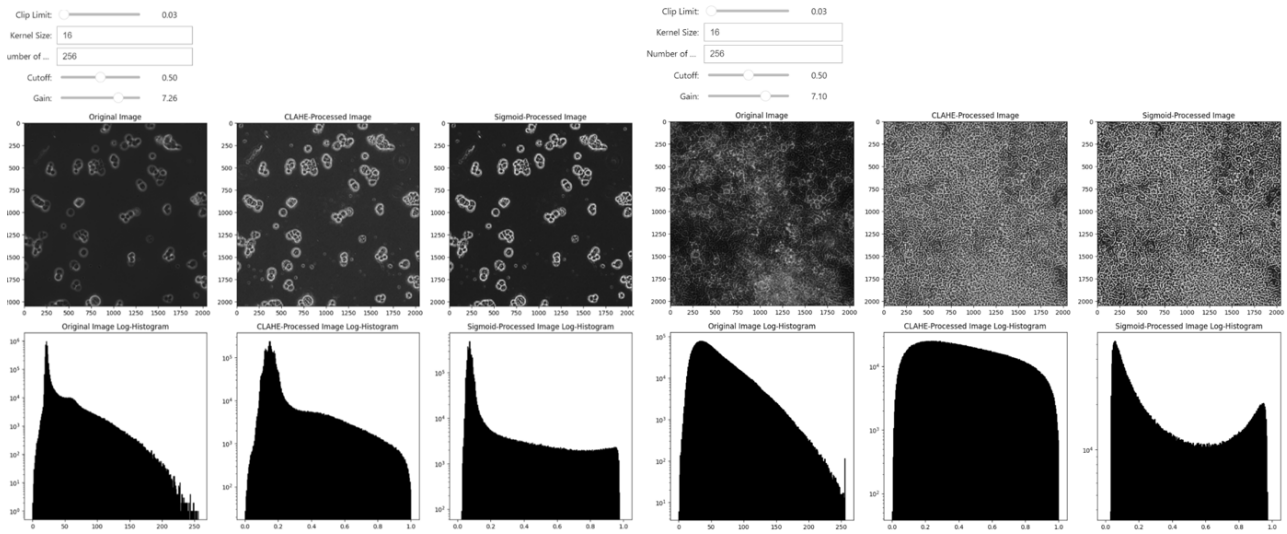

Figure 2.4: Interactive Image Display with ipywidgets.

2.2.3 Image quality metrics

Peak Signal-to-Noise Ratio (PSNR)

PSNR is a commonly used metric in image processing for measuring the quality of reconstructed or processed images. It quantifies the difference between a reference image and a distorted image in terms of error per pixel.

It is defined as the ratio between the maximum possible power of a signal (image) and the power of corrupting noise (error) that affects the fidelity of its representation. First, the mean-squared-error(MSE) is calculated as follows:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

where: $I(i,j)$ and $K(i,j)$ are the pixel intensities of the reference and distorted images at location (i,j), m and n are the dimensions of the images.

Then, the PSNR is calculated from the MSE:

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right)$$

where MAX_I is the maximum possible pixel value of the image. For an 8-bit grayscale image, this would be 255. The PSNR is expressed in decibels (dB), and higher values indicate better quality, i.e., less difference from the reference image.

Higher PSNR values indicate better quality reconstructions or less difference from the reference image. However, it's important to note that PSNR might not always align with human subjective assessment of image quality as it does not consider visual perception and the spatial characteristics of the human eye.

Structural Similarity Index Measure (SSIM)

SSIM is another metric used for measuring the similarity between two images. Unlike PSNR which is a simple signal fidelity measure, SSIM is designed to model the perceived change in the structural information of the image[10].

The SSIM index is a method for comparing similarities between two images. The SSIM index is calculated on various windows of an image. The measure between two windows x and y of common size $N \times N$ is:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

where: μ_x is the average of x; μ_y is the average of y; σ_x^2 is the variance of x; σ_y^2 is the variance of y; σ_{xy} is the covariance of x and y; $C_1 = (k_1 L)^2, C_2 = (k_2 L)^2$ are two variables to stabilize the division with weak denominator; L is the dynamic range of the pixel-values (typically this is $2^{\#bits\ per\ pixel} - 1$); $k_1 = 0.01, k_2 = 0.03$ by default.

The resultant SSIM index is a decimal value between -1 and 1, where 1 means perfect structural similarity. It is often used in research for measuring the quality of reconstructed or processed images.

2.2.4 Hyperparameter tuning

Hyperparameters associated with the CLAHE and Sigmoid function were manually tuned to optimize the image preprocessing steps. The same image was processed under different parameter sets and the quality of the resulting images was evaluated using two metrics: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). However, we noticed that the changes in image quality as perceived visually

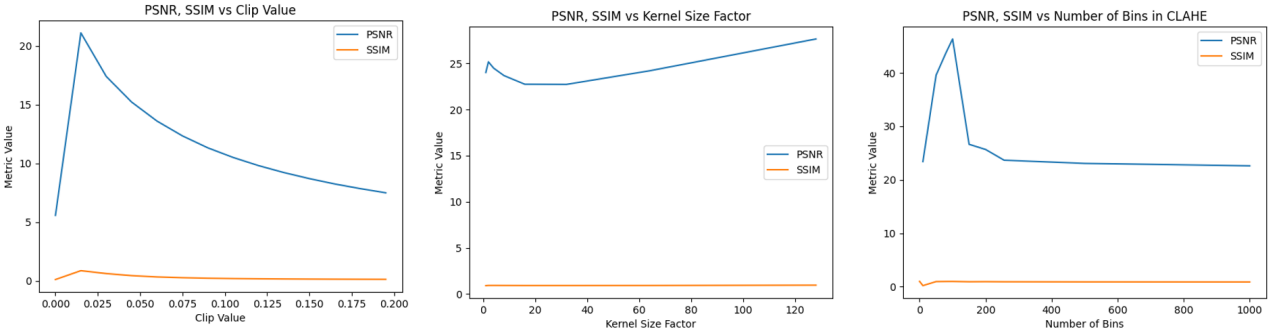

Figure 2.5: Hyperparameter tuning for CLAHE filter.

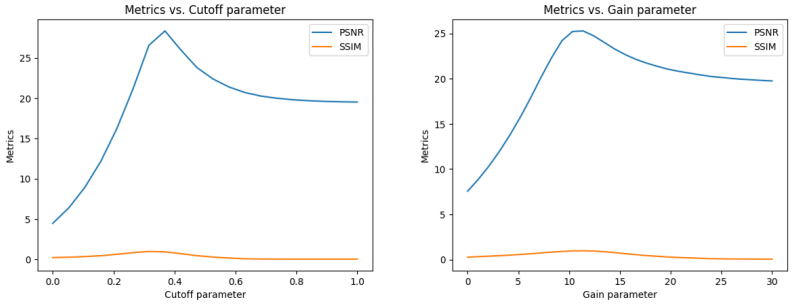

Figure 2.6: Hyperparameter tuning for Sigmoid filter.

did not correspond well with the changes in PSNR and SSIM values(figure:2.5 and 2.6). This discrepancy could be due to the fact that PSNR and SSIM, while popular for their ease of computation, may not perfectly align with human perception of image quality, particularly in complex biological images.

Given this observation, we decided to rely on a combined approach for final parameter selection. We took into account both the empirical guidelines provided by Francois' report[11] and our own visual assessments. So we used CLAHE with clip limit = 0.01, kernel size = 31, nbins = 256, Sigmoid with cutoff=0.05 and gain=1.

Chapter 3

Segementation and Statistical Analysis

3.1 Segementation

To train our own model, we randomly selected 5 images from the image set of each region of each cell to be labeled. Thus, for each cell we will have 25 images to train a model.

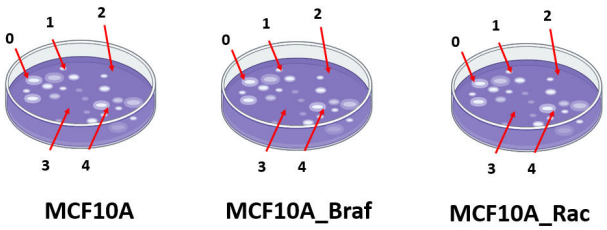

Figure 3.1: Schematic diagram of cell culture medium.

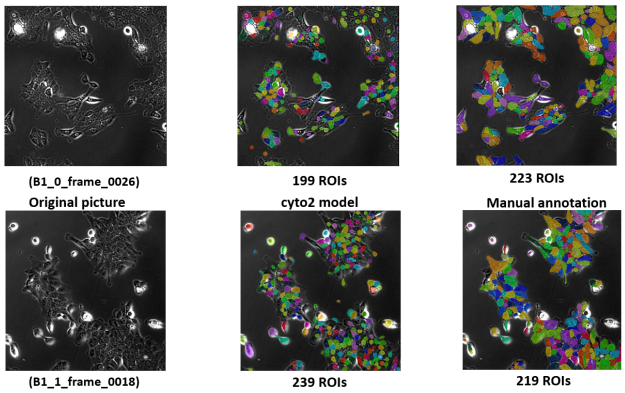

Figure 3.2: Labeling of Lag phase cells.

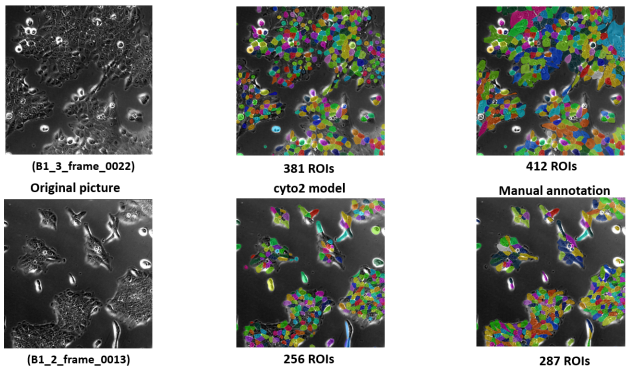

Figure 3.3: Labeling of Log phase cells.

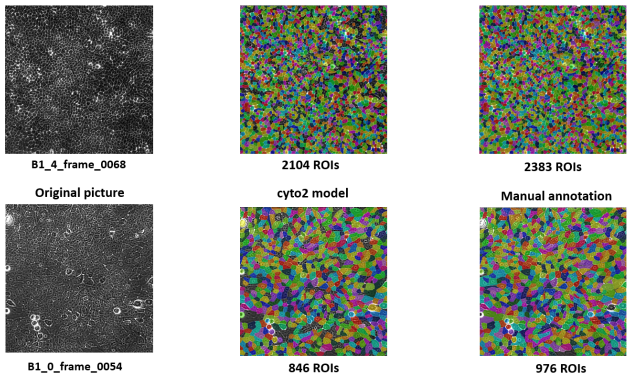

Figure 3.4: Labeling of Stationary phase cells.

3.1.1 Training validation

In our study, we trained models using two types of images: the original images and the images processed with CLAHE. These images were accompanied by corresponding annotated mask files that provided ground truth for the training process.

This setup allowed us to compare the performance of the models trained on the original images versus those trained on the CLAHE-processed images. In particular, we were interested in understanding the impact of CLAHE preprocessing on the effectiveness of the trained models.

To assess the accuracy of the different models, we employed a cross-validation approach with a training and test split. Given n images for training, we randomly selected n images out of a pool of 25 images for the training set. We then randomly selected 5 additional images from the remaining set to form the test set.

This random selection process was repeated 10 times. For each iteration, the model was trained on the selected training set and evaluated on the corresponding test set. The accuracy of the model for each iteration was recorded(table:3.1).

Finally, the accuracy values from the 10 iterations were averaged to yield the final accuracy of the model. This approach allowed us to gain a more robust estimate of the model's accuracy, accounting for variability

Table 3.1: Table of MCF10A Average precision without preprocessing

No. of Times \ No. of Images	0	1	2	3	4	5	10	15	19	20
1	0.059784	0.55479	0.618353	0.771552	0.717469	0.725682	0.519449	0.81153	0.692138	0.793142
2	0.343049	0.676033	0.789893	0.810835	0.689755	0.780583	0.765614	0.626612	0.716685	0.660858
3	0.297369	0.646625	0.719073	0.653984	0.619325	0.705854	0.752007	0.82028	0.671683	0.755927
4	0.39194	0.665027	0.642348	0.692325	0.716717	0.718591	0.78785	0.714763	0.720586	0.638437
5	0.169239	0.717071	0.717993	0.329424	0.697914	0.652429	0.722066	0.645332	0.755488	0.834234
6	0.578645	0.663415	0.665311	0.593466	0.724634	0.647641	0.391769	0.751642	0.740909	0.712418
7	0.503457	0.756406	0.753846	0.723521	0.743783	0.633457	0.744406	0.697954	0.706094	0.816996
8	0.176268	0.616039	0.472458	0.611674	0.823836	0.504305	0.732172	0.841255	0.816489	0.853089
9	0.233718	0.024588	0.527256	0.704218	0.717119	0.728587	0.650823	0.703828	0.710452	0.746428
10	0.063528	0.672908	0.761537	0.802666	0.70557	0.623774	0.720278	0.7529	0.821972	0.809417
Average Value	0.2817	0.59929	0.666807	0.669367	0.715612	0.67209	0.678644	0.73661	0.73525	0.762095
Standard Deviation	0.175254	0.208988	0.103947	0.140549	0.050496	0.077531	0.126533	0.072674	0.049985	0.073024

due to the random selection of training and test sets.

3.1.2 Intersection over Union (IoU)

Intersection over Union, often abbreviated as IoU, is a commonly used evaluation metric in image segmentation tasks. It measures the overlap between the predicted segmentation and the ground truth annotation.

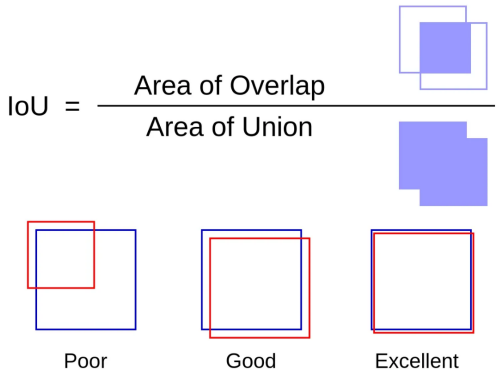

Figure 3.5: A diagram representing the Intersection over Union[12].

The IoU is defined as the size of the intersection divided by the size of the union of the predicted and ground truth segmentation masks. This can be expressed mathematically as:

$$IoU = \frac{|A \cap B|}{|A \cup B|}$$

where: A represents the predicted segmentation mask, B represents the ground truth.

The IoU score ranges from 0 to 1, where a score of 1 means perfect overlap (i.e., the prediction and ground truth are exactly the same), while a score of 0 means there is no overlap at all.

It's a robust measure as it takes into account both false positives (areas incorrectly identified as the object

of interest) and false negatives (areas of the object of interest that were missed).

3.1.3 Results

Our experimental results yielded significant insights regarding the effect of preprocessing and the number of training images on model accuracy(figure:3.6).

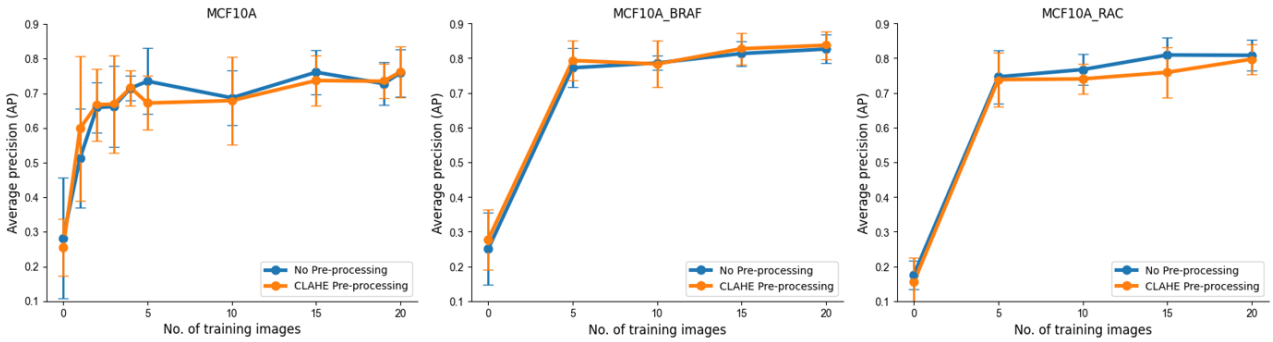

Figure 3.6: Results of Training Validation.

We found that preprocessing the images with CLAHE did not lead to an improvement in the accuracy of the model when compared to using the original, unprocessed images. This indicates that for our specific application, preprocessing might not be necessary.

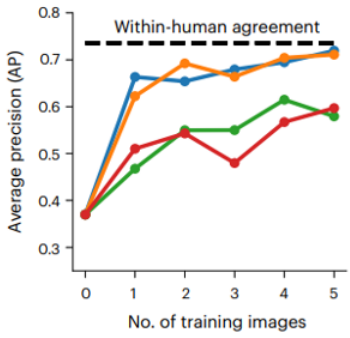

Figure 3.7: Average precision as a function of No. of images[5].

Consistent with the findings presented in the referenced paper[5], our results also indicated that a small number of images, specifically 5, was sufficient to train a model with satisfactory accuracy. This suggests that our model was capable of learning representative features from a relatively small dataset, a highly beneficial attribute considering the often limited availability of annotated biomedical images.

3.1.4 Model Application and Parameter Optimization

After training models on 25 labeled images without preprocessing of each cell type, the next step was to apply these models to segment all the images of the three cells. For each cell type, five distinct regions were imaged, yielding a total of 70 images per region for segmentation.

Before conducting the segmentation, two crucial parameters, the Flow threshold and the Mask threshold, had to be determined(figure:3.8).

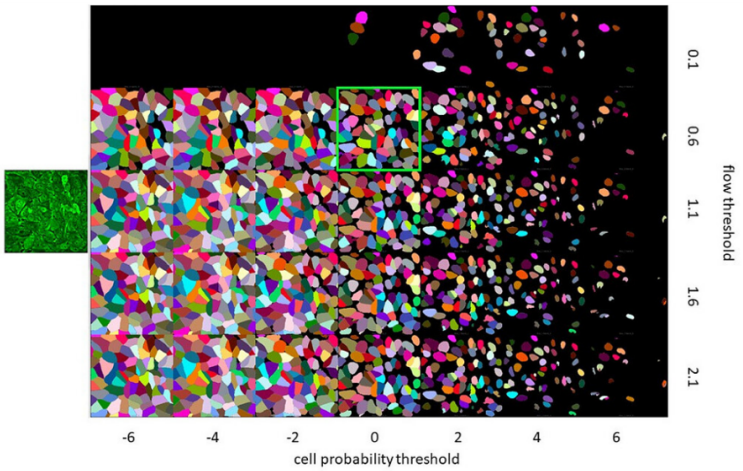

Figure 3.8: The effect of two parameters on accuracy[13].

Flow Threshold

The Flow threshold parameter controls the maximum allowed error of the flows for each mask. This setting is crucial as it can affect the shape consistency of the predicted Regions of Interest (ROIs).

During training, the network learns from image flows consistent with real shapes. However, when the network is uncertain, it may output inconsistent flows that do not correspond to any real shapes. By setting the Flow threshold, we can ensure that the shapes recovered after the flow dynamics step are consistent with real ROIs.

The default value for the Flow threshold is set to 0.4. If fewer ROIs are returned than expected, the threshold should be increased. Conversely, if too many ill-shaped ROIs are being returned, the threshold should be reduced.

Mask Threshold

The Mask threshold parameter dictates the minimum cell "probability" for a pixel to be considered for ROI determination. This cell probability is a prediction made by the network, which varies from around -6 to +6 and is input to a sigmoid function centered at zero.

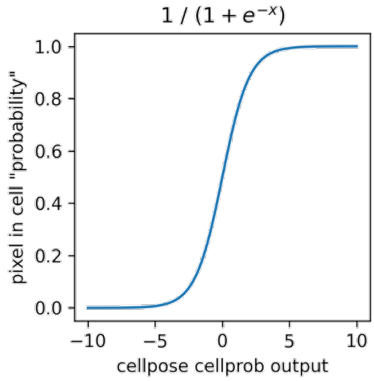

Figure 3.9: Cell probability output.

The default value for the Mask threshold is set to 0.0. If fewer ROIs are returned than expected, the threshold should be decreased. Conversely, if too many ROIs, particularly from dim areas, are being returned, the threshold should be increased.

Determination of Optimal Parameters

Through experimentation, we sought to identify the optimal values for the Flow threshold and the Mask threshold parameters that would result in the highest accuracy.

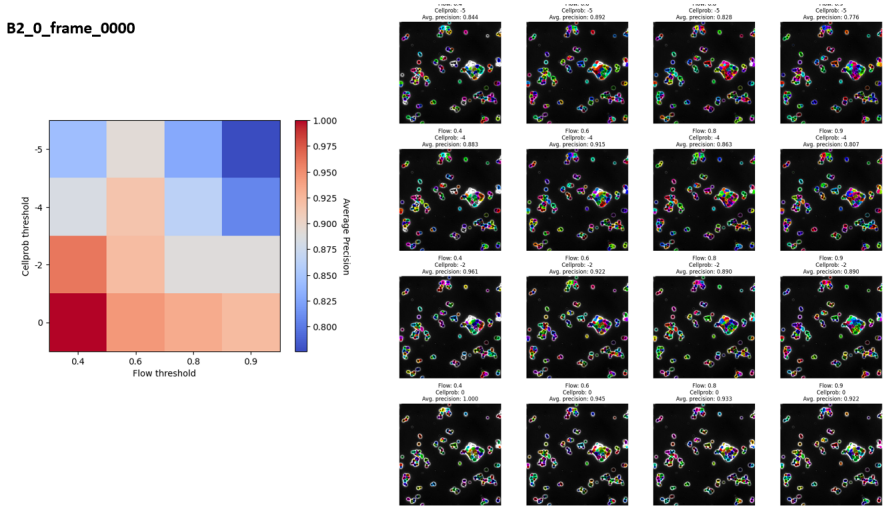

Figure 3.10: The threshold experiments in lag phase of cells.

A series of tests were performed on several images with varying values of Flow threshold and cell probability(figures: 3.10, 3.11 and 3.12). Our results demonstrated that the optimal values for the highest accuracy were consistently found to be a Flow threshold of 0.4 and a Mask threshold (cell probability) of 0. This finding held true regardless of the cycle under investigation. These optimal parameter settings served to ensure that our segmentation models were finely tuned and optimized for accuracy.

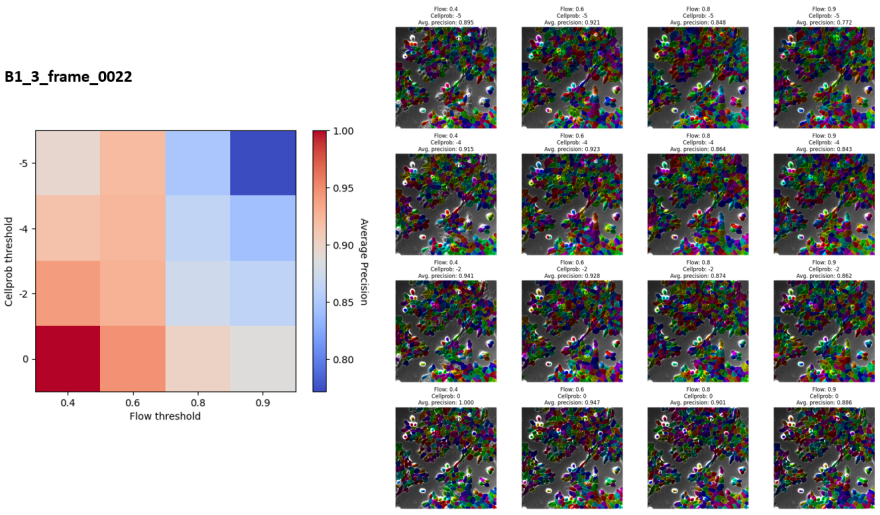

Figure 3.11: The threshold experiments in log phase of cells.

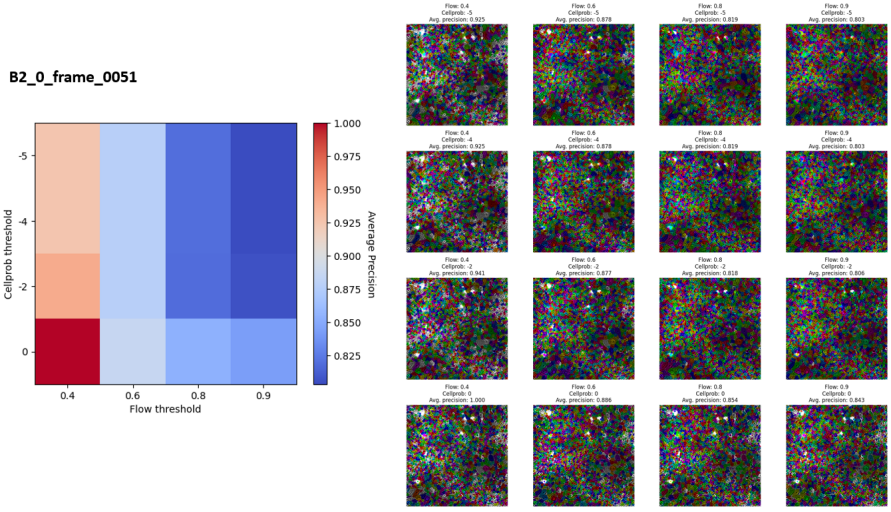

Figure 3.12: The threshold experiments in Stationary phase of cells.

3.2 Statistical Analysis

Having trained three distinct models - MCF10A, MCF10A_Braf, and MCF10A_Rac - using 25 labeled images of each cell type, we proceeded to the application phase.

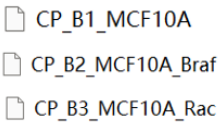

Figure 3.13: Models - MCF10A, MCF10A_Braf, and MCF10A_Rac.

These models were employed to segment all our cell images, using the optimal parameters determined earlier: Flow threshold set to 0.4, and Cell probability set to 0. The models were able to effectively segment the cell regions, delineating boundaries and identifying individual cells.

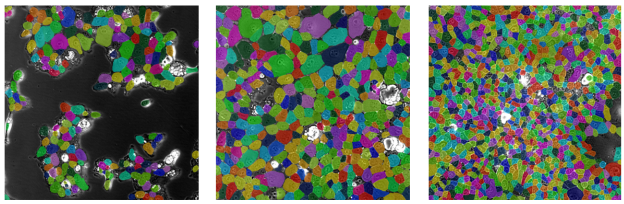

Figure 3.14: Some examples of segmentation.

Upon successful segmentation, we carried out statistical analysis on these images. Two key metrics were assessed: cell area and cell number. These metrics provide valuable insight into the cell morphology and density respectively, and their variations across different conditions[14].

3.2.1 Analysis of Cell Area Over Time

For each time point, we computed the average cell area across all cells in the image. This process was repeated for all images to generate a timeline of average cell area. The resulting data was then plotted to visualize the changes in cell area over time, as depicted in Figure 3.15.

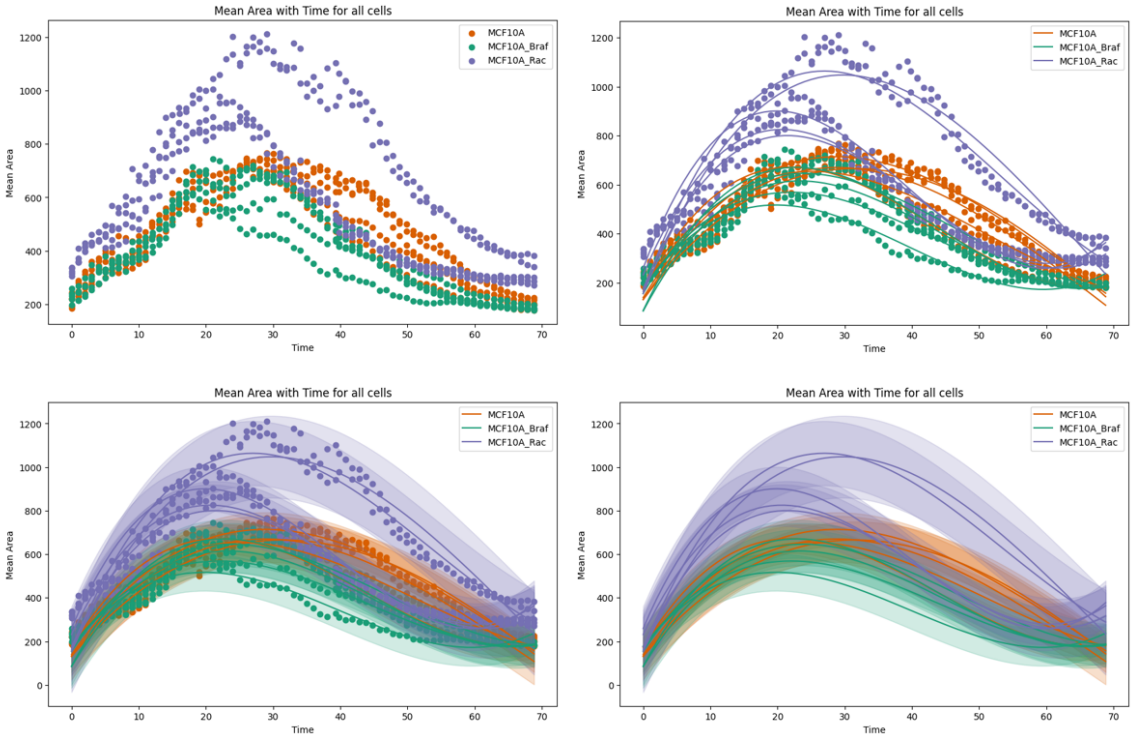

Figure 3.15: Cell Mean Area with Time.

To better understand the trend of the data, we fitted the data points with a cubic function: $y = ax^3 + bx^2 + cx + d$. This approach allowed us to model the potential nonlinear behavior of cell growth over time. The fitting results are also shown in Table 3.2.

	MCF10A				MCF10A_Braf				MCF10A_Rac			
	a	b	c	d	a	b	c	d	a	b	c	d
0	0.0024	-0.6888	36.3152	130.7171	0.011	-1.3042	38.6966	176.7749	0.0166	-2.071	65.9205	175.1953
1	0.005	-0.9841	44.1371	138.7425	0.0112	-1.4968	51.618	83.7941	0.0204	-2.438	72.6686	258.8071
2	0.0074	-1.1259	42.6385	176.6137	0.0093	-1.3122	46.5621	155.3754	0.007	-1.428	66.3045	154.4975
3	0.0119	-1.5579	51.2702	171.3022	0.0131	-1.7224	58.1146	85.9344	0.0115	-1.8602	75.138	163.5699
4	0.0044	-0.9024	41.082	130.7608	0.0092	-1.2171	40.1413	176.501	0.0174	-2.1157	65.022	231.1507
Avg	0.00622	-1.05182	43.0886	149.6273	0.01076	-1.41054	47.02652	135.676	0.01458	-1.98258	69.01072	196.6441

Table 3.2: Coefficients and averages for MCF10A, MCF10A_Braf, and MCF10A_Rac

Apart from the average cell area, we also investigated the distribution of cell areas at each time point. The distributions were then visualized using a three-dimensional (3D) plot. This representation allows for a more nuanced understanding of the cell area distribution at each time point, as it incorporates both the size and frequency of cells into a single visualization. The resulting 3D plot (Figure 3.16 and 3.17) provides an overview of the cell area distribution over time.

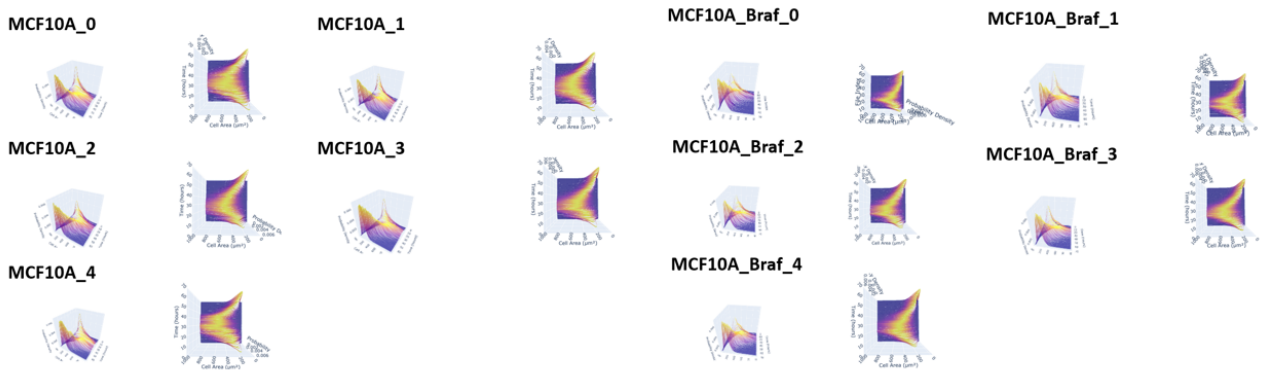

Figure 3.16: 3D plot - MCF10A and MCF10A_Braf.

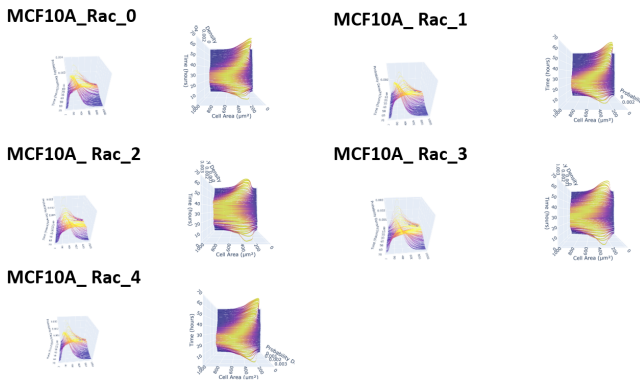

Figure 3.17: 3D plot - MCF10A_Rac.

Our analysis reveals intriguing trends in cell area over time. It was observed that the cell area tends to increase initially and then decrease as time progresses, which could be indicative of various biological processes such as cell division or changes in cell morphology.

Comparative analysis of the MCF10A, MCF10A_Braf, and MCF10A_Rac cell types revealed notable differences. MCF10A and MCF10A_Braf demonstrated similar cell area distributions over time, suggesting comparable growth and division patterns in these cell types.

In contrast, MCF10A_Rac cells exhibited a distinct trend. They not only had a larger cell area on average, but also a more dispersed cell area distribution over time. This increased dispersion indicates a higher variability in cell size for MCF10A_Rac cells, potentially reflecting distinct cellular behaviors.

3.2.2 Analysis of Cell Number Over Time

In addition to the cell area, we also analyzed the trend in cell numbers over time(figures: 3.18). We plotted the cell count data over time and fitted these curves using a logistic function, which is a common model for population growth dynamics[15].

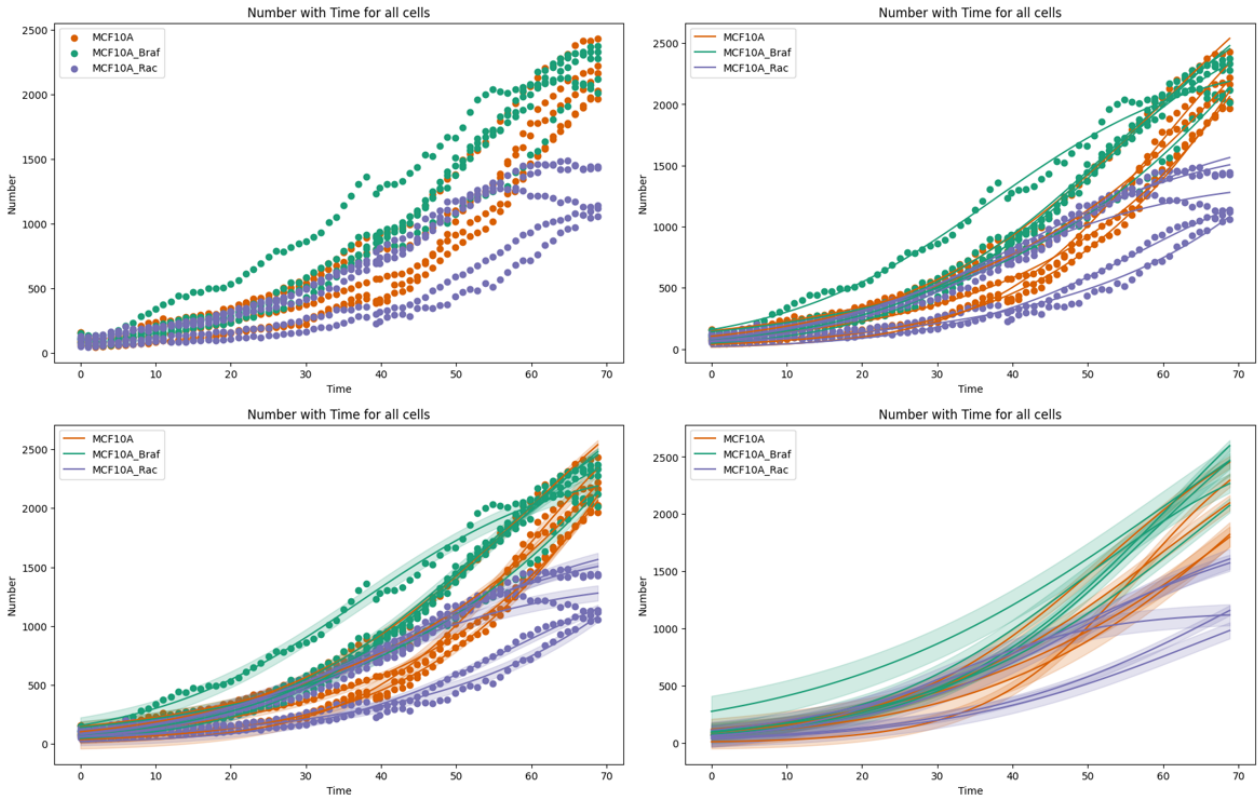

Figure 3.18: Cell Number Over Time.

The logistic function is defined as follows:

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}} \tag{3.1}$$

where L is the maximum population size (carrying capacity), k is the growth rate, and x_0 is the x-value of the sigmoid's midpoint, which corresponds to the point of fastest growth.

To fit the logistic function to our data, we employed a genetic algorithm (GA)(there are also annealing algorithms[16] and so on[17]) using the Distributed Evolutionary Algorithms in Python (DEAP) library. Each individual in the GA population represented a set of parameters for the logistic function (L, k, x_0). The GA was configured to use tournament selection, two-point crossover, and Gaussian mutation.

Our fitness function was the sum of squared residuals between the observed cell numbers and the numbers predicted by the logistic function. By minimizing this fitness function, the GA could effectively fit the logistic function to the cell number data, giving us a model of cell population growth over time. The fitting results are also shown in Table 3.3.

	MCF10A			MCF10A_Braf			MCF10A_Rac		
	L	k	x0	L	k	x0	L	k	x0
0	4305.79	0.0547	74.47	4248.44	0.0435	61.29	2027.79	0.0592	47.85
1	4512.59	0.0464	77.66	3644.23	0.0692	58.19	2020.49	0.0692	49.16
2	3310.85	0.0598	59.66	3855.42	0.0547	66.01	2455.00	0.0564	70.81
3	3202.51	0.0718	52.29	4153.86	0.0654	60.98	1737.70	0.0594	64.38
4	2968.19	0.0997	56.53	2836.05	0.0694	48.90	1147.77	0.0958	30.71
Avg	3659.00	0.06648	64.12	3747.60	0.06044	59.07	1877.75	0.06800	52.58

Table 3.3: Logistic Parameters and averages for MCF10A, MCF10A_Braf, and MCF10A_Rac

ANOVA Test

To test the hypothesis that the logistic function parameters (L, k, x_0) are the same across all three cell groups, an Analysis of Variance (ANOVA) was performed.

The results were as follows:

- L: F = 16.091605304148214, p = 0.0004013690218431329
- k: F = 0.2924956710484599, p = 0.7515708883701422
- x_0: F = 1.2149188084393887, p = 0.33076443998359134

For the parameter L, the p-value was less than 0.05, indicating a statistically significant difference in the means of the L parameter across the three cell groups.

However, for the parameters k and x_0, the p-values were greater than 0.05, suggesting that there is no statistically significant difference in the means of these parameters across the three cell groups.

Independent T-Test

As there was a significant difference in the L parameter, post hoc pairwise comparisons were conducted using independent t-tests to further investigate this difference. The results were as follows:

- MCF10A vs MCF10A_Braf (L): t = -0.21825611954870167, p = 0.8326941280081706
- MCF10A vs MCF10A_Rac (L): t = 4.694691376227137, p = 0.0015520784101948667
- MCF10A_Rac vs MCF10A_Braf (L): t = -5.640745264236826, p = 0.0004866208187123805

These results indicate a statistically significant difference in the L parameter between MCF10A and MCF10A_Rac, and between MCF10A_Rac and MCF10A_Braf. However, there was no significant difference in the L parameter between MCF10A and MCF10A_Braf.

Our analysis revealed that the cell number dynamics could be appropriately modeled by a logistic function, illustrating the characteristic sigmoidal curve of population growth and carrying capacity.

When comparing the fitted curves for the three cell groups, MCF10A and MCF10A_Braf demonstrated similar variation patterns. Their respective curves exhibited analogous rates of increase and subsequent plateau, suggesting a resemblance in their growth behavior and carrying capacity (L parameter).

On the contrary, MCF10A_Rac's cell count was comparatively less, with the logistic function revealing a lower carrying capacity. The reduced cell number, indicated by a lower L parameter in the logistic function, suggests a different growth dynamic for MCF10A_Rac compared to MCF10A and MCF10A_Braf.

Chapter 4

Convolutional Autoencoder

In our study, we used a convolutional autoencoder and K-Means clustering to classify cell images into two categories: images with few cells and images with many cells. The goal is to facilitate the subsequent application of different models to improve accuracy based on the density of cells in the images.

Data Preprocessing

The cell images were stored in a directory on our local machine. We looped over each file in the directory, opened it, resized it to a desired size of 256x256 pixels, and then normalized the pixel intensity values to a range of [0,1]. This process was designed to ensure consistency among the images and to scale the pixel values, which aids in the training process of the neural network.

Convolutional Autoencoder

We defined a convolutional autoencoder with an encoder and a decoder part. The encoder consisted of four convolutional layers, each followed by a ReLU activation function. The decoder consisted of four transposed convolutional layers, each followed by a ReLU activation function, except for the last layer which used a sigmoid activation function to output values between 0 and 1.

The purpose of the autoencoder is to learn a lower-dimensional and dense representation of the images, which can be used to capture the main characteristics of the cells' distribution in the images.

Training

We used the Mean Squared Error (MSE) loss and the Adam optimizer with a learning rate of 0.001 for training. The training loop involved the following steps for a fixed number of epochs:

- Forward propagation of the input through the network

- Computation of the loss
- Backward propagation of the loss gradients
- Update of the network weights

Encoding and Clustering

Once the autoencoder was trained, we used the encoder part to encode all the images into a lower-dimensional space. These encodings were then passed to a K-Means clustering algorithm which classified them into two clusters(figures: 4.1).

The purpose of the K-Means clustering is to group the cell images into two categories based on their cell density, i.e., few cells versus many cells.

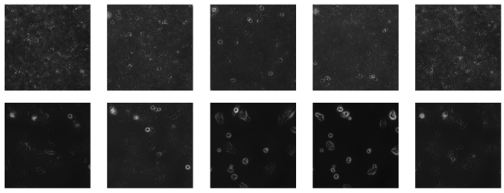

Figure 4.1: Classification result.

Visualization

Finally, we used t-SNE, a dimensionality reduction technique particularly effective for visualizing high-dimensional data, to plot our encoded images in a 2D space(figures: 4.2).

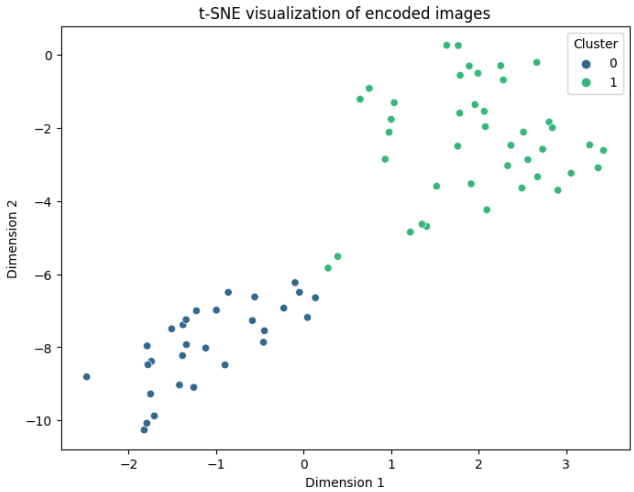

Figure 4.2: t-SNE.

Chapter 5

Conclusion

The project successfully demonstrated the potential of deep learning in cellular image analysis, particularly in the context of ECIS. Experimental results revealed that preprocessing the images with CLAHE did not significantly improve the model's accuracy. Additionally, it was found that a small number of images, specifically 5, was sufficient to train a model with satisfactory accuracy. This suggests that the model was capable of learning representative features from a relatively small dataset, a highly beneficial attribute considering the often limited availability of annotated biomedical images.

The study also found that cell area and cell numbers both increase and then decrease with time, fitting these changes with a cubic function and a logistic function, respectively. Comparisons between MCF10A, MCF10A_Braf, and MCF10A_Rac revealed distinct patterns in cell area distribution and number changes. While MCF10A and MCF10A_Braf showed similar cell area distribution and number changes, MCF10A_Rac exhibited a larger cell area, a more dispersed cell area over time, and fewer cells. These findings advance our understanding of ECIS and its applications in biological and medical research, particularly in the context of deep learning-based cellular image analysis.

The project not only improved my technical proficiency in building and training machine learning models, but it also developed my problem-solving skills, particularly in the realm of data preprocessing and feature selection. Working with biomedical image data, a type of data with which I was initially unfamiliar, was a stimulating challenge. Not only did it push me to learn new concepts and techniques, but it also helped me appreciate the nuances and complexities that can arise in real-world data science projects.

One of the crucial takeaways from this internship was the appreciation of the intersection of AI and biomedical research. As AI is becoming increasingly important in many fields, including biology and medicine, the demand for such cross-disciplinary expertise is growing. I am excited by the potential for machine learning and artificial intelligence to revolutionize healthcare, from diagnosis to treatment strategies.

Bibliography

[1] I. Giaever and C. R. Keese, "Monitoring fibroblast behavior in tissue culture with an applied electric field.", *Proceedings of the National Academy of Sciences*, vol. 81, no. 12, pp. 3761–3764, Jun. 1984.

[2] V. Viswam, R. Bounik, A. Shadmani, J. Dragas, C. Urwyler, J. A. Boos, M. E. J. Obien, J. Muller, Y. Chen, and A. Hierlemann, "Impedance Spectroscopy and Electrophysiological Imaging of Cells With a High-Density CMOS Microelectrode Array System", *IEEE Transactions on Biomedical Circuits and Systems*, vol. 12, no. 6, pp. 1356–1368, Dec. 2018.

[3] P. Messina, C. Garcia, J. Rambeau, J. Darcourt, R. Balland, B. Carreel, M. Cottance, E. Gusarova, J. Lafaurie-Janvore, G. Lebedev, F. Bozsak, A. I. Barakat, B. Payrastre, and C. Cognard, "Impedance-based sensors discriminate among different types of blood thrombi with very high specificity and sensitivity", *Journal of NeuroInterventional Surgery*, vol. 15, no. 6, pp. 526–531, Jun. 2023.

[4] A. R. A. Rahman, J. Register, G. Vuppala, and S. Bhansali, "Cell culture monitoring by impedance mapping using a multielectrode scanning impedance spectroscopy system (CellMap)", *Physiological Measurement*, vol. 29, no. 6, S227–S239, Jun. 2008.

[5] C. Stringer, T. Wang, M. Michaelos, and M. Pachitariu, "Cellpose: A generalist algorithm for cellular segmentation", *Nature Methods*, vol. 18, no. 1, pp. 100–106, Jan. 2021.

[6] M. Pachitariu and C. Stringer, "Cellpose 2.0: How to train your own model", *Nature Methods*, vol. 19, no. 12, pp. 1634–1641, Dec. 2022.

[7] Y. Qu, B. Han, Y. Yu, W. Yao, S. Bose, B. Y. Karlan, A. E. Giuliano, and X. Cui, "Evaluation of MCF10A as a Reliable Model for Normal Human Mammary Epithelial Cells", *PLoS ONE*, vol. 10, no. 7, e0131285, Jul. 2015.

[8] N. Xu, X. Chen, J. Rui, Y. Yu, D. Gu, J. J. Ruan, B. H. Ruan, N. Xu, X. Chen, J. Rui, Y. Yu, D. Gu, J. J. Ruan, and B. H. Ruan, "Cell Growth Measurement", in *Cell Growth*, IntechOpen, Mar. 2020.

[9] K. Zuiderveld, "Contrast limited adaptive histogram equalization", *Graphics gems*, pp. 474–485, 1994.

[10] A. Horé and D. Ziou, "Image Quality Metrics: PSNR vs. SSIM", in *2010 20th International Conference on Pattern Recognition*, Aug. 2010, pp. 2366–2369.

[11] F. Thenier, E. Angelini, and B. Roellinger, "Segmentation of Microfluidic Images using Deep-Learning - Focusing on Image enhancement and domain transfer challenges",

[12] N. Tomar, *What is Intersection over Union (IoU) in Object Detection?*, https://idiotdeveloper.com/what-is-intersection-over-union-iou/, Feb. 2023.

[13] *CellPose flow and cell threshold*, https://forum.image.sc/t/cellpose-flow-and-cell-threshold/70347/4, Sep. 2022.

[14] H. G. Kilian, D. Bartkowiak, D. Kaufmann, and R. Kemkemer, “The General Growth Logistics of Cell Populations”, *Cell Biochemistry and Biophysics*, vol. 51, no. 2-3, p. 51, Jul. 2008.

[15] D. E. Wachenheim, J. A. Patterson, and M. R. Ladisch, “Analysis of the logistic function model: Derivation and applications specific to batch cultured microorganisms”, *Bioresource Technology*, vol. 86, no. 2, pp. 157–164, Jan. 2003.

[16] S. P. Brooks, N. Friel, and R. King, “Classical Model Selection via Simulated Annealing”, *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 65, no. 2, pp. 503–520, May 2003.

[17] W. S. DeSarbo, R. L. Oliver, and A. Rangaswamy, “A simulated annealing methodology for clusterwise linear regression”, *Psychometrika*, vol. 54, no. 4, pp. 707–736, Sep. 1989.