AUTOMATED CONFLUENCY OF FIBROBLAST SKIN CELL DETECTION USING U-NET BASED NETWORKS

BY

MUHAMMAD HAFIZI BIN ABDUL MALIK

A thesis submitted in fulfillment of the requirement for the degree of Master of Science in Engineering

Kulliyyah of Engineering
International Islamic University Malaysia

DECEMBER 2023

ABSTRACT

Fibroblast cell culture monitoring necessitates thorough attention for the continuous characterization of cultivated cells. Deep learning has recently emerged to engage in a process, such as a microscopy segmentation task; however, the trained data may not be comprehensive for other datasets. Most algorithms do not encompass a wide range of data attributes and require distinct system workflows. To address this, the research proposed a U-Net based pipeline specifically for detecting and segmenting fibroblast cell confluency at varying magnifications and distributions. U-Net is a widely recognized deep learning architecture commonly employed for biomedical segmentation tasks. Patch-based segmentation was employed for predictions, with three U-Net based networks proposed using inception, dilation, and residual mechanisms. Additionally, a smooth blending technique was introduced to address edge effects arising from patch-based segmentation. Model B, with an inception mechanism of various kernel sizes, demonstrated the highest overall IoU score of 0.707, with a mean relative performance of 6.5% across all datasets. It also outperformed human observation by 15% more accuracy through validation analysis. However, when disregarding deployment considerations, Model C, featuring a dilation mechanism, emerges as a viable alternative, offering optimal performance and moderate complexity compared to Model B during training sessions. Another main objective of this study is to fully automate cell monitoring, traditionally involving high human interaction. Thus, a complete design of automated cell culture monitoring was also designed for a specified working space incorporating AI-based cell screening, IoT-based monitoring and communication system through Cloud Firebase, and sample handling automation using robotic arm. OpenVINO and NCS2 were used to optimize the segmentation on Raspberry Pi where a cell image can be segmented in under 9 seconds by Model B. Overall, the proposed design for automated cell culture monitoring demonstrates its efficiency, with the complete cycle of its workflow taking under 1 minute and an average power rate of 12.2W, while requiring just approximately 200MB of downloaded data for a full day of operation in a real-time application. However, the high precision of the proposed model is currently limited to around 80% confluency and the implementation of the automation system has yet to be adjusted to accommodate varying lab structures. The primary contribution of this study revolves around providing guidance in refining existing models to adapt diverse magnification levels of microscopy images under specific conditions. Furthermore, the integration of Raspberry Pi and OpenVINO in biomedical tasks, a relatively unexplored area, adds a distinctive aspect of this research. Consequently, this study introduces a modular, comprehensive, and optimized cell imaging system with the ability to integrate various external systems, which is off the shelf, enabling cost-effective automated cell culture monitoring.

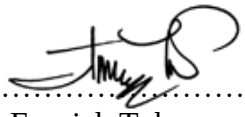
ملخص البحث

تتطلب مراقبة زراعة الخلايا الليفية اهتماما شاملا للتوصيف المستمر للخلايا المزروعة. ظهر التعلم العميق مؤخرا للانخراط في عملية مثل مهمة تجزئة الفحص المجهري. ومع ذلك ، قد لا تكون البيانات المدربة شاملة لمجموعات البيانات الأخرى. لا تشمل معظم الخوارزميات نطاقًا واسعًا من سمات البيانات وتتطلب مهام سير عمل متميز للنظام. لمعالجة هذه المشكلة، اقترح هذا البحث خط أنابيب قائم على شبكة U-Net خصيصا وتجزئة تقارب الخلايا الليفية بتكبيرات وتوزيعات مختلفة. U-Net هي عبارة عن بنية تعلم عميق معترف بها على نطاق واسع تستخدم عادة لمهام التجزئة الطبية الحيوية. تم استخدام التجزئة القائمة على التصحيح للتنبؤات، مع ثلاث شبكات قائمة عل U-Net مقترحة باستخدام آليات البدء والتمدد والمتبقية. بالإضافة إلى ذلك، تم تقديم تقنية مزج سلسة لمعالجة تأثيرات الحافة الناشئة عن التجزئة القائمة على التصحيح. أظهر النموذج B ، مع آلية البدء بأحجام نواة مختلفة، أعلى درجة إجمالية ل IoU بلغت 0.707، بمتوسط أداء نسبي يبلغ 6.5٪ عبر جميع مجموعات البيانات. كما تفوقت على الملاحظة البشرية بدقة اكبر بنسبة 15٪ من خلال تحليل التحقق من الصحة. ومع ذلك، عند تجاهل اعتبارات التوزيع، يظهر النموذج C الذي يتميز بآلية التوسيع كبديل قابل للتطبيق، حيث يقدم الأداء الأمثل والتعقيد المعتدل مقارنة بالنموذج B أثناء جلسات التدريب. الهدف الرئيسي الآخر لهذا البحث هو أتمتة مراقبة الخلايا بشكل كامل، والتي كانت تتطلب في السابق مستوى كبيرًا من الاتصال البشري. ونتيجة لذلك، تم إنشاء تصميم كامل للمراقبة الآلية لثقافة الخلايا لمساحة عمل محددة تتضمن فحص الخلايا القائم على الذكاء الاصطناعي ونظام المراقبة والاتصالات القائم على (IoT) باستخدام Cloud Firebase، وأتمتة معالجة العينات عبر الذراع الآلية. تم

استخدام OpenVINO وNCS2 لتحسين التجزئة على Raspberry Pi حيث يمكن تجزئة صورة الخلية في أقل من 9 ثوانٍ بواسطة النموذج B. بشكل عام، يوضح التصميم المقترح لمراقبة ثقافة الخلية الآلية كفاءته، حيث تستغرق الدورة الكاملة لسير العمل أقل من دقيقة واحدة ومتوسط معدل طاقة يبلغ 12.2 واط ، بينما يتطلب ما يقرب من 200 ميجابايت فقط من البيانات التي تم تنزيلها لمدة يوم كامل من التشغيل في تطبيق في الوقت الفعلي. ومع ذلك، فإن الدقة العالية للنموذج المقترح تقتصر حاليًا على حوالي 80٪ من التقارب ولم يتم بعد تعديل تنفيذ نظام التشغيل الآلي لاستيعاب هياكل المختبرات المختلفة. تدور المساهمة الأساسية لهذه الدراسة حول توفير التوجيه في تحسين النماذج الحالية لتكييف مستويات التكبير المتنوعة للصور المجهرية في ظل ظروف محددة. علاوة على ذلك ، فإن دمج Raspberry Pi و OpenVINO في المهام الطبية الحيوية، وهي منطقة غير مستكشفة نسبيا ، يضيف جانبا مميزا من هذا البحث. وبالتالي ، تقدم هذه الدراسة نظاما معياريا وشاملا ومحسنا لتصوير الخلايا مع القدرة على دمج أنظمة خارجية مختلفة ، وهو جاهز للاستخدام، مما يتيح مراقبة ثقافة الخلايا الآلية الفعالة من حيث التكلفة.

APPROVAL PAGE

I certify that I have supervised and read this study and that in my opinion, it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of Master of Science in Engineering



……..
Siti Fauziah Toha
Supervisor

……..
Azhar Mohd Ibrahim
Co-Supervisor

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of Master of Science in Engineering

……..
Zulkifli Zainal Abidin
Internal Examiner

……..
Mohd Fua'ad Rahmat
External Examiner

This thesis was submitted to the Department of Mechatronics Engineering and is accepted as a fulfilment of the requirement for the degree of Master of Science in Engineering

……..
Ali Sophian
Head, Department of
Mechatronics Engineering

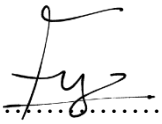
This thesis was submitted to the Kulliyyah of Engineering and is accepted as a fulfilment of the requirement for the degree of Master of Science in Engineering

……..
Sany Izan Ihsan
Dean, Kulliyyah of Engineering

DECLARATION

I hereby declare that this dissertation is the result of my own investigations, except where otherwise stated. I also declare that it has not been previously or concurrently submitted as a whole for any other degrees at IIUM or other institutions.

Muhammad Hafizi Bin Abdul Malik



…….. 10 December 2023

Signature Date

INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

DECLARATION OF COPYRIGHT AND AFFIRMATION OF FAIR USE OF UNPUBLISHED RESEARCH

AUTOMATED CONFLUENCY OF FIBROBLAST SKIN CELL DETECTION USING U-NET BASED NETWORKS

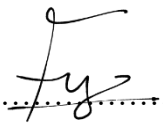
I declare that the copyright holder of this thesis/ dissertation are jointly owned by the student and IIUM.

No part of this unpublished research may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without prior written permission of the copyright holder except as provided below

1. Any material contained in or derived from this unpublished research may only be used by others in their writing with due acknowledgement.
2. IIUM or its library will have the right to make and transmit copies (print or electronic) for institutional and academic purpose.
3. The IIUM library will have the right to make, store in a retrieval system and supply copies of this unpublished research if requested by other universities and research libraries.

By signing this form, I acknowledged that I have read and understand the IIUM Intellectual Property Right and Commercialization policy.

Affirmed by Muhammad Hafizi Bin Abdul Malik



…….. ……10 December 2023…..

Signature Date

ACKNOWLEDGEMENTS

All glory is due to Allah, the Almighty, whose Grace and Mercies have been with me throughout the duration of my programme. Although, it has been tasking, His Mercies and Blessings on me ease the herculean task of completing this thesis.

I am most indebted to my supervisor, Prof. Ir. Dr Siti Fauziah Toha, whose enduring disposition, kindness, promptitude, thoroughness and friendship have facilitated the successful completion of my work. I put on record and appreciate his detailed comments, useful suggestions and inspiring queries which have considerably improved this thesis. I am also grateful to my co-supervisor, Asst. Prof. Dr. Azhar Mohd Ibrahim, and my external supervisor, Asst. Prof. Ir. Dr. Ahmad Syahrin Idris, from University of Southampton Malaysia, whose support and cooperation contributed to the outcome of this work. I am also thankful to our research collaborators, Dr Muhammad Fauzi Daud from Institute of Medical Science Technology at University Kuala Lumpur and Dr Izyan Mohd Idris from Institute of Medical Research at National Institutes of Health Malaysia for providing resources, equipment, and guidance needed to complete this study.

Lastly, my gratitude goes to my parents for their prayers and understanding of my study journey.

Once again, we glorify Allah for His endless mercy on us one of which is enabling us to successfully round off the efforts of writing this thesis. Alhamdulillah.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 GENERAL BACKGROUND

Skin is the largest organ in the human body and also serves as a barrier against outside pollutants (Kaur et al., 2019). It is mostly composed of the protein collagen, which is produced by fibroblast cells. When the skin is injured, the wound-healing process stimulates the production of new fibroblasts, which create collagen to close the wound. However, when there is extensive skin injury, such as from burns, chronic wounds, or infections, the natural healing mechanism is incapable of regenerating the affected tissue. Therefore, fibroblast cell culture is critical for growing wound grafts, as illustrated in Figure 1.1, and helping tissue regeneration in surgery.

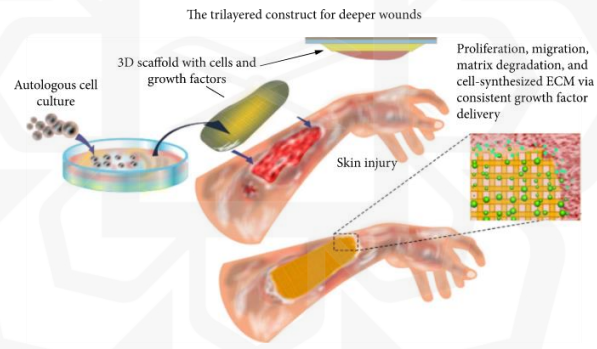


Figure 1.1 Graft Production from Skin Cell Culture (Kaur et al., 2019)

Cell culture is a significant tool that has served as a gold standard in many fields of study, including tissue engineering, drug development, and diagnostics. It refers to the process of extracting human, animal or insect cells and growing them in favorable artificial environment. Current cell culture workflows rely on manual operations which are laborious and tedious, requiring the lab personnel to pay close attention to ensure the cells are sub-cultured at the appropriate stage during monitoring phase. Figure 1.2 shows cells theoretically grow over extended periods of time, and their growth rate is measured by confluency, which indicates the total percentage of cells within a sample. Ideally, 80% confluency is targeted for optimized

and continuous cell reproduction. If the confluency is over the limit, they will die. As a result, to compensate for the reasons, the process necessitates the employment of powerful automated image analysis technology.

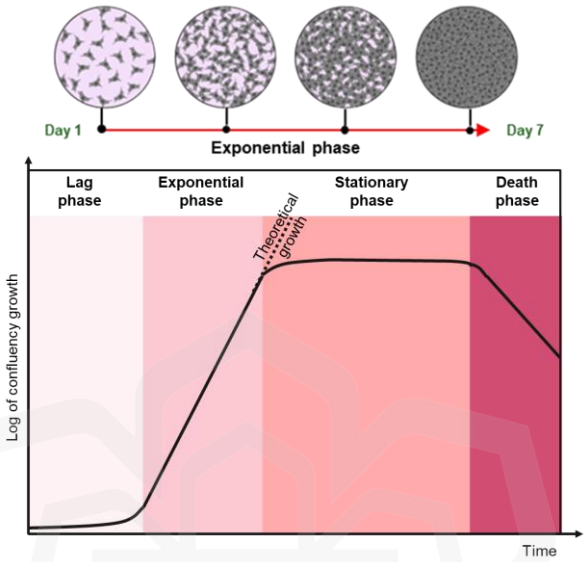


Figure 1.2 Life Cycle of Cell Culture

It is now much easier to analyze label-free images thanks to breakthrough automated image analysis technologies based on artificial intelligence in the 1980s and greater advancement in the 2010s. To visually analyze data in the artificial intelligence sector, three approaches are commonly used: image classification, object detection, and image segmentation. Image classification helps to classify the entire image as one of the classes. Object detection specifies the location of multiple objects in the image, usually with rectangular bounding boxes. Image segmentation classifies objects by pixels, precisely segmenting them with the background. The second and third are most likely to be considered to assist medical imaging analysis, depending on the focal problems.

1.2 PROBLEM STATEMENT

Efforts have been made in recent years to automate cell segmentation determining the appropriate time for cell passaging using artificial intelligence techniques; however, not all of them are perfect for diverse instances, such as for varying color images and cell shapes. Historically, measurement of cell confluency is based solely on visual observation under microscope. Yet, human eyes are prone to optical illusions, as stated by Korzynska et al. (2021), and ineffective and unstable at measuring something quantitatively. Consequently, the technique leads to inconsistent results especially when involving various interuser. Thus, this traditional technique is heavily reliant on human expertise and time as it is highly subjective, mentioned by Mota et al. (2021), and time-consuming. U-Net, a deep learning-based segmentation algorithm, has been shown to perform well on biomedical images such as cells and offers high possibility for improvements due to its simpler architecture. Despite the performance of U-Net and its variants, Punn and Agarwal (2022), and Siddique et al. (2021) highlighted that the major limitation of reducing the error for deep learning techniques is computational power, which limits the feasibility of the approach. Thus, comprehensive analysis of U-Net based model performance while accounting for its complexity is essential for optimizing the application among the end users in a newly developed cell culture system. Another significant problem lies in the labor-intensive nature of traditional cell culture processes, demanding close and repetitive monitoring on the cells over extended periods. Despite a few existing modern systems being introduced, they are highly costly and not suitable for small-scale or immediate research endeavors. Therefore, the development of a new automated cell culture system is crucial to make it accessible for small research groups and urgent situations, such as the sudden emergence of COVID-19 pandemic in year 2019.

1.3 RESEARCH OBJECTIVES

The research is mainly to reduce user bias in measuring and segmenting fibroblast cell growth of different magnification and confluency levels in microscopy images. To achieve that, the objectives are:

1. To model and refine fibroblast cell segmentation pipelines for a precise and consistent measurement at varying confluency and magnification levels.
2. To develop a comprehensive evaluation framework to assess the efficacy of the segmentation models on the datasets during both training and inferencing, considering the complexity of each model.
3. To analyze and validate the robustness of the segmentation models against manual human observation, and towards real-world applications.
4. To design an automated cell culture monitoring system, and to evaluate its feasibility in conjunction with the deployed segmentation models.

1.4 RESEARCH METHODOLOGY

A tailored methodology has been developed to precisely accomplish the predefined objectives. Objective 1 encompasses specific tasks such as image processing, selection of training settings, the implementation of a U-Net based network for modeling, and the subsequent training of data. Objective 2 focuses on monitoring training graphs, evaluating average metrics, observing visual segmentation of the proposed models, and comparing their performance against the baseline U-Net architecture. While previous objectives highlight on confluency segmentation precision, the evaluation extends to validating the performance of chosen models also for confluency value predictions against human subjective predictions and towards the natural fibroblast cell growth cycle. This comparison is executed through an online survey and a corresponding lab experiment to meet the requirements of Objective 3.

To verify the feasibility of the proposed models for real-world deployment, integration into a selected edge device was performed. This step led to the creation of a user-friendly and portable device capable of cell segmentation. Additionally, these actions were extended to include a mobile app-based remote monitoring system for

users who have been at a distance and a robotic arm-based sample handling system for automated cell culture replacement. The combination of these three systems ensures the fulfillment of Objective 4.

The primary outcomes anticipated from this methodology encompass the assessment of model complexity, relative performance comparison, real-world robustness evaluation, and the efficiency of the entire system workflow. The schematic representation of this research methodology is delineated in Appendix A.

1.5 SCOPE AND LIMITATIONS OF RESEARCH

The scope of this research is as follows:

1. Sample population: Fibroblast cell growth images obtained from a phase-contrast microscope, with three different magnification levels depending on the number of cells plated for two days.
2. Expected outcome: The study only interests on the fibroblast cell growth, thus, binary segmentation would be sufficient to represent the variable.
3. Method: Only U-Net Convolutional Neural Network (CNN) is experimented for the segmentation due to the proven adaptation of its simpler architecture on a relatively small number of biomedical images and allowing various possible refinements.

The limitation of this research is as follows:

1. Segmentation precision: The high precision for segmenting the cell growth is only guaranteed up to 80% confluency value as it portrays the maximum value of the cell growth within a petri dish and ready for passaging based on the cell culture experts and collaborators.
2. Hardware usage: The study strictly requires the Intel Movidius Neural Compute Stick (NCS2) because the OpenVINO framework is exclusively compatible with Intel processors, ensuring an optimized segmentation pipeline on Raspberry Pi.

1.6 THESIS ACHIEVEMENTS

The papers published as main author:

1. Malik, H., Idris, A. S., Toha, S. F., Idris, I. M., Daud, M. F., & Azmi, N. L. (2023). A review of open-source image analysis tools for mammalian cell culture: algorithms, features and implementations. PeerJ Computer Science, vol. 9, e1364.
2. Malik, H., Idris, A. S., Toha, S. F., Idris, I. M., Daud, M. F., & Tokhi, M. O. (2023). Deploying patch-based segmentation pipeline for fibroblast cell images at varying magnifications. IEEE Access, vol. 11, 98171-98181.

The paper published as co-author:

1. Shamhan, M., Idris, A. S., Toha, S. F., Daud, M. F., Idris, I. M., & Malik, H. (2023). An automated approach for fibroblast cell confluency characterisation and sample handling using AIoT for bio-research and bio-manufacturing. Cogent Engineering, vol. 10, no. 1.

The awards achieved for this research work:

1. Third Place in ZTE IoT Innovation Challenge 2022
2. First Place in Intel Industry-University Challenge 2022
3. Best Technical Awards in Intel Industry-University Challenge 2022
4. First Place in Petrosains RBTX Innovation Varsity 2021

1.7 THESIS ORGANIZATION

Chapter 1 discusses the introductory points regarding the importance of skin cell culture for tissue engineering and diagnostics, the challenges of manual cell culture workflows as well as the benefits of automated image analysis technology.

Chapter 2 reviews existing segmentation approaches and refinements and the recent development of cell image analysis tools and automated cell culture systems, including the highlights of their gaps.

The research methodology, detailed across two chapters, divides its focus. One part is dedicated to the modeling process, while the other focuses on the deployment aspects.

Chapter 3 presents methodology for modelling segmentation pipeline for three datasets of different magnification levels, analyzing their training and prediction performance, and validating the results against biologists and towards cell cycle nature.

Chapter 4 demonstrates a design for an automated cell culture monitoring system, developing its prototype using off-the-shelf hardware, and analyzing its feasibility with the proposed segmentation pipeline.

Chapter 5 shows a comprehensive analysis and findings derived from the methodologies discussed in Chapters 3 and 4.

Chapter 6 concludes the entire research work with the established objectives and extends recommendations for potential future work while acknowledging the study's limitations.

CHAPTER 2

LITERATURE REVIEW

2.1 GENERAL OVERVIEW

Biomedical image analysis is an interdisciplinary field that ranges from the macroscopic to the microscopic level, including human anatomy and cell culture. Medical personnel typically employ visual observation to manually analyze the medical images. However, human eyes are prone to optical illusions and inconsistency when dealing with various interpersonal, particularly images that contain fatigue errors and high disturbance. Thus, image segmentation has taken on a prominent role in the field of biomedical applications, ensuring the analysis favored by experts and assisting newly appointed personnel, thereby improving diagnostic confidence and analysis accuracy. Several methods for segmentation have been introduced using traditional computer vision and artificial intelligence approaches, including unsupervised learning and supervised learning. Despite the remarkable performance of the methods on specific test subjects, various techniques have been used to achieve state-of-the-art performance while keeping resource and experiment handling constraints in mind. The underlying review provides an overview of medical and biomedical segmentation, particularly cells and skin, and their existing approaches and refinements. It also delves into recent developments in cell segmentation tools and automated cell culture systems. Overall review highlights gaps, indicating better alternatives in the respective fields.

2.2 MEDICAL AND BIOMEDICAL SEGMENTATION

Medical segmentation and biomedical segmentation are both techniques used in the field of medical imaging to analyze and interpret images for various purposes. While there is some overlap in their goals, there are differences in their specific applications and contexts. Compared to medical segmentation, biomedical segmentation encompasses a wider range of applications beyond traditional medical imaging. It

involves segmenting not only anatomical structures but also details to cellular level. Biomedical segmentation techniques can be applied to diverse imaging modalities. In this particular literature review, the emphasis lies on cells and skin-related segmentation, specifically on humans, which would cover both medical and biomedical domains.

2.2.1 Cell Segmentation

Generally, cell segmentation is the process of segmenting microscopic image pixels into binary class where true and false classes represent cell regions and unoccupied regions, respectively. A well-segmented image can capture biologically relevant morphological information such as shape, health, contamination; however, fuzzy boundaries can cause segmentation leaks in which a segmented region expands into neighboring structures (Cao et al., 2018). The degree of difficulty to segment cells widely varies depending on, but not limited to, their cell morphologies and image modalities.

A. Cell Morphologies

Cell morphology is the study of the shape, appearance, and structure of cells. Understanding cell morphologies is an important criterion for designing the best procedure for detecting, segmenting, and classifying cells. In culture, Figure 2.1 shows that common cell morphology can be classified into three categories (Aida et al., 2020; Matsuzaka et al., 2021): (A) fibroblastic cells, (B) epithelial-like cells, and (C) lymphoblast-like cells. Lymphoblast cells are spherical such as blood cells in humans. Fibroblastic cells have elongated shapes with irregular dimensions while epithelial cells have polygonal outlines with more regular dimensions where both are commonly found within human tissues (Malik et al., 2023a).

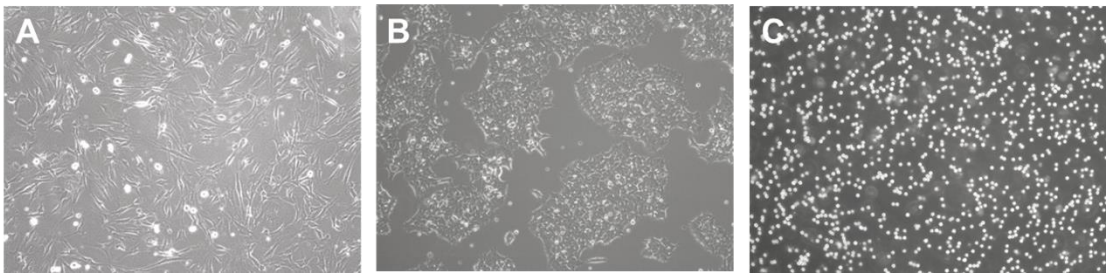


Figure 2.1 Types of Mammalian Cell Morphologies.

B. Microscopy Image Modalities

Cell imaging necessitates adjusting multiple lighting parameters, such as brightness, sharpness, contrast, and saturation, based on the microscope being used, as each microscope offers distinct image modalities. Certain cell characteristics may require specific microscope modalities in order to efficiently monitor the cells. For example, some cells may be better visualized using fluorescence imaging, while others may be better observed using brightfield imaging. However, not all cells require specific modalities, and it also depends on the study objectives and the cell type. Among the image modalities that can be seen in the literature review are illustrated in Figure 2.2 including phase contrast (Loewke et al., 2018), confocal (Eschweiler et al., 2019), fluorescence (Weigert et al., 2018), light sheet (Lo Vercio et al., 2022), bright-field (Din & Yu, 2021), and electron (K. Konishi et al., 2019) microscopy. A review by (Hollandi et al., 2022) has also shown the diversity of optical microscopy images representing nuclei as well as other cell structures.

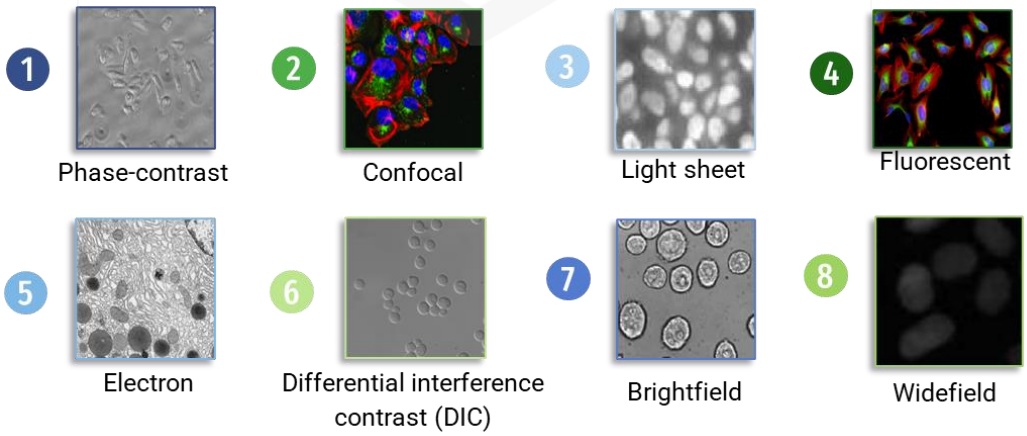
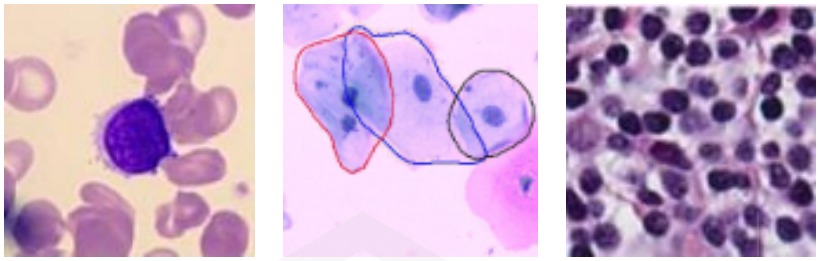


Figure 2.2 Types of Cell Image Modalities

Regardless of the image modalities, most research on mammalian cell segmentation has focused on cells that have less significant shape changes over time as portrayed in Figure 2.3, such as blood cells (Al-Dulaimi et al., 2021; Zhang et al., 2020), cervical cells (J. Huang et al., 2021; Jia et al., 2020), and nucleus (Kumar et al., 2020), and has been conducted at single magnification level. These cells tend to have closed shapes and regular dimensions.



(a) Lymphocyte cells (b) Cervical cells (c) Nucleus cells

Figure 2.3 Types of Cell Segmentation Using H&E Staining

2.2.2 Skin Related Segmentation

For skin-related segmentation, recent research has largely focused on medical images of the outer areas of the skin as depicted in Figure 2.4, such as skin pores (Kachi et al., 2020), chronic wounds (Gholami et al., 2018), and dominantly, lesions(Anjum et al., 2020; Tong et al., 2021), or the outer layer of the skin, such as epidermal tissue (Oskal et al., 2019) rather than the cellular structure of the skin.

(a) Pores (b) Chronic wound

(c) Lesions (d) Epidermal tissue

Figure 2.4 Types of Skin Related Segmentation

Based on the review, it is observed that the research on pore segmentation mostly focused on facial area as conducted by Campiche et al. (2019) and Li et al. (2023). Li et al. (2023) used standard 2 (S2) and cross-polarized (XP) modalities. XP image prevents surface reflection, giving accurate color informfation of the skin to measure color of skin and pores while S2 image captures both surface and subsurface reflection and is used for pore segmentation. Figure 2.5 is the detailed steps for the pore segmentation. However, this study did not focus on superior segmentation performance. It emphasizes the correlation between pore size segmented by the algorithm and visual grading. Therefore, the study showed that mean pore size highly correlated with visual perception by 0.86.

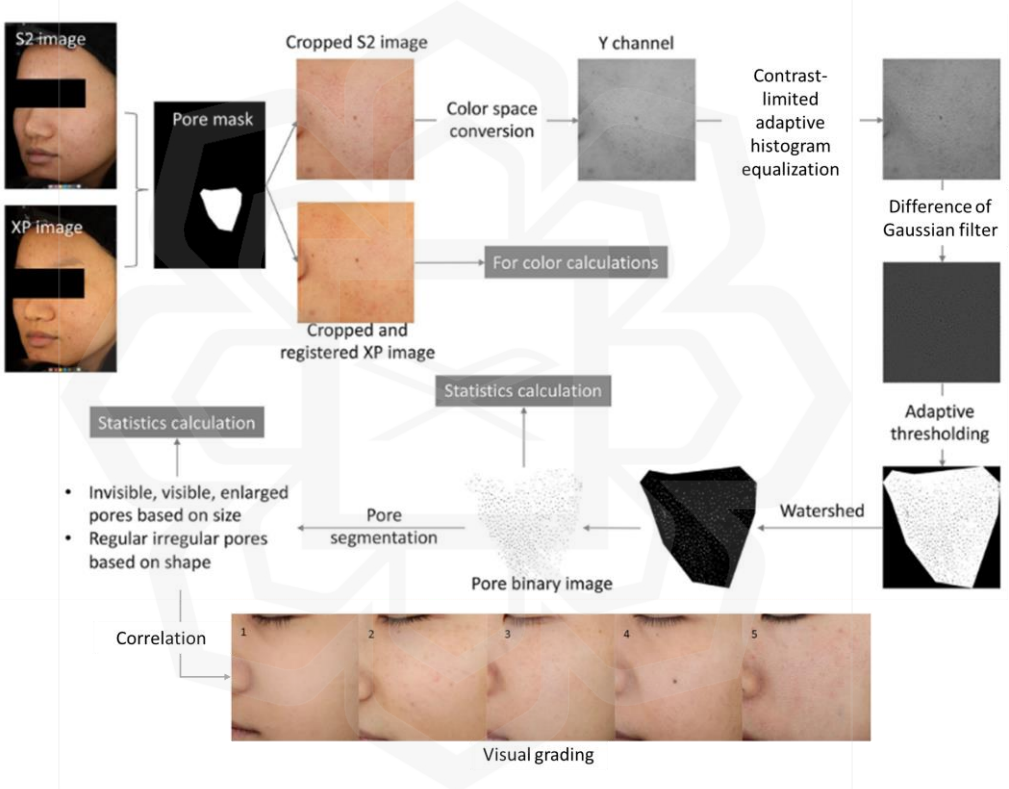


Figure 2.5 Pore Segmentation Schematic (Li et al., 2023)

Rather than applying traditional techniques, Kachi et al. (2020) chose to directly segment the pores using U-Net segmentation model where the images were first preprocessed pore images with bilateral filter. Instead of using moving average filters or Gaussian filters, bilateral filter with normal distribution was used to remove noise while preserving pore edge information using its variance parameter. The study presented higher dice coefficient by 6.11% on test data compared to the validation data.

Gholami et al. (2018) employed several segmentation methods for measuring skin wound geometry based on traditional computer vision, such as edge detection, morphological operations, region-growing, Livewire, active contours and Houhou-Thiran-Bresson (HTB) texture model. The study found that Livewire achieved the best performance in segmenting the skin chronic wound with 96.22 Jaccard Index. The final purpose of the study is to print a 3D wound geometry based on the segmented wound region as described in Figure 2.6.

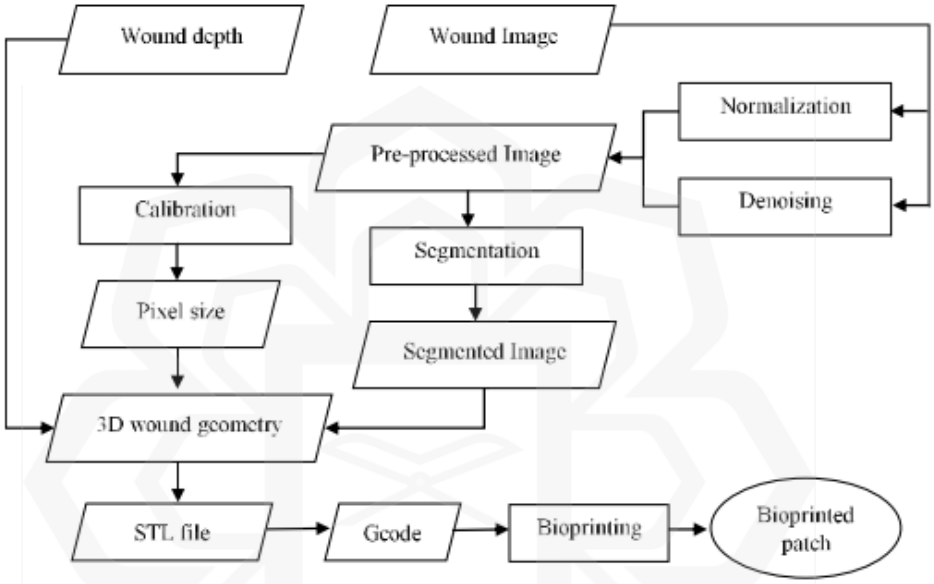


Figure 2.6 Procedure of Bioprinting 3D Wound Geometry (Gholami et al., 2018)

MobileNetV2 was adopted by C. Wang et al. (2020) to segment wound images. The study included connected component labelling as a post-processing to improve the true positive rate of segmentation results by filling holes within segmented wound regions. The study presented the highest performance of 90.47% compared by VGG16, SegNet, U-Net and Mask-RCNN, as measured by dice coefficient. To enhance future research, the authors proposed a novel multi-stream neural network architecture, visually depicted in Figure 2.7, designed to independently extracts shape features from pixel-wise convolution in their model.

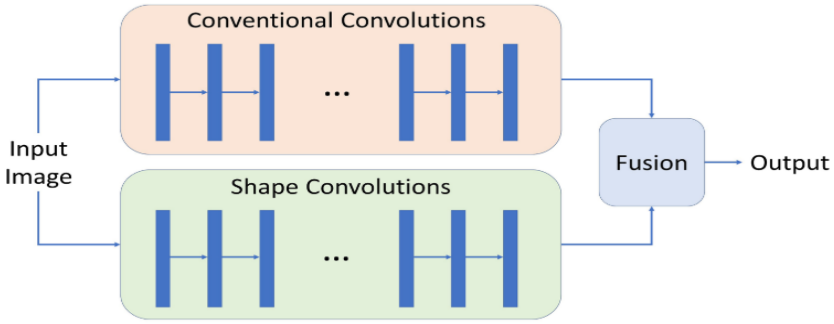


Figure 2.7 Multi-stream Model (C. Wang et al., 2020)

Anjum et al. (2020) proposed a deep learning approach for segmenting lesions where it includes localization, segmentation, and classification as demonstrated in Figure 2.8. YOLOV2-squeezeNet was first employed to localize the skin lesions and a 13-layer convolution neural network (CNN) with dilated convolution layers was used to segment the lesions with the detected area previously. Then, the lesion features were extracted and optimized using ResNet-18 and ant colony optimization (ACO), respectively. Based on the feature extraction, the lesions were classified based on optimized support vector machine (O-SVM) classifier and Naive Bayes (O-NB) classifier. Tong et al. (2021) modified the decoder blocks of the U-Net architecture by incorporating attention gates, spatial attention modules, and channel-wise attention modules for lesion segmentation. Their modifications resulted in a notable 4-8% improvement in the Jaccard Index.

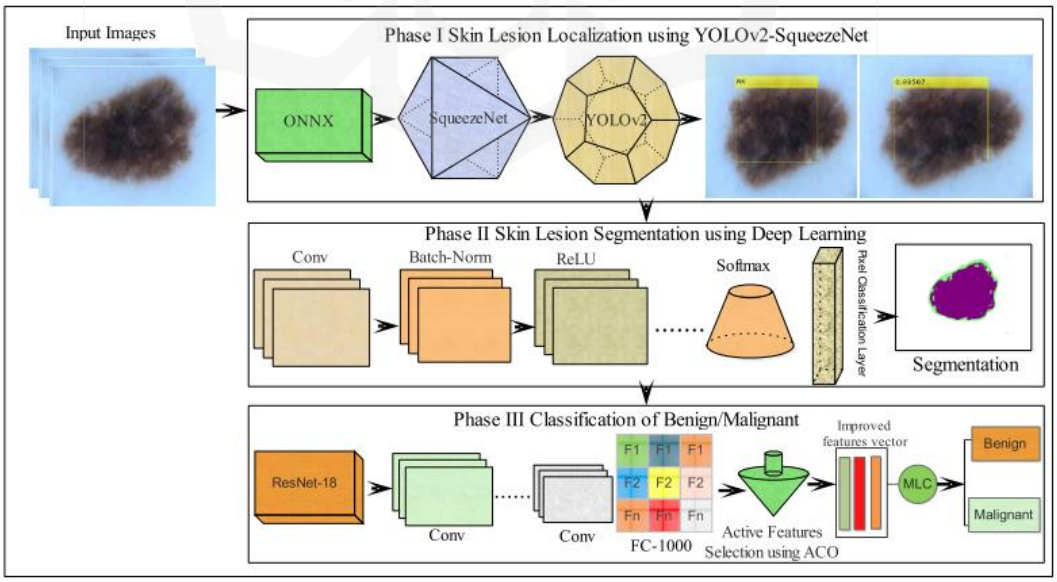


Figure 2.8 Skin Segmentation Model by Anjum et al. (2020)

Oskal et al. (2019) also utilized U-Net based network to segment epidermal tissue where the receptive fields of the model were increased to 5x5 filter size from 3x3 due to the high-resolution images. The study compared various traditional segmentation models, such as contrast enhancement and thresholding (CET), global thresholding and shape analysis (GTSA), and porosity analysis and strain concentration analysis (PASC). The proposed model demonstrated superior performance across multiple evaluation metrics, including Positive Predictive Value, Sensitivity, Dice Similarity Coefficient, and Matthews Correlation Coefficient.

In brief, the scope of previous studies, particularly for skin segmentation, is heavily limited to the visible features rather than the underlying cells and to cells with more consistent, predictable, and closed shapes, rather than cells that exhibit more complex and varied shape changes over time. The previous studies solely differ in that they introduce new models or approaches to address similar object segmentation on various image modalities from open-source databases such as Kaggle.

2.2.3 Multi-scale Segmentation

According to the literature review, the majority of biomedical and medical research on multi-scale segmentation has primarily focused on enhancing the model architecture rather than exploring its applications. While researchers claim to have developed multi-scale architectures, they have mainly tested them on single-scale object images. Below is the literature that involves the use of multi-scale subjects as depicted in Figure 2.9.

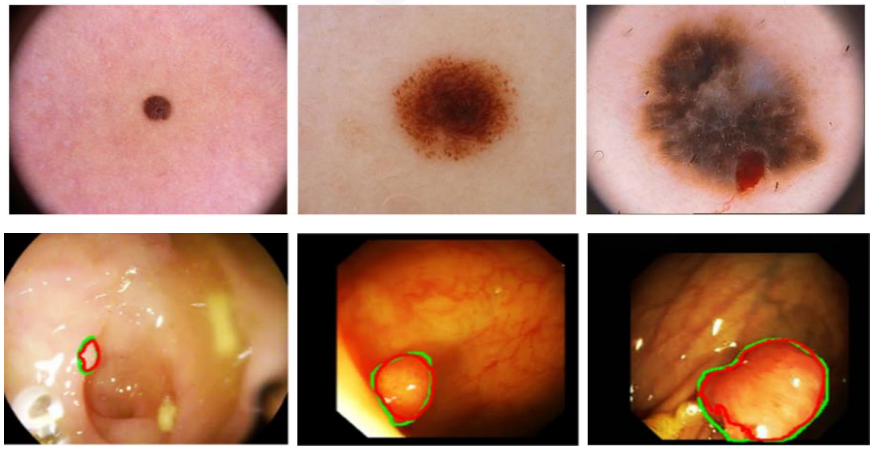
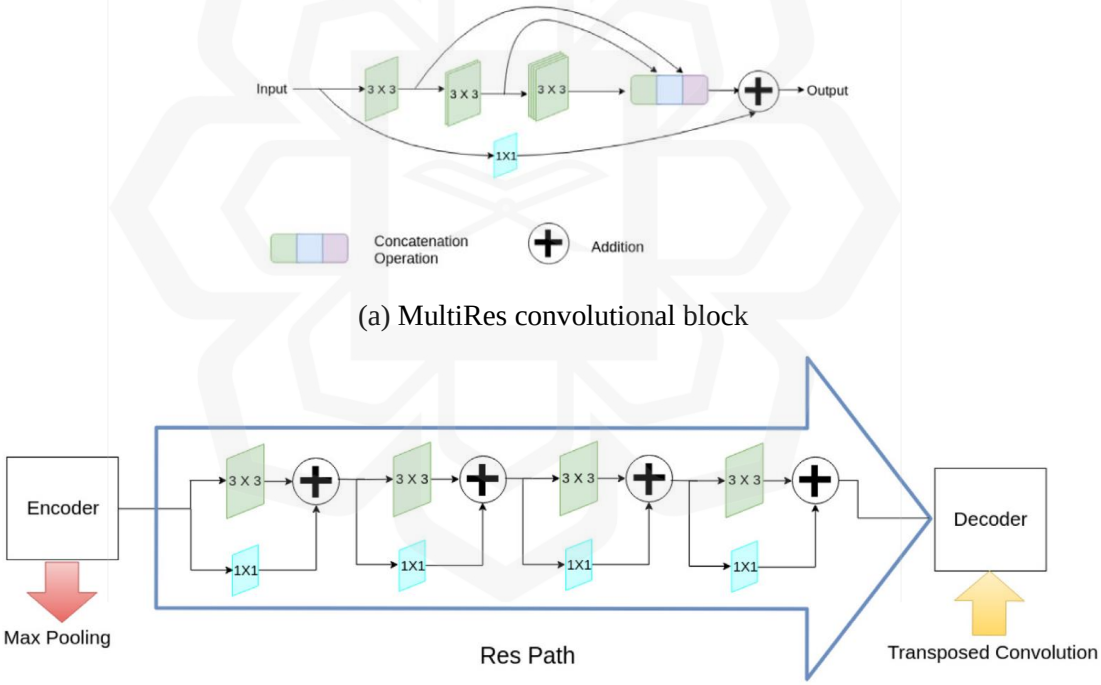


Figure 2.9 Example Images of Multi-scale Subject

Ibtehaz & Rahman (2020) rethought UNet architecture for multi-scale biomedical image segmentation, introducing MultiResUNet. Lesion images are one of the datasets tested, as Figure 2.9 shows them on the upper row in multiple sizes. The researchers increased the number of 3×3 filters of each convolutional block in the successive three layers. The feature maps of each successive filter are concatenated and added with the feature maps of a residual connection from the input along with 1×1 filters. They also replaced the original skip connection of UNet with a sequence of four successive 3×3 filters, each accompanied by a short residual connection consisting of 1×1 filters. However, the researchers opted for a lower number of filters in the added convolutional layers compared to the original UNet. As a result, the MultiResUNet model featured only 7,262,750 parameters. The modifications are illustrated in Figure 2.10.



(a) MultiRes convolutional block

(b) Residual path

Figure 2.10 MultiResUNet Modules (Ibtehaz & Rahman, 2020)

Iqbal et al. (2022) proposed a UNet architecture with modified receptive field with feature-fused module and attention gate mechanism as depicted in Figure 2.11. The feature-fused module replaced traditional U-Net convolutions with two branches of convolution operation of 3x3 kernel filter and one with dilation size 2, where both

operations are fused using element-wise addition. The model resulted in 3.94 million parameters which is 49% less than the classical UNet. The model was experimented on datasets of skin lesion, colorectal polyps, and nuclei images. The study found that FF-UNet model without attention gate performs better on most of the open datasets.

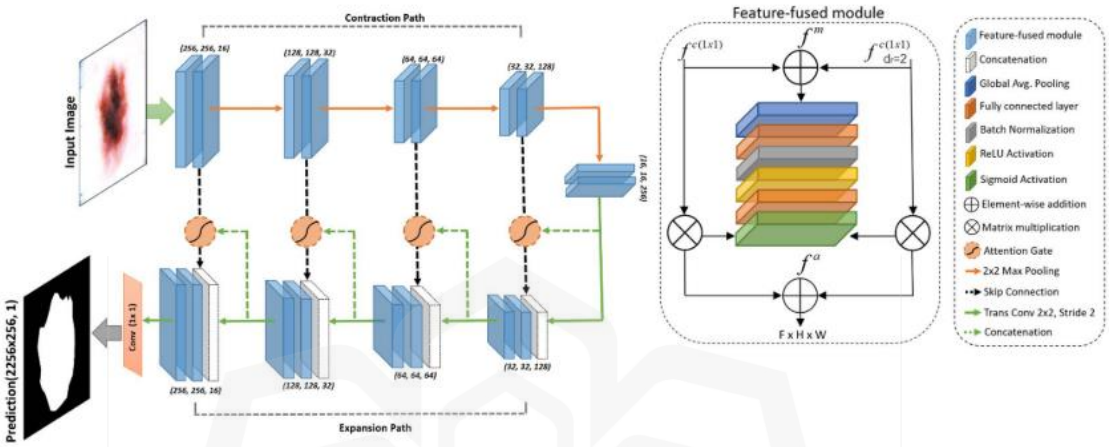


Figure 2.11 FF-UNet Architecture (Iqbal et al., 2022)

In contrast to the previously mentioned studies, the literature referenced below utilized multi-scale images based on different image resolutions. While it is worth discussing these findings, it should be noted that the multi-scale images in this study exhibited a lack of feature variations in their internal structure. However, the conceptual framework employed in this research holds significant potential, even if the implementation falls short in terms of capturing diverse features at various scales. A. Khan et al. (2021) proposed PMED-Net that consists of six small encoder-decoder networks (six level) where each network generates coarse predictions and they are upsampled with stride 2, concatenated with different scale input image, as demonstrated in Figure 2.12, and used as input for next level network. Each instance of the proposed network has only three stages with a much fewer number of feature maps as 16, 32, and 64. Overall, the proposed architecture comprises 1,465,974 parameters for its six-pyramid level training.

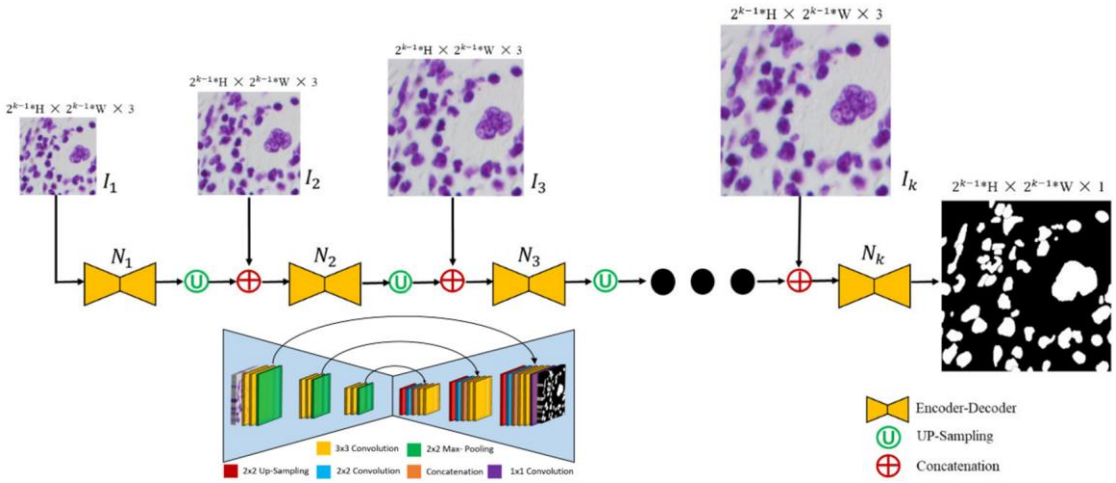


Figure 2.12 PMED-Net Architecture (A. Khan et al., 2021)

Wu et al. (2021) and Nofallah et al. (2022) introduced a scale-aware transformers that allows the system to learn local and global representations from digital whole slide histopathological images of skin tissue images at multiple scales in an end-to-end fashion. Based on Figure 2.13, the proposed model has three main steps, including learning patch embeddings using CNN for each input scale, learning contextualized patch embeddings for each input scale using transformers, and learning scale-aware embeddings across multiple input scales using transformers.

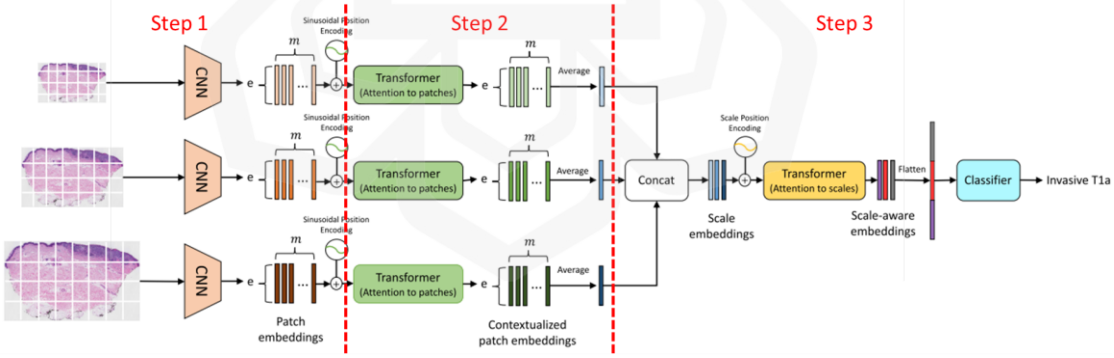


Figure 2.13 ScATNet End-to-end Framework (Wu et al., 2021)

2.2.4 Summary of Respective Medical and Biomedical Segmentation

This subsection is to summarize the significant information within section 2.2, which are also organized in Table 2.1, detailing the methods employed and the applications they have been tested on. Despite numerous models having been introduced and

applied in these domains, the review found that the dominance of U-Net based networks has become increasingly evident, including on multi-modal images. Therefore, this study firmly believes that focusing on the implementation of U-Net and its variants is not only relevant but also crucial for advancing the field.

Table 2.1 Overview of Methods and Applications of Reviewed Models

Proposed Models	**Segmentation Methods**	**Applied Datasets**
Cell Segmentation		
Zhang et al. (2020)	Deformable U-Net	Blood cells
Jia et al. (2020)	Computer Vision (Black Otsu + Gradient Vector Flow Snake Model)	Cervical cells
J. Huang et al. (2021)	Generative Adversarial Networks (GAN)	
Skin-related Segmentation		
Li et al. (2023)	Adaptive thresholding	Pore
Kachi et al. (2020)	U-Net	
Gholami et al. (2018)	Computer Vision (Livewire)	Wound
C. Wang et al. (2020)	MobileNetV2	
Anjum et al. (2020)	13-layers CNN	Lesion
Tong et al. (2021)	Attention, spatial and channel attention U-Net	
Oskal et al. (2019)	5x5 filter size U-Net	H&E skin tissue
Multi-scale Segmentation		
(a) Different subject scales		
MultiResUNet by Ibtehaz & Rahman (2020)	Modified U-Net • Residual block and path	• Nuclei microscopy • Nerve cord microscopy • Colon polyp • Skin lesion • MRI brain
FF-UNet by Iqbal et al. (2022)	Modified U-Net • Feature-fused module (dilated inception) • Attention module	• Colon polyp • Nuclei microscopy • Skin lesion
(b) Different resolution scales		
PMED-Net by A. Khan et al. (2021)	Cascaded U-Net	• Skin lesion • Brain tumor • Xray chest • Nuclei
ScATNet by Wu et al. (2021)	• MobileNetV2 • Transformers	H&E skin tissue

2.3 SEGMENTATION MODELS AND REFINEMENTS

Multiple segmentation models have been introduced to segment biomedical images, particularly cells, from traditional computer vision to deep learning. There are numerous computer vision techniques capable of segmenting cells in images, such as thresholding, watershed, edge detector, and region-based method; however, these techniques have been shown to be ineffective and less accurate in overcoming challenges, especially those involving cell imaging analysis, such as high density, low contrast, deformable cell shapes, appearance variation, and cell occlusion (H. Hu et al., 2019). Moreover, these techniques often require experiment-specific parameter tuning.

Compared to traditional techniques, DL achieves superior accuracy in visual analysis tasks such as for object detection and semantic segmentation as it is trained using existing data rather than programmed. It has the advantage of requiring less expert analysis and experiment-specific parameter tuning to respond flexibly to varied images with relatively close features such as the object shapes (O'Mahony et al., 2019). Therefore, recent research on cell segmentation mostly implements deep learning such as Convolutional Neural Network (CNN). Convolutional neural network (CNN or DCNN), a supervised learning-based model, has been practically employed for image analysis and has been extended to various fields such as robotics, medical imaging, smart cities, and self-driving cars.

Based on previously mentioned literature review on biomedical and medical segmentation, despite there are multiple CNN architectures have been developed for various object segmentations, U-Net, introduced by Ronneberger et al. (2015), has been proved to be widely used and claimed to be effective in the respective domains. The key advantage of U-Net is that it can learn global and local context simultaneously on high resolution data due to its skip connections (Punn & Agarwal, 2022). Moreover, it also can generate highly detailed segmentation maps even with limited or less data training (Siddique et al., 2021). The standard architecture of U-Net is shown in Figure 2.14.

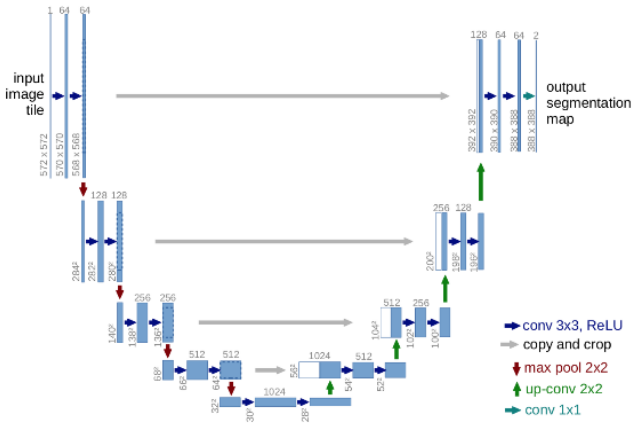


Figure 2.14 Standard U-Net Architecture (Ronneberger et al., 2015)

Even for the same objects to segment, they also have different structure conditions and image settings. Model refinement can be achieved through changes in both the architecture of the model and the properties of the dataset used for training. By making these adjustments, it is possible to improve the performance of the model while also reducing its complexity.

2.3.1 Architecture Refinement

This section will focus on the refinement of deep learning architectures since they are dominant in recent research. Multiple mechanisms (M. Z. Khan et al., 2021) have been suggested to improve performance deep learning architectures to reduce error of various types of structure including U-Net. One of the mechanisms is called an ensemble or cascade where two or more of the same networks are simply used to develop a complete model. An example was shown by Lugagne et al. (2020) where two U-Net was applied for segmentation and tracking tasks for parent and daughter cells. (Chen et al., 2019) used two consecutive U-Nets for esophagus and esophageal cancer images. A model of three parallel DenseNet was developed by S. Wang et al. (2022) to segment the object and the output of all networks were combined by training fusion network for classification task.

Model enhancement can also be done through attention mechanism. An attention gate, as shown in Figure 2.15, can automatically learns to focus on varying characteristics of target structures by suppressing irrelevant regions and highlighting salient features (Schlemper et al., 2019). Oktay et al. (2018) proposed Attention U-Net

by combining the standard U-Net skip connections with attention gate to increase the model sensitivity to target foreground pixels without complicating the skip connections.

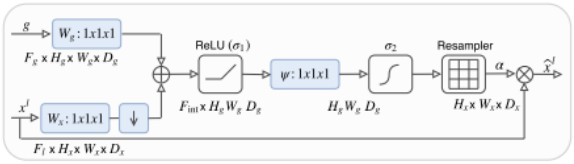


Figure 2.15 Attention Mechanism (Schlemper et al., 2019)

ResNet was proposed by He et al. (2015), which introduced a residual mechanism. Residual employs technique known as skip connections, as shown in Figure 2.16, to solve vanishing or exploding gradient problems by skipping layers that perform poorly and concatenating feature maps from the performing layer directly to the output. C. Huang et al. (2020) applied residual paths between input and output of each convolutional blocks to help information spread and accelerate network convergence.

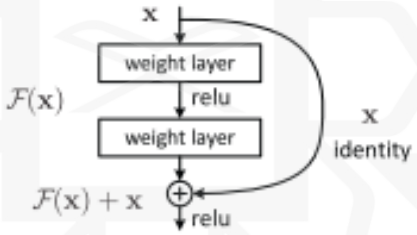


Figure 2.16 Residual Mechanism (He et al., 2016)

A dense mechanism was inspired by DenseNet block introduced by Huang & Weinberger (2018) where feature maps of all preceding layers are concatenated with feature maps of succeeding layers as an input for the subsequent layers as illustrated in Figure 2.17. An example can be seen on U-Net++ proposed by Zhou et al. (2018) by redesigning skip connections with nested dense convolution blocks to reduce the semantic gap between feature maps of the contraction and expansion path by encouraging more frequent feature reuse and strengthening feature propagation across the architecture.

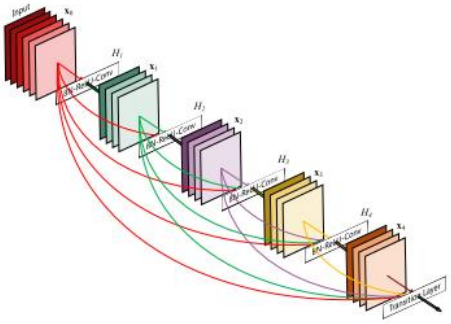


Figure 2.17 Dense Mechanism (G. Huang & Weinberger, 2017)

GoogleNet is a contemporary networks that introduced an inception mechanism (Szegedy et al., 2015), as displayed in Figure 2.18. H. Hu et al. (2019) and Su et al. (2021) used the mechanism and proposed multi-scale convolution U-Net namely MC-UNet and MSU-Net respectively by redesigning the convolution layers with different kernel sizes concurrently to extract more semantic features and more diverse features at different scales. The fixed 3x3 convolutional layers of U-Net were replaced with multiple concurrent kernel sizes and feature maps were concatenated for the following layers.

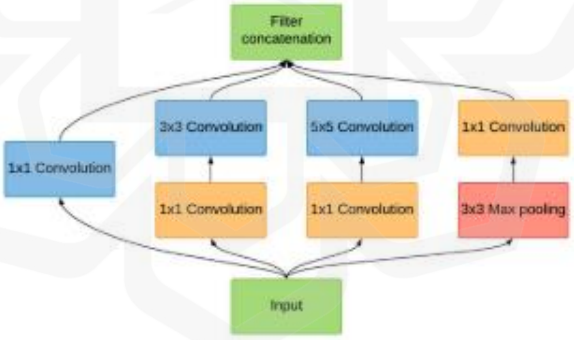


Figure 2.18 Inception Mechanism (Siddique et al., 2021)

Dilated mechanism, also known as atrous convolution, expands the kernel by defining spacing between the values based on dilation rate as it covers larger area of the input as depicted in Figure 2.19. Gu et al. (2019) included atrous convolution in their architecture to compensate for the loss of image semantic information caused by pooling layers. Deformable convolution was introduced by Dai et al. (2017) where it adds free 2D offsets, which, unlike regular dilated convolution, to the regular square grid sampling locations learned from the preceding feature maps, and Zhang et al.

(2020) and Lei et al. (2022) applied it to enhance the network capability of dealing with object shape transformations. The mechanism comparisons are illustrated graphically in Figure 2.19.

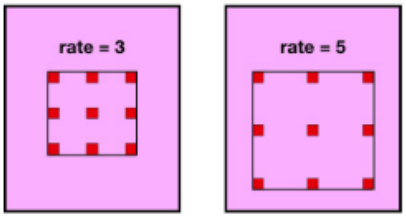


Figure 2.19 Dilated Kernel Mechanism (Gu et al., 2019)

Concisely, despite numerous attempts to refine machine learning models, many of these approaches prioritize achieving higher performance without considering the trade-off towards model complexity. In other words, some methods may sacrifice simplicity and interpretability in favor of improved accuracy or predictive power. The issue of model complexity is particularly crucial when considering the deployment of the refined models on edge devices for real-world applications.

2.3.2 Dataset and Segmentation Refinement

Refining a segmentation model not solely involves adjusting the model architecture, but also optimizing the pre-processing and post-processing stages to improve the accuracy and performance of the model as well as adapting the dataset conditions.

A. Dataset Pre-Processing

Image pre-processing aims to modify images so that they are best suited for training and inference by eliminating or reducing the noise of images, enhancing information, and simplifying data. Common examples of the process include, but not limited to, color conversion, color correction and normalization. Commonly, image pre-processing was decided manually based on individual experience which transformations should be applied to particular image dataset where it is dependent on intuition and trial-and-error approaches. Therefore, to mitigate the inefficiency,

automated data pre-processing using deep reinforcement learning was introduced by (Minh et al., 2018). As shown in Figure 2.20, training on polar image transformations has been proposed by Benčević et al. (2021) to improve biomedical segmentation where it involves centerpoint prediction to identify polar origin. However, it only and most effectively applies on segmenting a single elliptically distributed object.

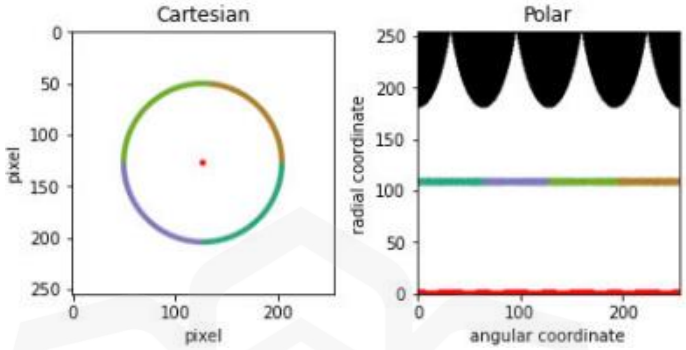


Figure 2.20 Example of Polar Transformation (Benčević et al., 2021)

There are some training strategies that have been proposed by researchers to improve the model performance and reduce computational power. A transfer learning from patch-based learning to image-based training was proposed by Sekou et al. (2019), as demonstrated in Figure 2.21, to overcome the problem of having small number of medical samples. The original image is patched to any numbers as long as each patch can carry significant information of the object of interest. Otherwise, the transfer learning of patch-based training may fail.

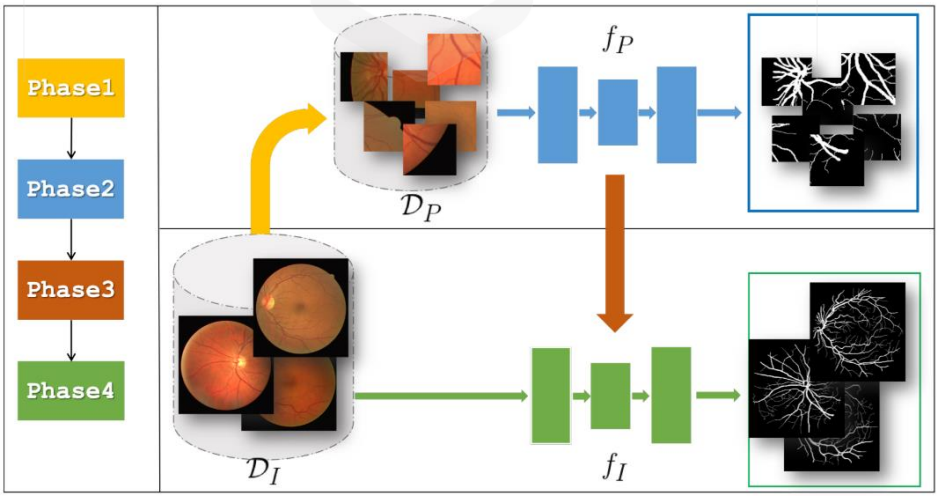


Figure 2.21 From Patch to Image Segmentation (Sekou et al., 2019)

B. Segmentation Post-Processing

Despite the lack of traditional computer vision techniques, they can be integrated with machine learning approaches to address some potential issues of segmentation mask, such as eliminating the possibility that if some of these segmented objects are very close to each other, the objects will touch each other, not reflecting the true case, especially when involved with object measurement. can be improved by these techniques. For instance, segmentation based on classical watershed algorithm alone is troublesome for the pseudo edge of medical images mentioned by (Shen & Wang, 2018) while (Ng et al., 2006) proved that the hybrid of machine learning with watershed algorithm for post-processing which is able to split excessive segmentation and is more representative of segmented objects.

2.3.3 Summary of Segmentation Models and Refinements

This section emphasizes that it is essential to balance how well a model works with how complicated it is. It advises against mainly focusing on being very accurate if it makes the model too complex to implement on different devices. While architecture refinements advocate balancing between accuracy and complexity, enhancements in dataset handling and segmentation techniques highlight the significance of refining training efficiency and achieving better segmentation outcomes through suitable pre- and post-processing methods. Nevertheless, the choice of a refinement strategy should be influenced by the specific dataset and the objectives the study aims to achieve.

2.4 USER-FRIENDLY CELL SEGMENTATION TOOLS

To simplify and accelerate cell culture analysis and provide consistent and accurate measurement for user convenience, biologists and computer scientists have worked together for years to develop a number of tools or software with a user-friendly graphical interface for cell detection and segmentation on various cell morphologies and image modalities. In addition, the selection of the best image analysis tools depends on the specimen, object of interest, and imaging method (Belevich et al., 2016).

These tools can also be used either locally, on the cloud, or both. Compared to local installation, cloud-based tools have the advantage of eliminating configuration issues and hardware requirements that are commonly encountered in deep learning applications (Malik et al., 2023a). This makes them more accessible to end users who may not have access to high-performance computing devices, as long as they have an internet connection. Recent tools for cell segmentation developed in 2022 are listed in Table 2.2.

Table 2.2 Recent Segmentation Tools in 2022

Local installation	**Cloud-based**
DetecDiv (Aspert et al., 2022)	DeLTA (O'Connor et al., 2022)
microbeSEG (Scherr et al., 2022)	DeepLIIF (Ghahremani et al., 2022)
Cellpose 2.0 (Pachitariu & Stringer, 2022)	

To be outline, an issue commonly found in literature is the lack of comprehensive information on the hardware utilized in their evaluations, as noted in the works of T. Hu et al. (2021) and Zargari et al. (2021). This omission makes it challenging for readers to determine the true computational demands of the tools, particularly those utilizing deep learning approaches that can be resource intensive. However, some papers, such as Gallois and Candelier (2021) and Aspert et al. (2022), have provided a detailed description of their workstation hardware. This is crucial because it is what makes the tools more extensible to various end users.

2.5 SEGMENTATION MODELS FOR MOBILE DEPLOYMENT

The previous section discusses user-friendly integration of segmentation models and personal devices such as laptops or personal computers. On the other hand, this section emphasizes a review of existing lightweight models focusing on medical and biomedical segmentation and were claimed to be suitable for mobile and edge devices with low computational power.

Ramachandram et al. (2022) proposed two models for wound and tissue segmentations called as AutoTrace and AutoTissue as shown in Figure 2.22. AutoTrace utilizes an attention-based depthwise encoder-decoder architecture inspired

by U-Net, with notable enhancements across three key steps. Firstly, the conventional convolutional blocks have been substituted with depthwise separable convolutional layers. Second, the architecture incorporates strided depthwise convolutions, which dynamically learn to downsample activations instead of relying on a fixed max-pooling operation for downsampling. Thirdly, an additive attention gate was placed in each of the skip connections in the architecture. AutoTissue comprises EfficientNet-B0 encoder and AutoTrace decoder. While AutoTrace only focuses on wound identification, AutoTissue was tested on four different tissue proportions: epithelial, granulation, slough, and eschar. AutoTrace demonstrated acceptable performance for most tissue types. However, it encountered challenges specifically with epithelial tissue, exhibiting a sensitivity below 0.5 for this tissue category.

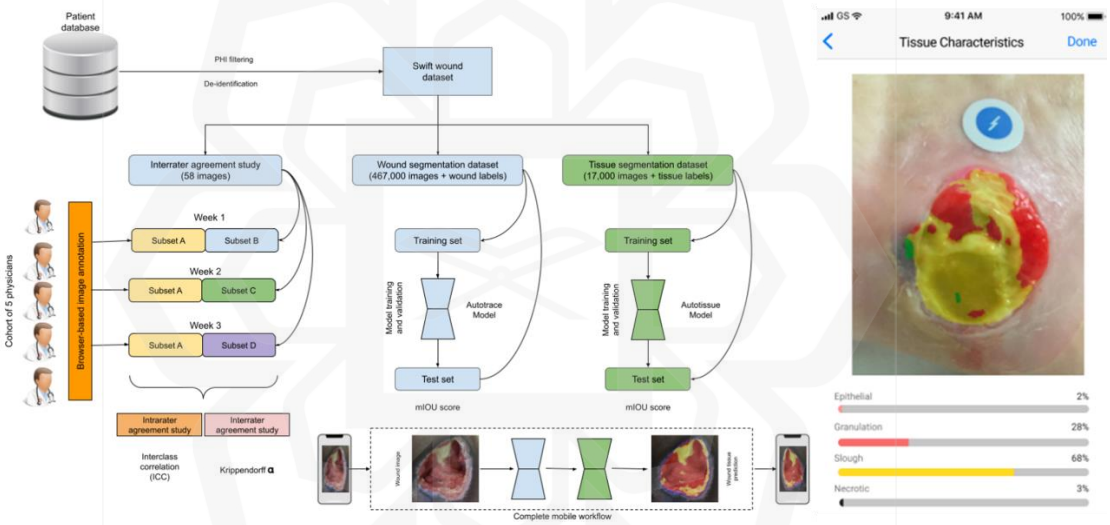


Figure 2.22 Mobile AutoTrace and AutoTissue (Ramachandram et al., 2022)

Gadosey et al. (2020) introduced a Stripped-Down UNet (SD-UNet) by simply replacing standard convolution layer of U-Net with depthwise separable layer. Various combinations of regularization techniques were analyzed on the model, including group normalization, batch normalization, and weight standardization, to resolve the performance degradation due to the depthwise separable convolutions. The final proposed model comprises 3.90 million parameters structured as shown in Figure 2.23. Compared to the standard U-Net, the model has 3.05% higher dice coefficient when experimented on electron microscopy of nerve dataset and brain tumor dataset.

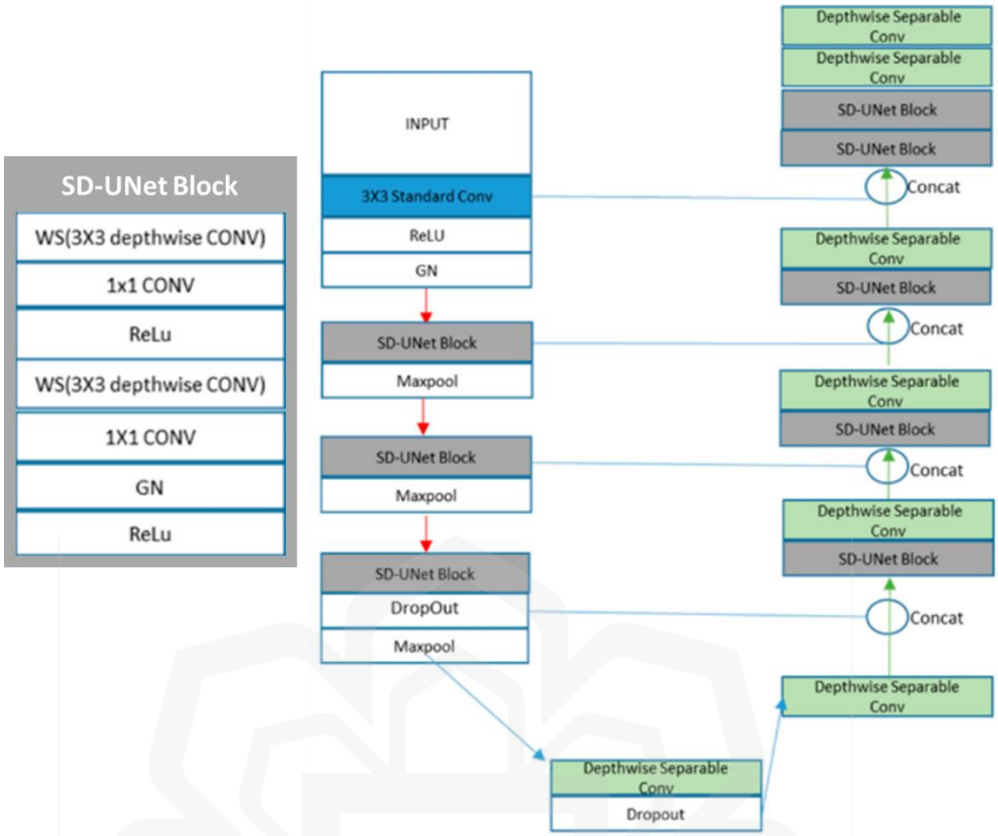


Figure 2.23 SD-UNet Architecture (Gadosey et al., 2020)

Kist and Dollinger (2020) suggested tiny-weight U-Net for EdgeTPU deployment considering the reduction of U-Net layer and filter number, the removal of second convolution layer, the replacement of depth-wise separable convolution and the addition of residual propagation. The study found that the second convolutional layer in encoder and decoder is not required in the U-Net architecture for high accuracy segmentations and only residual propagation is relevant, not its type. They were also able to reduce the parameters by over 99.8% while keeping the 95% of the standard U-Net accuracy as measured by the Intersection over Union (IoU) metrics. Four final architectures of tiny U-Net were proposed as presented in Table 2.3. The architectures are abbreviated as L (number of layers), F (base filter number), EE/ Ex ((Encoder with and without second convolution), DD/ Dx (Decoder with and without second convolution), C/ SC (Convolution and separable convolution), and Concat/ Add (Residual propagation type).

Table 2.3 Models Proposed for EdgeTPU Deployment

Architecture	Parameters (million)	IoU (test)
4L, 16F, EE, DD, C, Concat	1.3	0.769
4L, 8F, EE, DD, C, Add	0.3	0.776
4L, 8F, Ex, Dx, C, Add	0.09	0.746
4L, 8F, Ex, Dx, SC, Add	0.03	0.741

Vaze et al. (2020) employed a thin separable convolution U-Net adapted with knowledge distillation for ultrasound segmentation, using the thin regular U-Net as a teacher model, as portrayed in Figure 2.24. Again, the teacher model was introduced due to the use of the depthwise separable convolution causing the performance consistently worse across the depth parameter space. However, similar as the previous study, this study mainly contributed to reduced complexity of U-Net architecture to only 1.3 million parameters as no significant improvement of model segmentation on the ultrasound images.

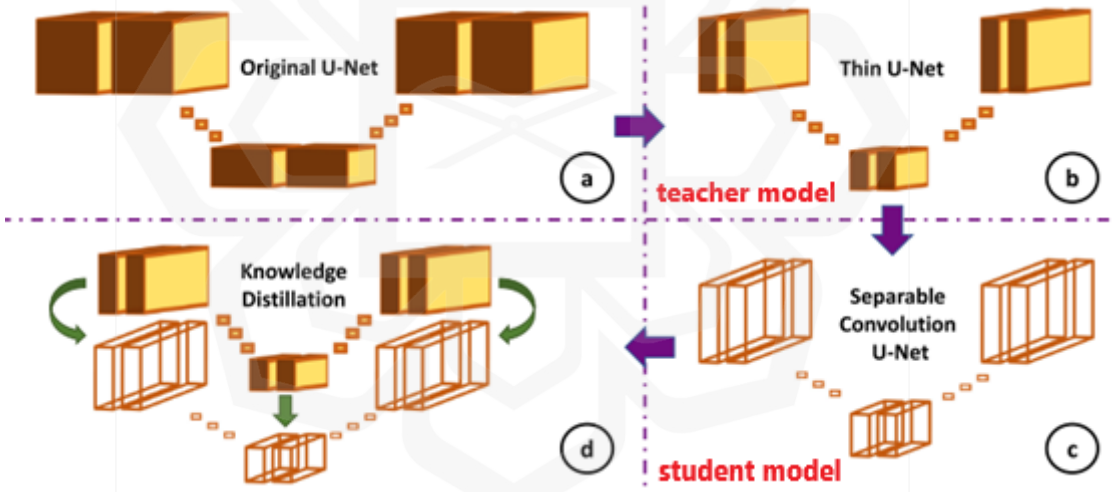


Figure 2.24 Distilled Thin Separable U-Net Development (Vaze et al., 2020)

In summary, a common approach that is used for mobile deployments is through depth-wise separable convolutions. Most studies also pointed out that the implementation of the depth-wise separable convolution at the first convolution block should be avoided as it can severely limit initial representational power. Moreover, regardless of its placement, it was also acknowledged that the approach can lead to performance degradation and have proposed various refinements to address this issue. The variations in these refinements distinguish the proposed models by different

researchers. Another similar observation as the previous section, the existing literature often fails to provide concrete evidence of the effectiveness of the proposed models in mobile deployments as claimed except from Vaze et al. (2020) and Kist and Dollinger (2020). While various models have been introduced, few studies have specifically validated their performance and complexity on specified mobile devices. The summary of the complexity characteristics of the mobile models is tabulated in Table 2.4.

Table 2.4 Existing Models for Mobile Deployments

Methods	**Year**	**Parameters (million)**	**Inf. time (ms)**	**Main device**	**Mobile device**
SD-UNet	2020	3.90	56	NVidia Tesla K40C GPU (12 VRAM)	-
Tiny-weight U-Net	2020	1.3	195	Titan RTX (24 VRAM)	Google Coral USB EdgeTPU
Distilled Thin Separable U-Net	2020	55.6×10^{-3}	124	NVIDIA GeForce GTX 1080Ti (11 VRAM)	Google Pixel 2
			388		Samsung Galaxy Tab A
AutoTrace (Attention Separable U-Net) & AutoTissue (EfficientNetB0)	2022	< 16	300	-	Mobile phone

2.6 REMOTE MONITORING AND CONTROL FOR CELL CULTURE

The advent of remote monitoring and control has brought about transformative changes across various industries, and the realm of cell culture is no exception. In light of technological advancements and an escalating need for precision and efficiency, remote monitoring has emerged as a pivotal solution for cell culture practices. This cutting-edge approach empowers researchers, scientists, and professionals to remotely oversee and analyze cell cultures in real-time, eliminating the requirement for physical presence in the laboratory. This section mainly discusses

the components of the communication line for monitoring the cells and the environment (incubator).

Pérez et al. (2018) developed a system for remote and non-invasive measurement of living cell assays using an oscillation-based test (OBT) sensor. As visualized in Figure 2.25, the system uses an electrical cell-substrate impedance spectroscopy (ECIS) sensor and an ARM Cortex-M7 microcontroller, with an Intel Edison device as the gateway which communicates with the local sensors using Bluetooth and transfer the data to MySQL database via Internet connection. The sensor devices are custom electronic prototypes that are also capable of measuring humidity, temperature, battery, and the analog sensor electrical properties. The sensor device is implemented using a printed circuit board (PCB) prototype with digital to analog (DAC) and analog to digital (ADC) converters to trigger and acquire the OBT sensor signal. The oscillation parameters, such as amplitude and frequencies of the main harmonics, are directly related to the biological variables under study, such as cell number, cell growth rate, and cell size, linking the biological properties to the electrical ones.

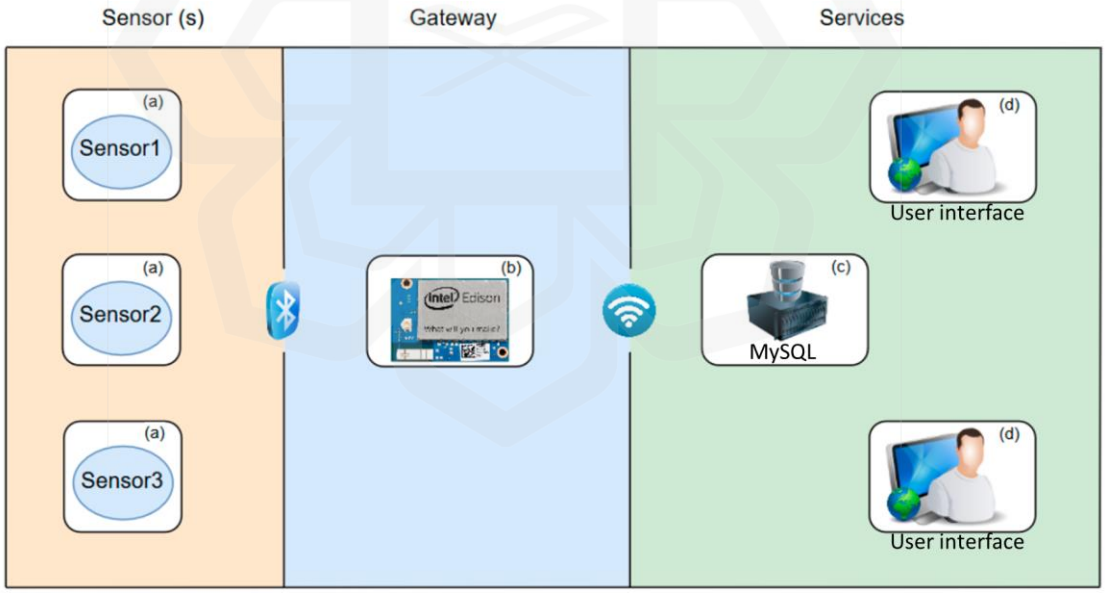


Figure 2.25 Remote System Architecture by Pérez et al. (2018)

Baudin et al. (2022) introduced Picroscope, a low-cost remote imaging platform for biological applications that allows for longitudinal imaging studies on multi-well cell culture plates, as shown in Figure 2.26. The system is comprised of a cluster of network-connected devices where the top-level control is handled by a

Raspberry Pi-based hub which communicates with 24 Raspberry Pi Zero Ws each of which control one camera for one culture plate, and also connects to an Arduino Uno to control motors and lights, report temperature and humidity, and prevent overheating event. As demonstrated in Figure 2.27, Picroscope uses a cloud based MQTT broker provided by Amazon IoT for authorized users to control experiment parameters and start or stop commands. A complete result of images is uploaded to an s3 object store on cloud hardware run on the Pacific Research Platform (PRP) and will be accessible over Internet through Public Access Website.

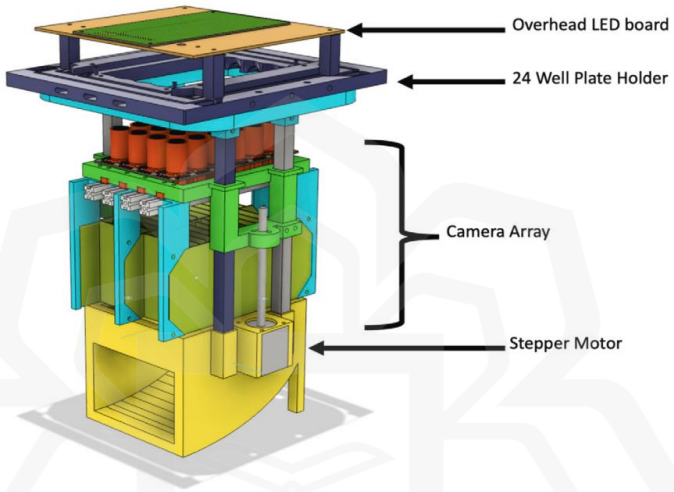


Figure 2.26 Picroscope (Baudin et al., 2022)

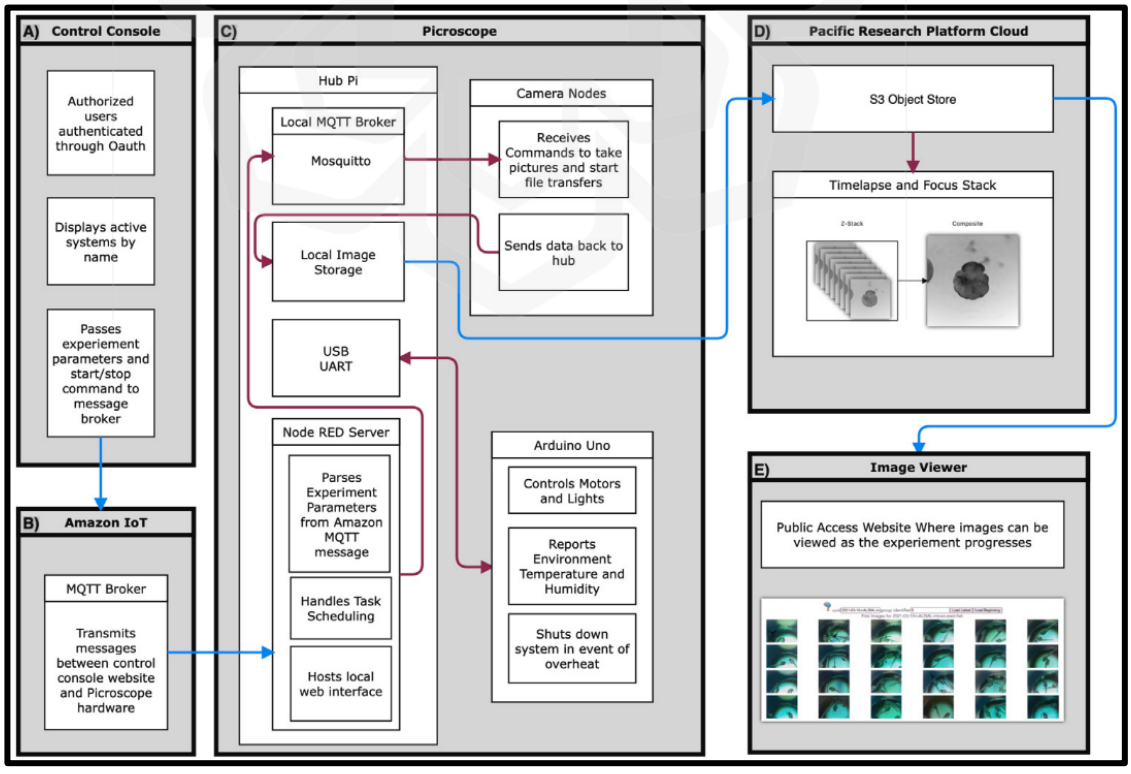


Figure 2.27 Remote System Architecture by Picroscope (Baudin et al., 2022)

Piphys is a cloud-based electrophysiology platform for recording neural activity in real-time developed by Voitiuk et al. (2021). An expansion shield was used to enable an interface between Raspberry Pi Model 3B+ and Intan RHD2132 bio amplifier chip to perform electrophysiology. Communication between the Raspberry Pi and Intan chip is connected using SPI as shown in Figure 2.28. A commercialized 6-well MEAR electrode plate was connected to the Intan RHD 32-channel recording headstage. Same as the previous study, the users can control the experiment parameters through MQTT broker provided by Amazon IoT. The results of the digitized neural signal are pushed to Redis for near real-time visualization on the online dashboard. After the recording is complete, all local data files are transferred to S3 to ensure permanent storage, while an additional backup is made to Amazon Glacier for long-term archiving purposes. Figure 2.29 depicts the detailed communication of Piphys architecture.

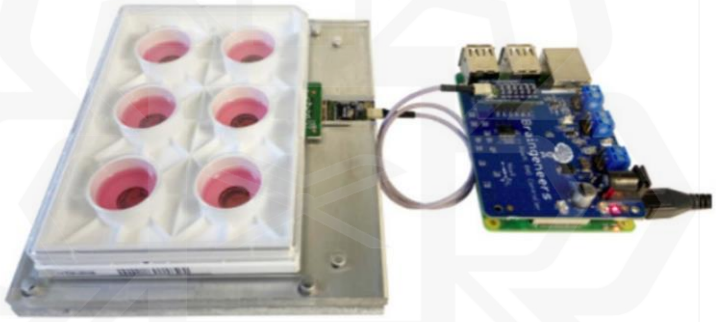


Figure 2.28 Piphys (Voitiuk et al., 2021)

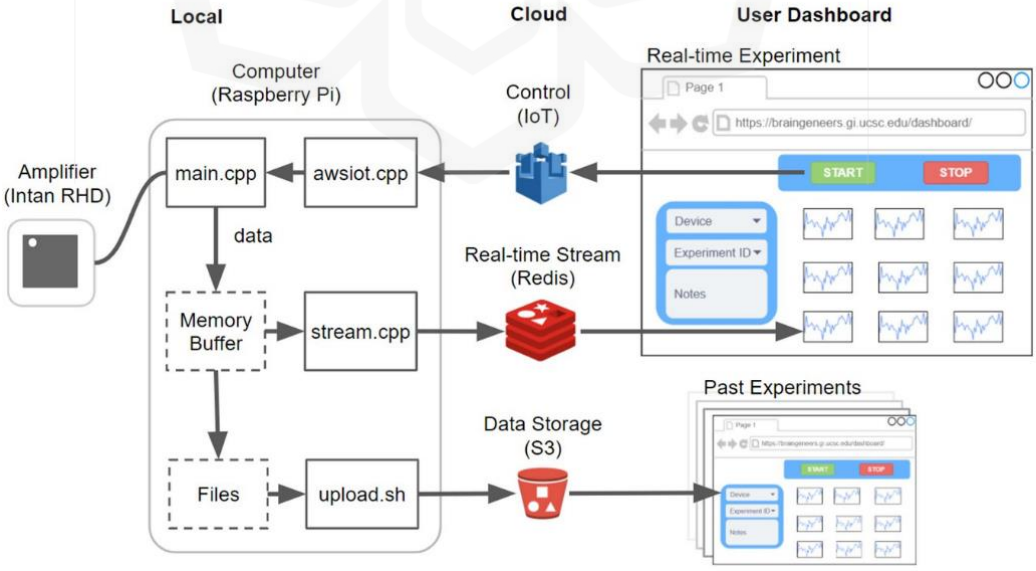


Figure 2.29 Remote System Architecture by Piphys (Voitiuk et al., 2021)

In summary, Raspberry Pi has emerged as a widely utilized platform for serving as the central processing unit in remote monitoring and control systems for cell culture applications. Raspberry Pi's computational capabilities enable researchers to implement sophisticated algorithms and decision-making processes from multiple input sensors including, but not limited to, digital, analog and camera, directly on the device. Furthermore, Raspberry Pi's connectivity options, including Ethernet, Wi-Fi, and Bluetooth, enable remote access and control of the cell culture system. Although alternative development boards like the NVIDIA Jetson, Google Coral, and UP Squared boards are accessible, previous studies in this specific domain have not extensively employed them, despite their similar functionalities. However, the application of the UP Squared board in this study is detailed in our related paper by Shamhan et al. (2023). Table 2.5 summarizes the reviewed system, particularly on the communication framework.

Table 2.5 Summarized Remote Monitoring and Control System for Cell Culture

<table>
<tr><th>System</th><th>Components</th><th>Cloud storage</th><th>Control input</th></tr>
<tr><td>Pérez et al. (2018)</td><td>• Raspberry Pi ARM Cortex-M7 (processor)
• Intel Edison (gateway)</td><td colspan="2">MySQL</td></tr>
<tr><td>Picroscope by Baudin et al. (2022)</td><td>• Raspberry Pi (processor)
• Raspberry Pi Zero (camera)
• Arduino Uno (sensors, actuators)</td><td rowspan="2">S3 PRP</td><td rowspan="3">MQTT broker by Amazon IoT</td></tr>
<tr><td rowspan="2">Piphys by Voitiuk et al. (2021)</td><td rowspan="2">• Raspberry Pi Model 3B+ with expansion shield (processor)
• Intan RHD2132 (sensors)</td></tr>
<tr><td>• Redis (visualization)
• Amazon Glazier</td></tr>
</table>

2.7 AUTOMATION AND MONITORING FOR CELL MANUFACTURING

Based on the literature review, there are several automation and monitoring for cell manufacturing. Doulgkeroglou et al. (2020) has pointed out some of them and among such system is StemCellFactory, an automated system for reprogramming and expanding induced pluripotent stem cells (iPSCs) for disease modeling and drug screening. As illustrated in Figure 2.30, it consists of various cell culture devices that are functionally joined and integrated into a central control system. Among the devices is an automated microscope in which to assess the morphological structure and confluency level of the iPSCs, while deep learning algorithms are used for image classification.

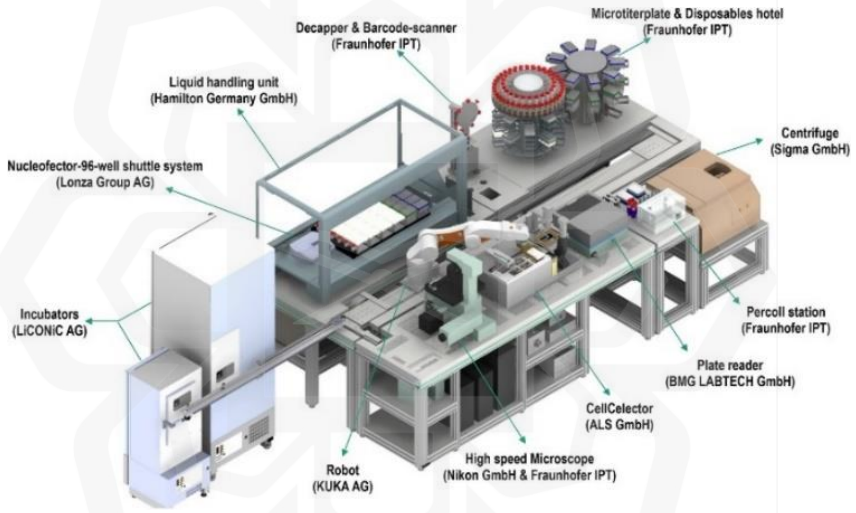


Figure 2.30 StemCellFactory (Doulgkeroglou et al., 2020)

S. Konishi et al. (2021) developed a cell and tissue system that allows for automated culture, stimulation, and monitoring, with the goal of controlling the feedback of organs-on-a-chip, particularly the size of lipid droplets (LDs) in adipose tissue. The system is designed to be controlled by a PC, as illustrated in Figure 2.31, and consists of four major subsystems: the culture subsystem, the sampling and dispensing subsystem, the electrical stimulation subsystem, and the imaging subsystem.

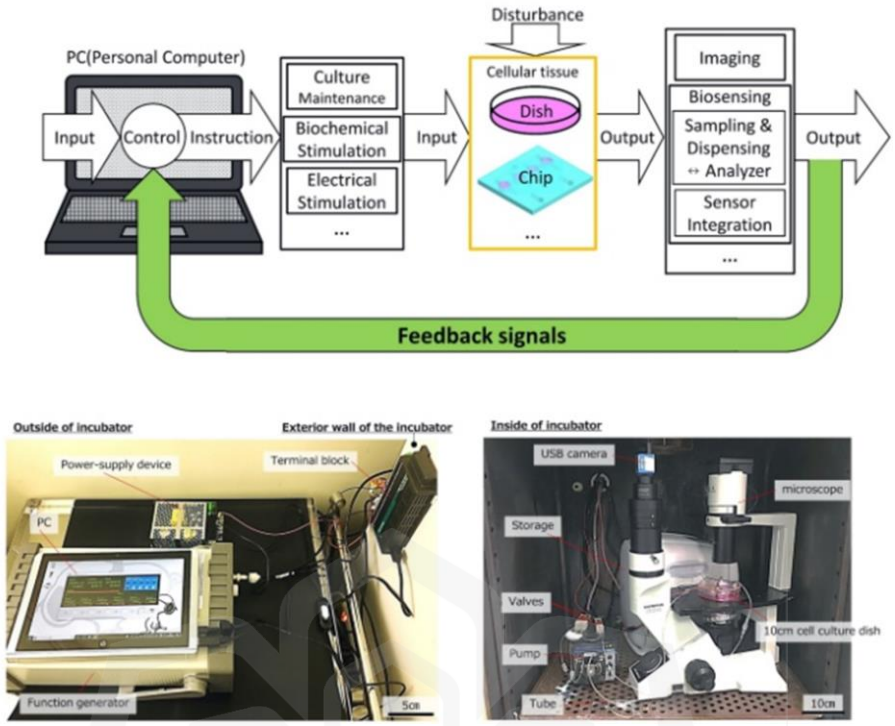
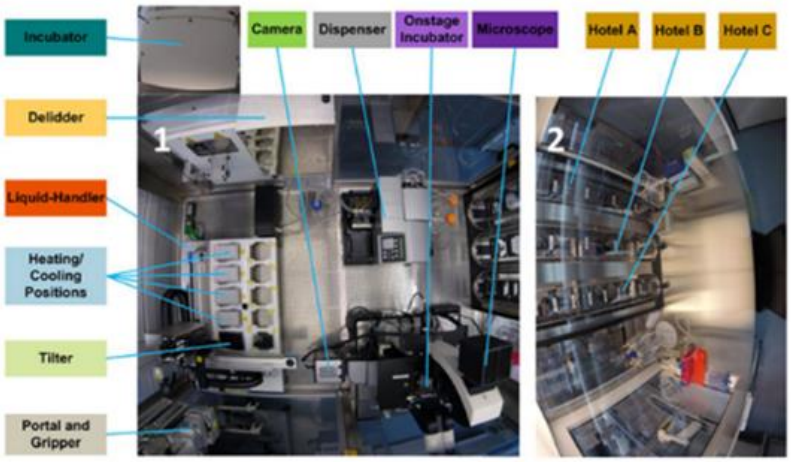


Figure 2.31 A PC-controllable Cell and Tissue System (S. Konishi et al., 2021)

The culture subsystem is responsible for the exchange of culture media and the delivery of biochemical stimuli to the cellular environment. The sampling and dispensing subsystem uses pump, tube, and valves to extract samples from the culture subsystem and dispenses them into separate containers for further analysis, such as storage tips. The electrical stimulation subsystem is used to study the effects of electrical signals on cellular behavior, while the imaging subsystem allows users to monitor culture conditions and visually evaluate the behavior of the cellular tissue.

Another example automated organs-on-a-chip culture was proposed by Kane et al. (2019). As illustrated in Figure 2.32, the system was installed in a Pelican workstation and a four-axis gantry robot was attached to the top of a stainless-steel frame support for handling the substrates and services of other devices within it. The support was connected through rails, allowing the robotic arm to move along three axes, in contrast to other automated systems that only have a fixed rotating robotic arm, which often has limited reach.



(a) Inside view

(b) Outside view

Figure 2.32 Pelican-based Automation System (Kane et al., 2019)

To sum up, in order to automate cell culture handling, three components can be considered including robotic arm, conveyer belt, and tubes. While both the robotic arm and conveyor belts can transport entire petri dishes, the tubes focus on transferring cells between dishes. However, among these, the robotic arm stands out for its superior flexibility, enabling movement along the X-, Y-, and Z-axes for future adaptability. Regardless of any equipment, one of the common issues in developing the system is their high manufacturing costs, involving a complex and expensive process that requires customized batches. This not only results in increased production costs but also limits the scalability of the system. Consequently, the automation and monitoring of cell culture systems has become less accessible to many end users, particularly those with limited budgets. Another limitation is their complexity in scaling up. Often, the development of these systems does not consider edge devices, which are modular and low-cost alternatives to act as central control systems. By incorporating edge devices, cell culture systems can be designed to be more modular and scalable, allowing for easy integration of new components and control systems.

2.8 CHAPTER SUMMARY

This chapter reviewed the basis of cell segmentation research, including common mammalian cellular morphologies, microscopy image modalities, and existing segmentation models. A number of segmentation model refinement mechanisms were discussed in terms of the architecture, dataset condition and prediction perspectives.

Presently, the number of cell segmentation models developed by computational researchers is increasing, but they are not widely adopted by non-technical biologists. This is because most of the developed models require programming scripting knowledge and are not interactive, which is a barrier for non-technical users. To address this issue, several cell segmentation tools were reviewed to investigate the key features in developing a user-friendly interface for cell image analysis. These tools include both local installation-based and cloud-based solutions.

As previously emphasized in the introductory chapter, a continuous demand for an automated system exists due to the laborious nature of the cell culture workflow. According to the existing developments, cell culture automation can either work with robotic arm, conveyer belt, or tubes. All discussions included the advantages and possible limitations, providing valuable references for further chapters.

CHAPTER 3

CELL SEGMENTATION MODELLING AND VALIDATION

3.1 GENERAL OVERVIEW

The main contribution of this study is to demonstrate the improved performance and minimal complexity of the proposed segmentation model on segmenting the features of fibroblast cell images at different magnification and confluency levels to be deployed on an edge device and is not to focus on developing new novel model architecture for the segmentation task. The complete workflow of cell segmentation modelling and analysis is displayed in Figure 3.1. This chapter begins by outlining the process of data collection and preparation for training. Subsequently, various model candidates will be introduced, refined from a baseline architecture, and trained using specified hardware and software settings. The validation of model performance will then be designed, incorporating real-world observations from field experts and assessments based on the inherent growth dynamics of the cells.

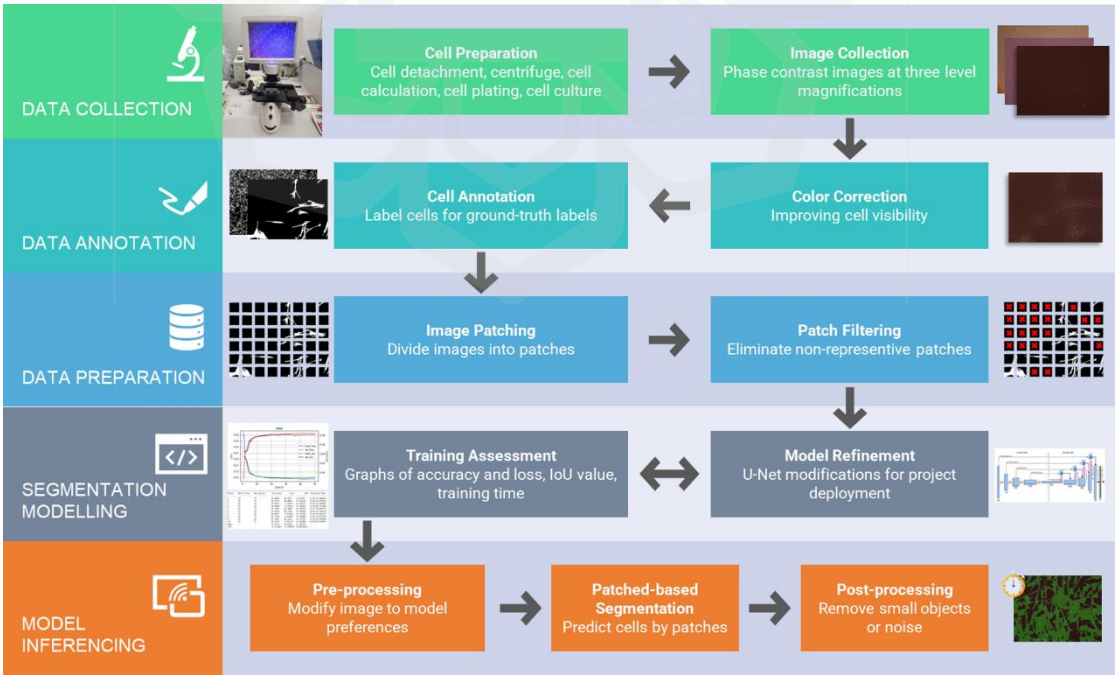


Figure 3.1 Segmentation Modelling and Analysis Workflow

3.2 EXPERIMENTAL SETUP

Before delving into the model development and training process, this section addresses key aspects that lay the groundwork for a robust and effective approach. These critical elements encompass the baseline architecture, hardware setup, training settings, evaluation metrics, and loss functions. A detailed baseline architecture is shown, where the standard U-Net is introduced and tailored to serve as the benchmark for subsequent comparisons. The hardware setup, specifically tailored for cell segmentation model training, is selected, accompanied by specific training settings such as the number of epochs, batch size, learning rate, and optimizer type. Following this, the choice of evaluation metrics is discussed, aligning with the unique characteristics of our dataset to ensure a good assessment of model performance. Lastly, the section outlines the selection of loss functions, strategically employed to quantify errors during training and enhance overall model metrics. This comprehensive preparation sets the stage for a methodical and effective model development and training phase.

3.2.1 Baseline Architecture

U-Net CNN was used as the baseline for comparison because it previously demonstrated satisfactory performance for biomedical images with low training images, which is relevant to our study. Any modification of a model will be done on the standard U-Net. An original U-Net was replicated having five layers deep encoder and decoder, with filter numbers of 64, 128, 256, 512, 1024, as tabulated in Table 3.1. However, a reduced version of U-Net is preferred because it has fewer parameters with 1,940,817 weights and was proven to perform more or less than standard U-Net with 31,030,593 trainable parameters. The detailed comparison of both architectures is shown in Table 3.1, Thus, this becomes the first step to reducing the model complexity of trainable parameters while maintaining its performance. In this experiment, U-Net starts with 16 filters of convolution layer and doubling for each layer at the encoding stage and vice versa at the decoding stage as illustrated in Figure 2.14 under the literature review.

Table 3.1 Baseline Architecture Details

Architecture		U-Net (original)	U-Net (baseline)
Block	**Layer**	**Output shape**	**Output shape**
Encoding Block 1	Conv2D	(256, 256, 64)	(256, 256, 16)
	Dropout (0.3)	(256, 256, 64)	(256, 256, 16)
	Conv2D	(256, 256, 64)	(256, 256, 16)
	MaxPooling2D	(128, 128, 64)	(128, 128, 16)
Encoding Block 2		(64, 64, 128)	(64, 64, 32)
Encoding Block 3		(32, 32, 256)	(32, 32, 64)
Encoding Block 4		(16, 16, 512)	(16, 16, 128)
Connecting Block	Conv2D	(16, 16, 1024)	(16, 16, 256)
	Dropout (0.3)	(16, 16, 1024)	(16, 16, 256)
	Conv2D	(16, 16, 1024)	(16, 16, 256)
Decoding Block 1	Conv2DTranspose	(32, 32, 512)	(32, 32, 128)
	Concatenate	(32, 32, 512)	(32, 32, 128)
	Conv2D	(32, 32, 512)	(32, 32, 128)
	Dropout (0.3)	(32, 32, 512)	(32, 32, 128)
	Conv2D	(32, 32, 128)	(32, 32, 512)
Decoding Block 2		(64, 64, 64)	(64, 64, 256)
Decoding Block 3		(128, 128, 32)	(128, 128, 128)
Decoding Block 4		(256, 256, 16)	(256, 256, 64)
Classification	Conv2D	(256, 256, 1)	(256, 256, 1)

3.2.2 Computer Hardware and Training Settings

The Anaconda Python distribution is recommended to accommodate the program and libraries on Windows 10 due to its ease of package movement and deployment. Each model was implemented and trained on an NVIDIA GeForce RTX 3060 GPU with 12 GB VRAM using the TensorFlow GPU v2.8.0/ Keras framework.

Adam optimizer was used, with a learning rate of 1e-3. All models are trained for a maximum of 50 epochs with a batch size of 24 training samples, with checkpoints used after each epoch to save only the best model with the lowest validation loss, employed to predict the segmentation on the test set.

The training employed in this study involves utilizing k-fold cross validation, as illustrated in Figure 3.2, to rigorously test the model's performance toward a variation of datasets. To achieve this, firstly, the data was split into k equal-sized folds. Then, the model was trained k times, each time using a different fold as the validation set and the remaining folds as the training set. 20 % of annotated data are allocated for the test data.

In this study, the value of k is set to 5, which means that each iteration will split the dataset into an 80% training set and a 20% validation set.

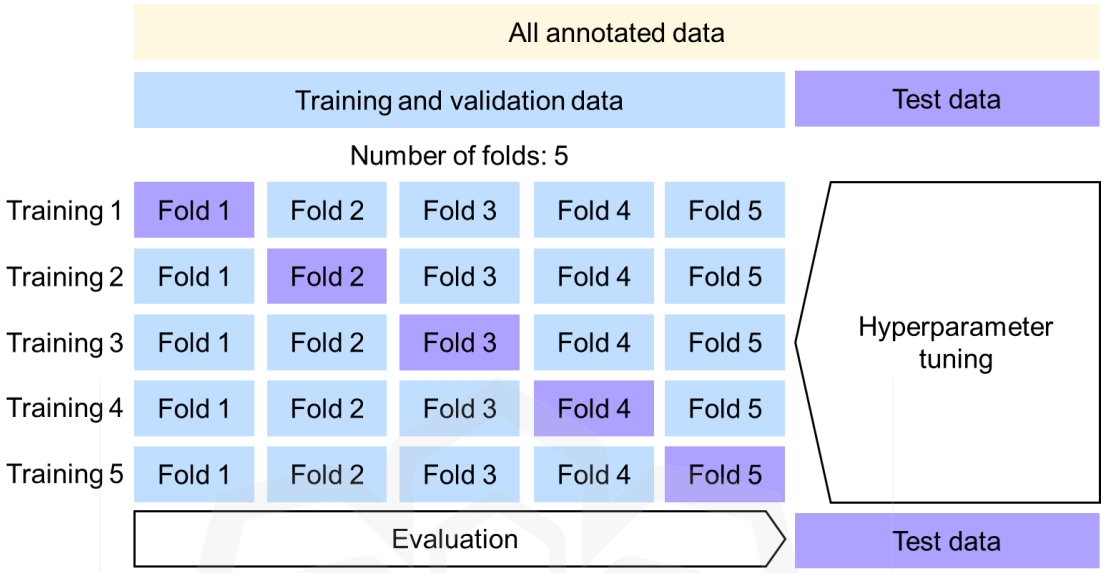


Figure 3.2 K-fold Cross Validation

3.2.3 Evaluation Criteria

In this study, IoU, also known as the Jaccard index, was used as the main evaluation indicator to select the best result. Pixel accuracy and F-score are also considered as the alternative metric to evaluate the model performance during the training process. In addition, the confusion matrix was organized into four terms: true positive, false positive, true positive, and false negative. In this context, the terms are described in Table 3.2.

Table 3.2 Confusion Matrix Elements

Terms		Descriptions		Remarks
True Positives	(TP)	-	True predictions on cell pixels	Merit
False Positives	(FP)	-	Not predicted cell pixels	Demerit
False Negatives	(FN)	-	Wrong predictions on background pixels	Demerit
True Negatives	(TN)	-	True predictions on background pixels	Merit

The elements will help in defining the evaluation metrics. Only the first three of the elements in Table 3.2 are important for calculating semantic segmentation evaluation metrics and providing insights of specific segmentation errors. Thus, the considerable metrics are computed as below:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

where it measures the proportion of pixels that are correctly classified as belonging to both cells and background, indicating under-segmentation errors either on cell or background predictions. However, this metric may be heavily influenced by class imbalance as high accuracy mostly represents true predictions of background pixels but has failed to identify cell pixels. Thus, other metrics were introduced to specifically focus on the errors of cell predictions which are:

$$IoU = \frac{TP}{TP + FP + FN} = \frac{R_{GT} \cap R_{PS}}{R_{GT} \cup R_{PS}} = \frac{labels \cap predictions}{labels \cup predictions} \tag{2}$$

where it measures the degree of overlap between predicted segmentation mask and ground truth mask, indicating localization errors of cell predictions.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

where it measures the proportion of true positives among all the predicted pixels without penalizing the false negatives, specifically indicating over-segmentation errors of cell predictions.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

where it measures the proportion of true positives among all the ground truth pixels without penalizing the false positives, specifically indicating under-segmentation errors of cell predictions.

$$Fscore = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{5}$$

where it weights average of precision or recall, indicating either under-segmentation or over-segmentation on cell predictions.

3.2.4 Losses

Binary cross-entropy (CE) is a general loss function for binary segmentation by penalizing the wrong predictions more than to reward the right predictions, differentiating positive/ negative examples correctly. However, in this study, we dealt with biomedical images which have an imbalanced object distribution where images dominated by background pixels instead of object-of-interest pixels. Furthermore, it is known that this loss function has a high probability more than 0.5 of ground truth classification still incur loss with non-trivial magnitude, contributing major no useful learning and the model cannot differentiate positive or negative examples efficiently which is not good for imbalanced dataset.

To address that, balanced cross-entropy (BCE) introduces a weighting factor $\propto \in [0,1]$ but still, it does not differentiate easy or hard examples. Therefore, focal loss (FL) was used as a derivation of the cross-entropy loss function, as shown in Equation (9), that would down-weight easy examples and focus training on hard negatives, mitigating the impact of the dominant background pixels.

$$CE(p) = -y \log p - (1-y)\log(1-p) \quad (6)$$

$$CE(p) = -\log p \quad (7)$$

$$BCE(p) = -\alpha \log p \quad (8)$$

$$FL(p) = -\alpha(1-p)^{\gamma}\log(p) \quad (9)$$

where $y \in \{0,1\}$ are ground truth classes and $p \in [0,1]$ are the estimated true class probability with y =1.

Another loss that was used is Dice loss (DL). It aims to minimize the mismatch or maximize the overlap regions between ground truth and predicted segmentation and optimize segmentation metrics such as Dice coefficient and IoU.

$$Dice\ coefficient = \frac{2|R_{GT} \cap R_{PS}|}{|R_{GT}| + |R_{PS}|} = \frac{2 \cdot y \cdot p + 1}{y + p + 1} \quad (10)$$

$$DL(p) = 1 - \frac{2 \cdot y \cdot p + 1}{y + p + 1} \quad (11)$$

where R_{GT} is the ground truth region and R_{PS} is the predicted segmentation region.

3.3 DATA ACQUISITION AND ANNOTATION

This study utilized two datasets on fibroblast cell growth, comprising a total of 400 images with a dimension of 2048 x 1536 pixels. These images were obtained using phase-contrast microscopy from the Institute of Medical Research (IMR) Malaysia and the Institute of Medical Science Technology of Universiti Kuala Lumpur. The detailed information of data collection is shown in Table 3.3.

Table 3.3 Detailed Information About Data Collection

<table>
<tr><td>Provider</td><td colspan="3">• Institute of Medical Research (IMR) Shah Alam
Period: Dec 2021 –Jan 2022
• UniKl-MESTECH Kajang
Period: Sep 2022 – Oct 2022</td></tr>
<tr><td>Cell type</td><td colspan="3">Skin cell (fibroblast) culture images of preparation days and growing days</td></tr>
<tr><td>Image type</td><td colspan="3">Magnification level of x4, x10, x20</td></tr>
<tr><td>Modality</td><td colspan="3">Phase-contrast microscopy</td></tr>
<tr><td>Image sizes</td><td colspan="3">2048 x 1536 pixels</td></tr>
<tr><td rowspan="2">Total images</td><td>x4</td><td>x10</td><td>x20</td></tr>
<tr><td>101</td><td>94</td><td>121</td></tr>
</table>

In the experiments, the fibroblast cells were allowed to grow and reach maximum confluency for up to 7 days, based on the initial number of cells placed in the petri dish. The datasets show significant differences in terms of their features, particularly the shape of the cells, as they grow. Specifically, cells in dataset A have spherical shapes while cells in dataset B have elongated shapes, as shown in Figure 3.3. Additionally, dataset A only includes images from the initial culture day, while dataset B includes images from the following days. This research focused on dataset B, which consists of 300 images and is the most accurate representation of the growing cells. Meanwhile, dataset A is only suitable for counting the number of cells, not for measuring their growth.

(A) Day 1 cells (B) Growing cells

Figure 3.3 Dataset Types of Different Cell Structures

Based on the zoom capability of the EVOS™ XL Core Imaging System at the data collection site, Dataset B was divided into three subsets based on image magnification: x4, x10, and x20, with approximately 100 images in each subset, as listed on Table 3.3. Their distinct features can be seen in Figure 3.4. The inner cell texture appears plain in the x4 dataset and becomes progressively complex at higher magnification levels.

(a) x4 (b) x10 (c) x20

Figure 3.4 Dataset Types of Different Magnification Levels

Brightness, contrast, and saturation are three manipulated variables that can be adjusted on the microscope. To define the experiment's standard procedure, their values are set to be constant for all magnification levels. However, the microscope only provides a sliding bar for adjusting contrast and saturation, as demonstrated in Figure 3.5, which can result in their values being estimated. Therefore, both contrast and saturation are set to the default values of the microscope, which are automatically positioned in the middle of the sliding bar to avoid any inconsistencies.

Figure 3.5 Setting Configurations Displayed on EVOS^{TM} XL Core Imaging System

Unlike brightness, which can be customized based on the magnification levels and ambient lighting conditions, ranging from 35 to 60. It is noticeable that at higher magnification levels, images seem darker, thus demanding more brightness, shown in Figure 3.6. The setup information is listed on Table 3.4.

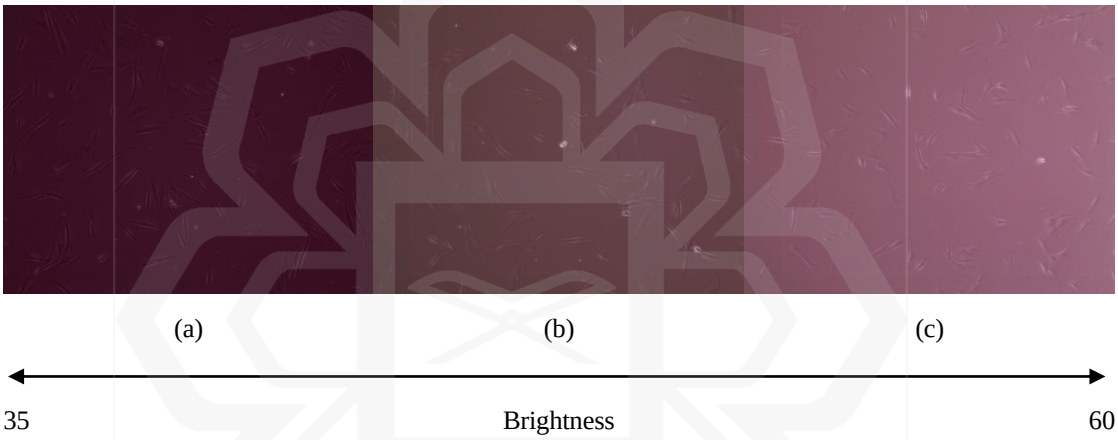


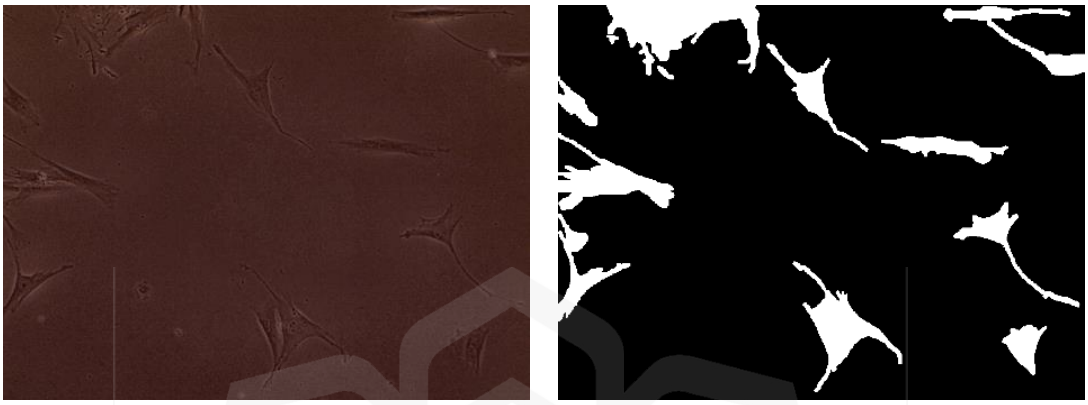
Figure 3.6 Example of x4 Images Ranging Brightness

Table 3.4 Microscope Control Parameters

Instrument	**Parameter**	**Control protocol**
EVOS XL Core Imaging System	Dimension	2048 x 1536
	Modality	Phase-contrast
	Brightness	35 - 60
	Contrast	Default (50)
	Saturation	Default (50)
	Sharpness	Analog

The implementation setup was chosen because our subsequent study aims to deploy the model to an edge device that can be easily used by individuals with little experience. Next, images are chosen from dataset B and as previously stated, split for training-validation set and test set to 80% and 20%, respectively. The training-validation set was then randomly divided by k-fold cross validation into 80% and 20%,

respectively. During the training process, there are two types of images, as displayed in Figure 3.7, that are used as inputs: the original images and their corresponding ground truth images that have been annotated using either the online website Apeer or the offline program LabelStudio.



(a) Original image (b) Ground truth

Figure 3.7 Example of Inputs for Segmentation Model Architecture

Regardless of any annotation software, it is important to note that after the annotation process, the ground truth labels should be carefully checked to ensure that they only have two unique pixel values, which are 0 (black) and 1 (white). This is crucial to ensure that the ground truth labels are compatible with the training process and will produce accurate results.

3.4 DATASET PRE-PROCESSING

Model performance and complexity can be influenced by the conditions of the dataset. In this study, we consider deep learning approaches to address these challenges. The images in our dataset present a challenging scenario where the edge line between the background and cell pixel color is indistinguishable, thereby posing a significant hurdle to annotate the data and for the model to train and segment the cells with precision. Moreover, biomedical images commonly have high resolution pixels which will outburst the available hardware capability. To address these issues, we believe that the problems can be solved before training by applying image correction, image patching, and patch filtering techniques as needed.

3.4.1 Image Processing for Inappropriate Imaging Quality

Common issues with biomedical images are that they have low contrast and high noise (Naidu et al., 2021). These issues may lead to reduced discriminative features, poor generalization, and reduced training efficiency. Thus, the study suggested two types of image processing techniques, which are Contrast Limited Adaptive Histogram Equalization (CLAHE) for contrast enhancement and Total Variation Denoising (TVD) for noise reduction, with respect to the dataset conditions.

A. Contrast Limited Adaptive Histogram Equalization (CLAHE)

CLAHE was introduced by Zuiderveld (1994) and was chosen to correct the image color for better distinguished annotation and segmentation process. Compared to standard histogram equalization, it often requires experiment-specific parameter tuning. CLAHE is applied per tile to improve the visibility level of objects in foggy images by stretching the histogram, limiting the contrast, and performing bilinear interpolation at the edges to match the next tiles. First, the images are converted to LAB space, where the L channel contains luminosity or intensity information, and the A and B channels contain color information. The channels will be separated, and the filter will be applied only to the L channel, with the channels being merged back in for the original image. The filter is imported from "createCLAHE" function from the opencv module. Figure 3.8 illustrates the potential results before and after the application of applying CLAHE.

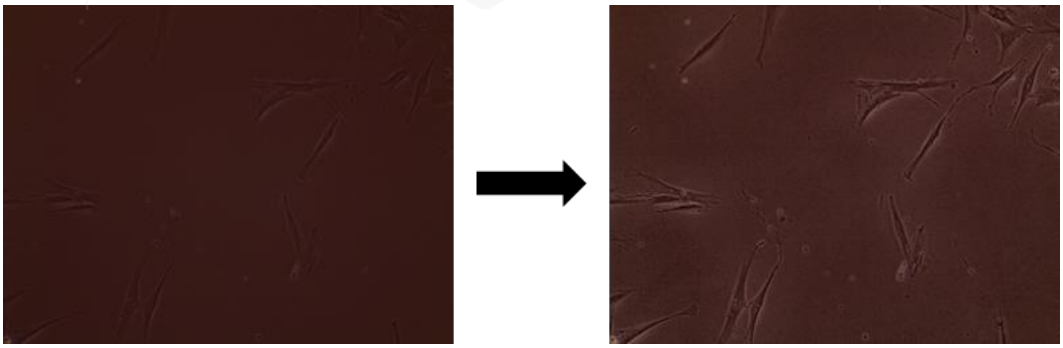


Figure 3.8 CLAHE Image Processing

B. Total Variation Denoising (TVD)

One common side-effect of using the CLAHE technique is the introduction of noise into the image. This can be especially problematic when dealing with images that have important details or edges that need to be preserved. In such cases, it is advisable to use denoising techniques in conjunction with CLAHE to achieve a better overall result.

One of the denoising algorithms that was selected for this study was TVD (Chambolle, 2004), which has been shown to effectively reduce noise while retaining the benefits of CLAHE in terms of improved visualization of object edges. To implement this denoising filter, the "denoise_tv_chambolle" function was imported from the "restoration" module in scikit-image. It is applied directly after the CLAHE algorithm and its regularization parameter or weight of 0.1 was set for each trial. This denoising filter was mainly applied for the segmentation tasks and was not applied to the training datasets, as no significant improvement in performance was observed.

3.4.2 Image Patching for High-Resolution Images

Each image has a resolution size of 2048x1536 pixels based on data collection. For training, the images are deemed too large to be handled by the computational power available at the time. To avoid heavy loss of model training, the images were patched into a number of smaller input sizes, as portrayed in Figure 3.9, each of which can still represent the object features; thus, the optimal patch size is dependent on the datasets. The expected number of patches that will be generated can be calculated using the following equation:

$$Number\ of\ patches = \frac{Image\ width}{Target\ size} \times \frac{Image\ height}{Target\ size} \tag{12}$$

For example, if the images are patched into 256x256 and 512x512 pixel images, each original image should yield 48 and 12 image patches, respectively. This method reduces the computational power needed for training and indirectly expands the number of images available for the training and validation sets.

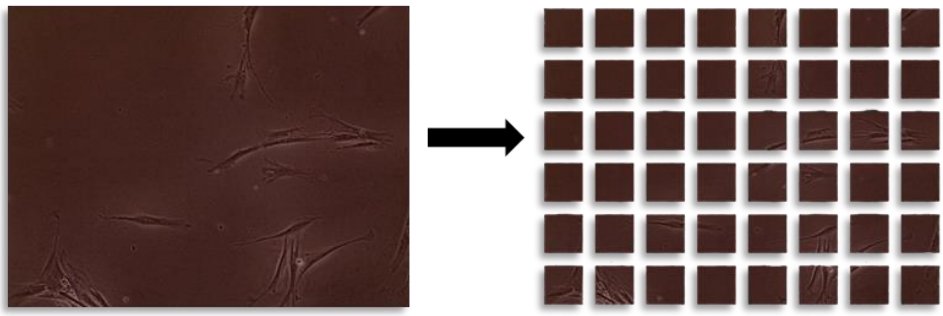


Figure 3.9 Image Patching

The analysis of patch dimensions is limited to 256x256 and 512x516 because smaller dimensions can negatively impact the representative features of the patches, while larger dimensions cannot be trained efficiently using the current hardware available.

3.4.3 Patch Filtering for Insignificant Feature Training

In image analysis, having an imbalanced dataset can significantly hinder the accuracy of the model. In the context of this study, an imbalanced dataset can occur when training images that contain either cells or background only. Therefore, it is crucial to employ precautions that help to balance the dataset. One such approach is filtering out non-representative patches that contain an excessive number of background pixels. These patches may not provide enough relevant information for the model to accurately identify the object of interest within a training batch.

To illustrate this point, consider Figure 3.10, where an example of imbalanced patches is shown. In this example, patches with no cell pixels occupied were removed. By doing so, the dataset becomes more balanced, and each patch has sufficient numbers of both cell and background pixels, enabling the model to identify significant feature differences for every training batch.

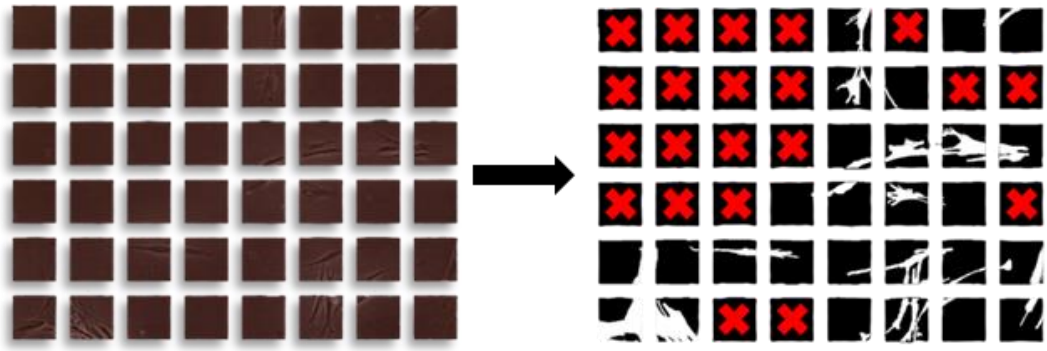


Figure 3.10 Patch Filtering

As previously mentioned, multiple trials are conducted through k-fold cross validation to evaluate the consistency of the model's performance across the dataset. Patch filtering plays a critical role in this process since the random division of data during training-validation may result in fewer complex or insignificant images for certain training batch within the epochs, leading to the invalidation of specific trials during the evaluation process. The estimated number of patches left after filtering are shown in Table 3.5.

Table 3.5 Number of Patches Available for Segmentation Modelling

Datasets	**x4**	**x10**	**x20**
Training images	15		
Generated patches	720		
Filtered patches	691	704	671

3.5 SEGMENTATION MODELLING

The modelling undergoes three phases. The idea begins with the implementation of multi-scale convolution blocks on U-Net to extract features at multiple scales or resolutions from microscopy images. This helps the model capture both low-level and high-level information for the segmentation task. The multi-scale mechanism has been introduced by the Inception network and several works have applied the mechanism on U-Net. Based on the review, the architecture is developed by replacing the U-Net convolutional blocks with different concurrent kernel sizes.

Then, the feature map of convolutional paths is concatenated to be an output for the following layers. However, the selection of concurrent kernel sizes seems to be varying in different experiments. In this case study, we proposed a different strategy from previous work to identify the best combinations for a multi-scale model, and trials are done to investigate their potential to what extent.

3.5.1 Phase 1: Multi-scale Path for Capturing Various Receptive Field

Rather than randomly combining multi-scale blocks, the trials are initially carried out by replacing the original kernel size 3x3 of U-Net convolutional layers with 5×5, 7×7, and 9×9, and their performance was recorded separately (Malik et al., 2023b). The best of the allocated kernel sizes will be chosen and paired in the final model. In conjunction with the specified number of folds for cross validation in this study, each configuration was subjected to five trials. Based on Table 3.6, architectures with kernel size 3×3 and 5×5 have significant performance to each dataset compared to the others. Moreover, two out of five trials have the possibility to be invalidated when using kernel size 7×7 and 9×9 to train the datasets. Therefore, both kernel sizes are included for the multi-scale convolution blocks as illustrated in Figure 3.11.

Table 3.6 Ablative Study on Appropriate Kernel Sizes

Model type		**Intersection over Union (IoU)**		
		Training datasets based on magnification levels		
		x4	**x10**	**x20**
Kernel sizes	**3x3**	0.707±0.002	0.661±0.024	0.626±0.017
	5x5	0.709±0.002	0.670±0.036	0.678±0.030
	7x7	0.704±0.009	0.652±0.032	0.596±0.074
	9x9	0.695±0.012	0.592±0.048	0.613±0.054

According to Table 3.1, a dropout layer is placed between the first and second convolution layers of the convolution path to improve the generalization of the model. The output of both kernel sizes is then concatenated to be an input for the subsequent layer. However, the concatenation dramatically increases the depth dimension of the input to the subsequent layers as well as results in considerably more parameters and higher complexity. In order to reduce the depth, each concatenated feature map will be

followed by a 1x1 convolution layer as demonstrated in Figure 3.11. The multi-scale model with and without the 1x1 convolution layer produces 7,063,937 and 9,238,705 trainable parameters, respectively. In terms of training time, the former takes approximately 1 minute less than the latter to train for 50 epochs. As a result, this initiative prevents larger space use and lesser time complexity.

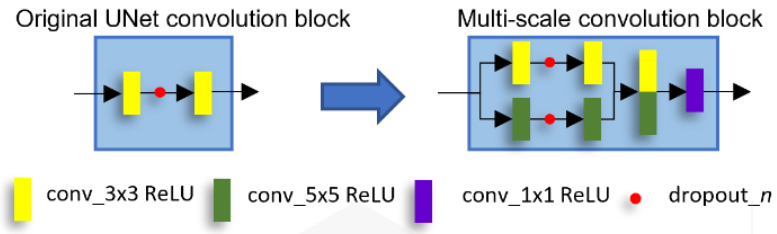


Figure 3.11 Multiscale Convolution Block

w_{ij} represents the output features of a convolutional layer where i is the kernel sizes, and j is the order of convolutional layer. Thus, x_i represents the output features obtained from a convolutional path of different kernel sizes. X_f is the output features of the proposed convolutional block and is computed as follows:

$$x_1 = w_{3,2}(w_{3,1} - d_{3,1}) \quad (13)$$

$$x_2 = w_{5,2}(w_{5,1} - d_{5,1}) \quad (14)$$

$$X_c = concatenate[x_1, x_2] \quad (15)$$

$$X_f = w_1 X_c \quad (16)$$

3.5.2 Phase 2: Dilated Path for Reduction of Model Parameters

The 5x5 kernel may be more suitable for capturing more fine-grained features, however, it significantly increases the number of trainable parameters of the baseline architecture. Therefore, the study proposed the kernel path be converted into the 3x3 kernel with dilation rate 2 to imitate the 5x5 kernel nature of capturing larger-scale patterns of the image. The structure of phase 2 is similar to phase 1 but with lower parameters. The new convolutional is illustrated in Figure 3.12 and the new convolutional path is computed as in Equation (17).

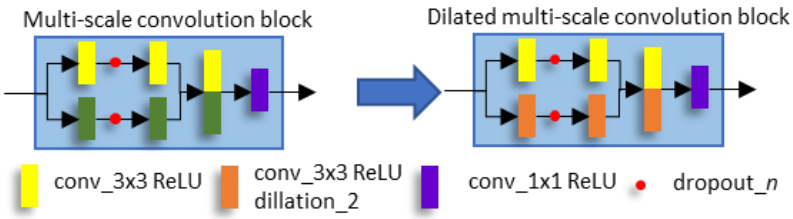


Figure 3.12 Dilated Path of Multi-Scale Convolution Block

$$x_2 = w_{3_{r=2},2}\left(w_{3_{r=2},1} - d_{3,1}\right) \tag{17}$$

where $w_{i_{r=2},j}$ is the dilated convolution layer with dilation rate, r, of 2.

3.5.3 Phase 3: Residual Connection for Preserving Input Information

Dilation rate determines the spacing between the kernel points and a higher dilation rate means that the kernel will skip more pixels in between each calculation, resulting in a larger receptive field, as displayed in Figure 2.19. Based on the fact, information from some regions of the input may not be captured as well. By adding residual connections, the network may retain information from the input and help to alleviate any information loss caused by the dilated convolutional kernel. 1x1 kernel convolution is also added on the residual path to adjust the number of channels in the input to match the number of channels in the output of the convolutional layer. The proposed convolution block and final model design are illustrated in Figure 3.13 and Figure 3.14, respectively.

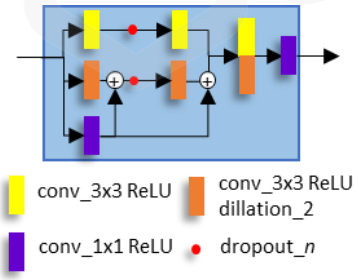


Figure 3.13 Proposed U-Net Convolution Block

$$x_2 = w_{3_{r=2},2}\left(w_{3_{r=2},1} + w_{1,1} - d_{3,1}\right) + w_{1,1} \tag{18}$$

where $w_{1,1}$ is the output features of 1x1 convolutional layer on the residual path from the input added to both output features of convolution layers on the dilated path.

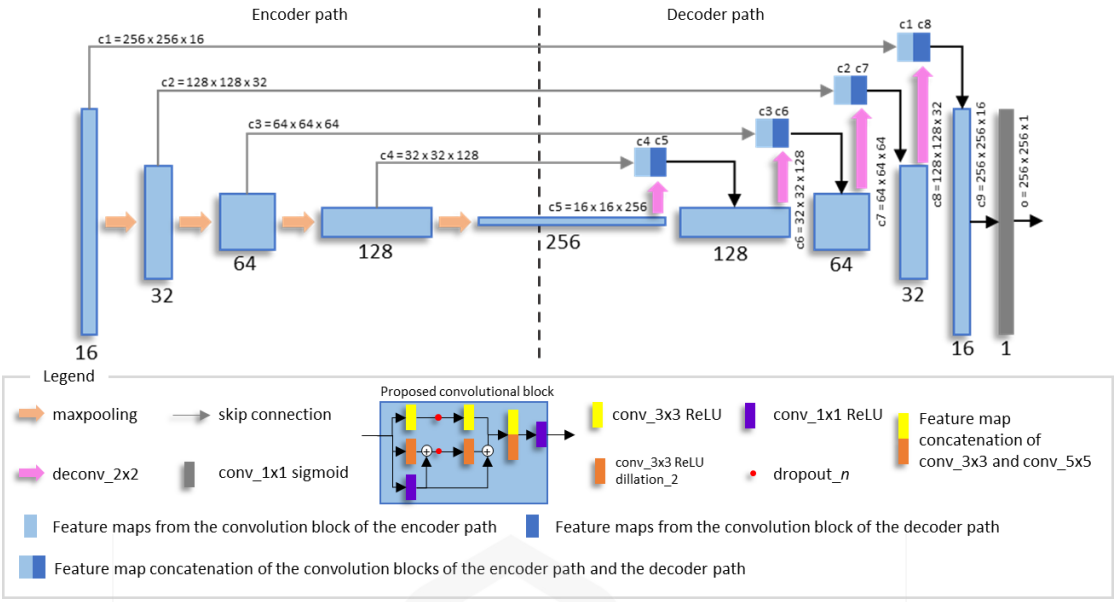


Figure 3.14 Proposed Model Architecture

3.6 VALIDATING MODEL ROBUSTNESS THROUGH EXPERIMENTS

To assess the effectiveness of the segmentation pipeline for a deployment, it is critical to compare it with end-user capabilities and evaluate its adoption in real-world scenarios in confluency value predictions. To achieve this, two methods have been suggested: conducting an online survey with experts and performing a lab experiment. This section will present the detailed experimental setup of each method. In this analysis, the model considered most effective through the evaluation outlined in the preceding methodology will be mainly selected for any comparison as the effectiveness factor is the focus and it has proven the best in performance previously.

3.6.1 Model Validation Against Human Evaluation

This survey aimed to compare the effectiveness of the proposed segmentation model and human evaluation in analyzing cell images with different confluency and magnification levels in terms of precision and consistency. To achieve this, a Google form survey was created and distributed to individuals with diverse biology backgrounds. The survey included a set of cell images with varying confluency and magnification levels that were chosen for segmentation. The overview of this analysis is as illustrated in Figure 3.15.

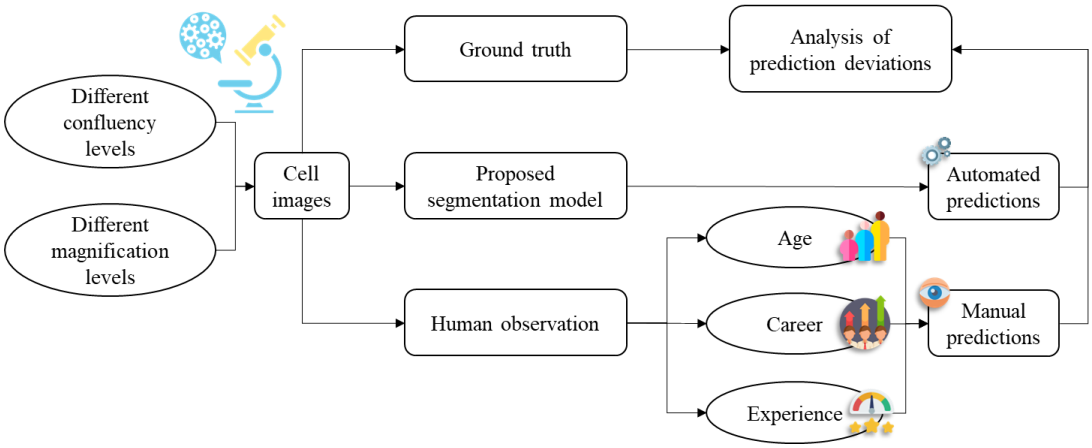


Figure 3.15 Overview of Computer-Human Comparison Analysis

Basically, the survey was divided into two sections. Section A, as shown in Figure 3.16, requires the respondents to provide their background information about the subject matter, such as age group, current position, and years of experience in cell culture. This information was crucial for ensuring a well-rounded evaluation of the system and accounting for any potential biases. Section B, as exhibited in Figure 3.17, contains eight images of fibroblast cell growth and requires them to provide an estimated percentage value of cell confluency within each image subjectively. The images feature various image settings and magnifications. By collecting these estimations alongside the background information, it became possible to analyze the relationship between respondents' profiles and their confluency predictions. Thus, this analysis is also able to determine whether specific background factors influenced the accuracy of predictions or not.

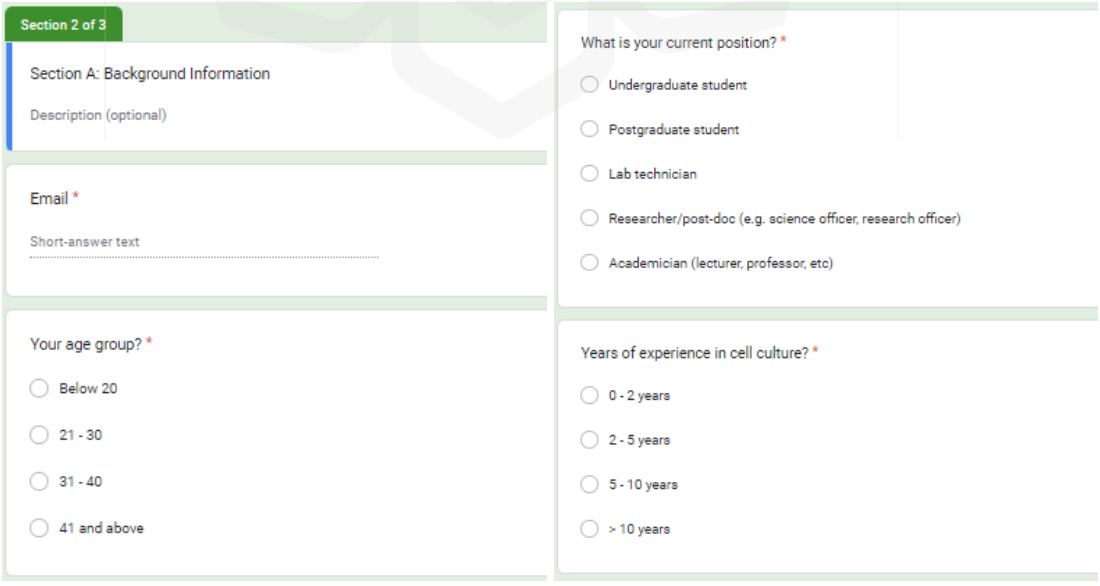


Figure 3.16 Survey on Biology Background Information

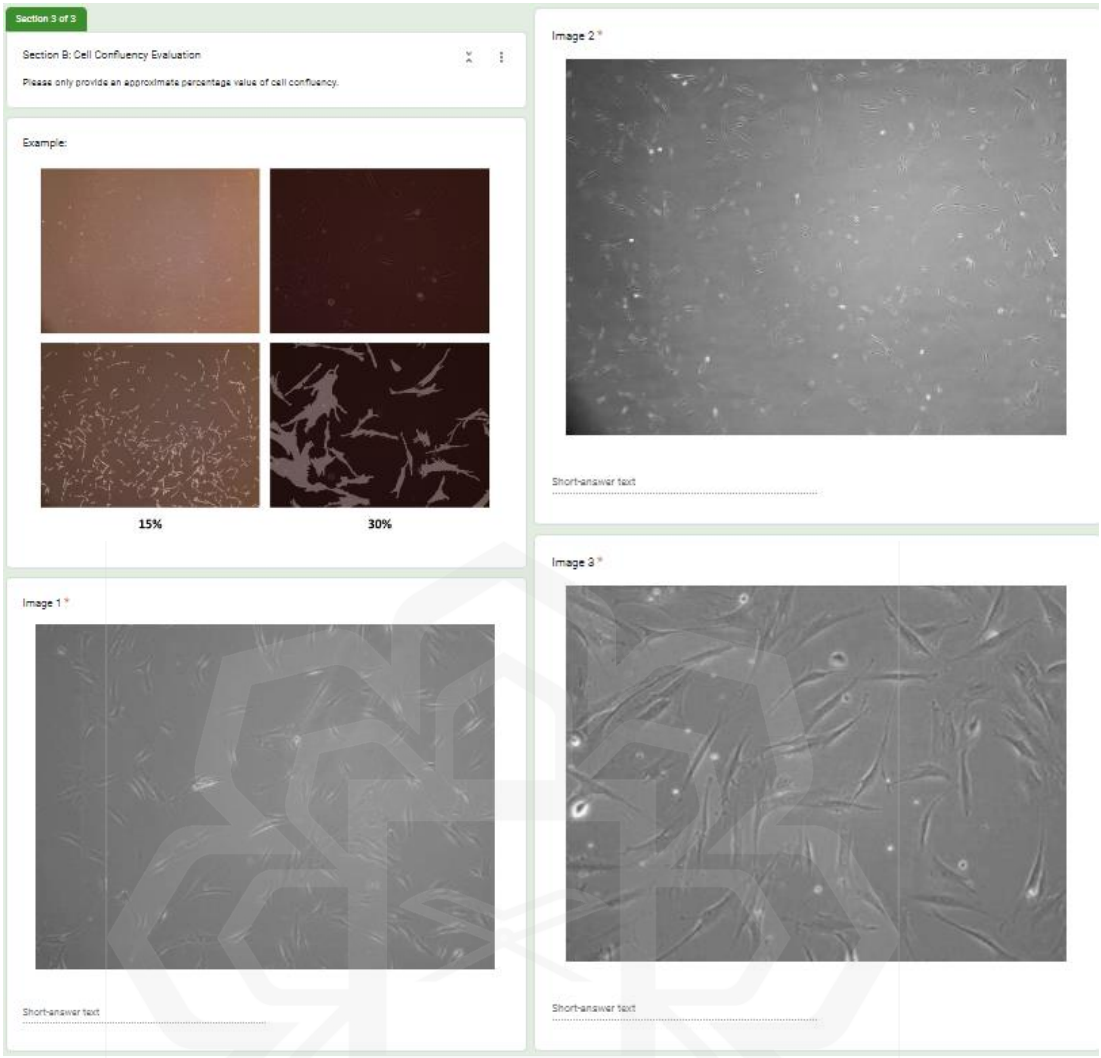


Figure 3.17 Survey on Subjective Predictions of Confluency Value

3.6.2 Model Validation Towards Cell Growth Cycle

The experiment was designed to achieve two specific objectives. The first objective was to analyze the relationship between the number of cells placed in a petri dish on the preparation day and the growing confluency. The second objective was to evaluate the performance of the models by testing it on x4 images with varying levels of confluency, ranging from low to high. The reason for using only x4 images for testing the segmentation model is because they represent the widest range of confluency within the petri dish. At this magnification level, the images capture a larger area of the petri dish, allowing for a more representative sample of the distribution of cells. The consideration better simulate real-world scenarios, where the distribution of cells in a petri dish is often uneven.

Two hypotheses were proposed for this experiment. The first hypothesis suggested that as the number of cells in the petri dish increased, the growing confluency would also increase, resulting in a stronger correlation between the two. The second hypothesis proposed that as the confluency increased, the difficulty level of segmenting the cells would also increase, as more cells would be in close proximity, making it challenging to precisely segment them.

To conduct this experiment, a two-day protocol was devised. On the first day, the researchers focused on sample preparation, as the steps depicted in Figure 3.18. The cell plating should be as illustrated in Figure 3.19. In total, ten samples were prepared, with the number of cells ranging from 10,000 to 100,000.

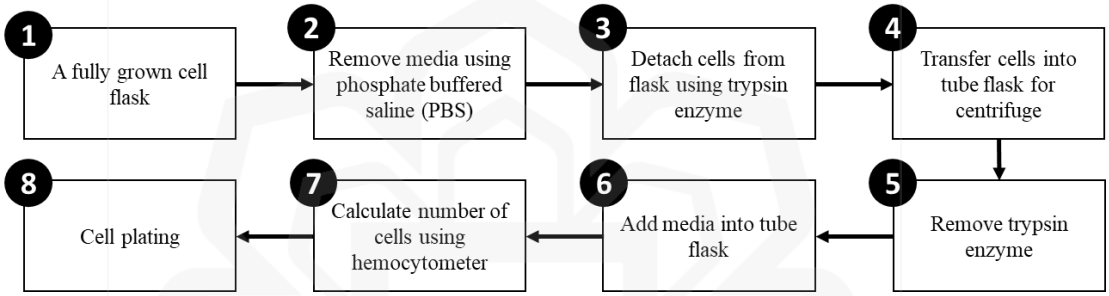


Figure 3.18 Steps for the Experimental Real-world Conditions

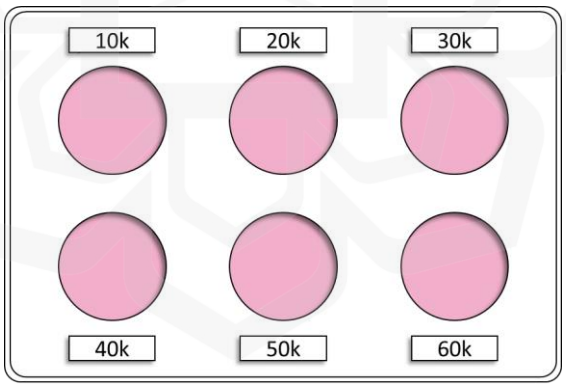


Figure 3.19 Cell Plating

The number of cells of each dish is determined by the volume of cell solution. To achieve that, the calculations of Step 7 in Figure 3.18 are outlined below:

$$average\ cells\ of\ both\ chambers, z = \frac{\left(\frac{\sum_{i=1}^{4} x_i}{n_x} + \frac{\sum_{i=1}^{4} y_i}{n_y}\right)}{number\ of\ chambers\ (2)} \quad (22)$$

where x_i and y_i are respectively the total number of cells on chamber x and y, and n_x and n_y are the number of x_i or y_i.

$$\text{Number of cells per ml } \times 10^5, \text{N } = \text{ z} \times \text{dilution factor (2)} \times 10^4 \quad (23)$$

where the dilution factor is the total ratio of cell volume and trypan blue volume, and both are measured in microliters. For example, if the ratio is 1:1, the dilution factor would be 2.

$$\text{Total millimeter per number of cells, M} = \frac{\text{Desired cell numbers}}{N} \times 1000 \quad (24)$$

where M is measured in microliters.

On the second day of the experiment, image capturing, and segmentation were carried out. During the occasion, an issue was encountered due to the presence of collective cells in the center of the petri dish, as displayed in Figure 3.20, which caused uneven cell spreading and resulted in tempered analysis. Generally, the petri dish is divided into several sections. To address the issue, the center (section 5) of the petri dish was excluded from the analysis so that it can be better evaluated. As a result, the images are collected from eight different sections and their confluency values were averaged to estimate the overall confluency within the petri dish.

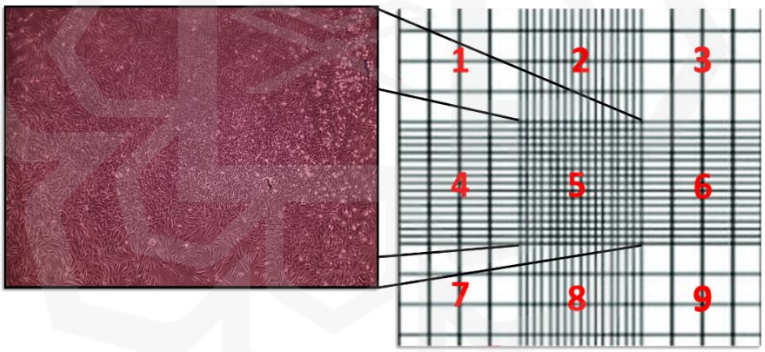


Figure 3.20 Uneven Cell Spreading

3.7 CHAPTER SUMMARY

This chapter introduced experimental setup, data acquisition and annotation, dataset preprocessing, segmentation modelling and their validations.

Experimental setup involves four aspects: a base model, training environment, metrics, and losses. A reduced version of U-Net architecture serves as a benchmark for subsequent model comparisons. Models undergo on-device training through five k-fold validations, The best one among the models will mainly be evaluated based on IoU scores and F-score elements, quantified by a hybrid loss of dice and focal losses during training.

Four preprocessing methods are proposed, including image correction using CLAHE and TVD, image patching, and patch filtering, to enhance discriminative features, reduce computational demands, and remove unnecessary bias on training data.

The model undergoes three-stage refinements, including inception, dilation, and residual, upon the baseline architecture. The refinement mechanisms were chosen based on theoretical knowledge in accordance with the case study of segmenting varying magnification levels.

For deployment purposes, the models will be validated based on individual observation and their background, and a lab experiment focused on studying the cell growth cycle. This chapter sets the stage before evaluating model performance and applicability. The next chapter will discuss the deployment of the models and explore the extent to which they can be integrated into the specific domain of cell culture monitoring.

CHAPTER 4

DESIGN OF AUTOMATED SYSTEM FOR CELL CULTURE MONITORING

4.1 GENERAL OVERVIEW

This study is centered around creating optimal pipelines for the segmentation of fibroblast cells. While the primary focus is on developing these pipelines, the ultimate goal is to integrate this system seamlessly into a real-world working environment. The project proposed a conceptual design, that incorporates three main systems: cell segmentation, remote monitoring, and sample handling, to create a robust and efficient system that can handle the cell culture of fibroblast growth using AIoT. The block diagram of this deployment flow strategy is visualized in Figure 4.1. The cell culture process begins with preparation and storage by assigned individuals. Subsequently, the petri dish is moved to the microscope, capturing an image sent to the Raspberry Pi for segmentation and confluency prediction using the pre-developed models. The results are shared with the online database and the user's mobile app. Based on confluency conditions, the user can monitor or authorize actions, and the robotic arm may autonomously continue segmentation or proceed to passaging as programmed.

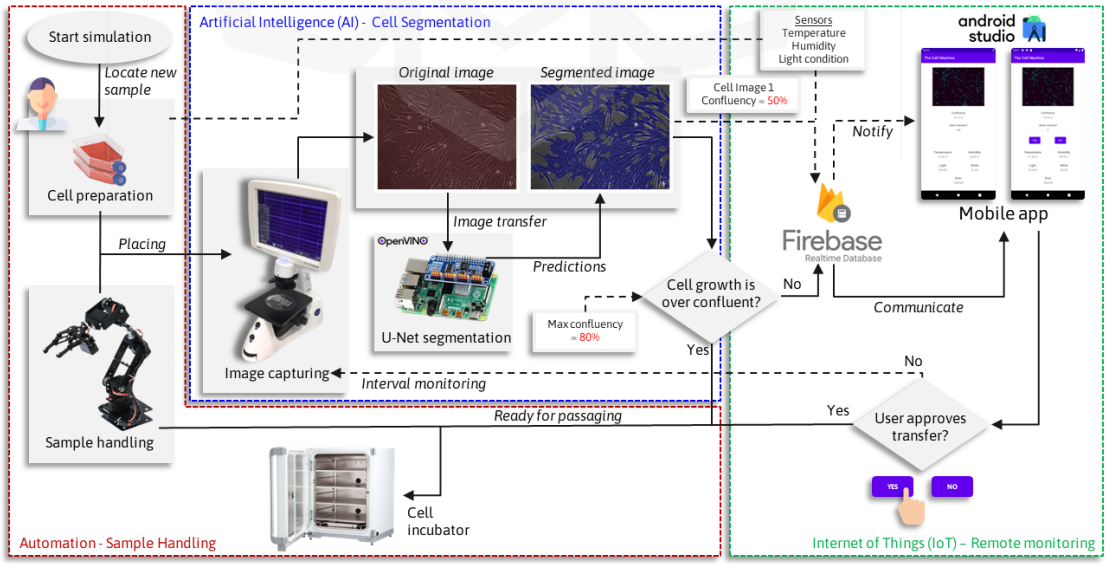


Figure 4.1 System Block Diagram

4.2 HARDWARE SETUP FOR DEPLOYMENT

The hardware setup of the system is illustrated in Figure 4.2, which shows the integration of segmentation, monitoring, and automation system. The entire setup requires a minimum space of (0.7×0.8) meters. For a comprehensive demonstration, conventional equipment commonly found in cell culture labs are included: a digital microscope, a DIY cell incubator comprising a three-storey display box and sample petri dish. The segmentation and monitoring system incorporates components such as a Raspberry Pi, an external vision processing unit (VPU), and an integrated circuit of sensors. The automation system consists of a robotic arm, a servo driver, and a power supply.

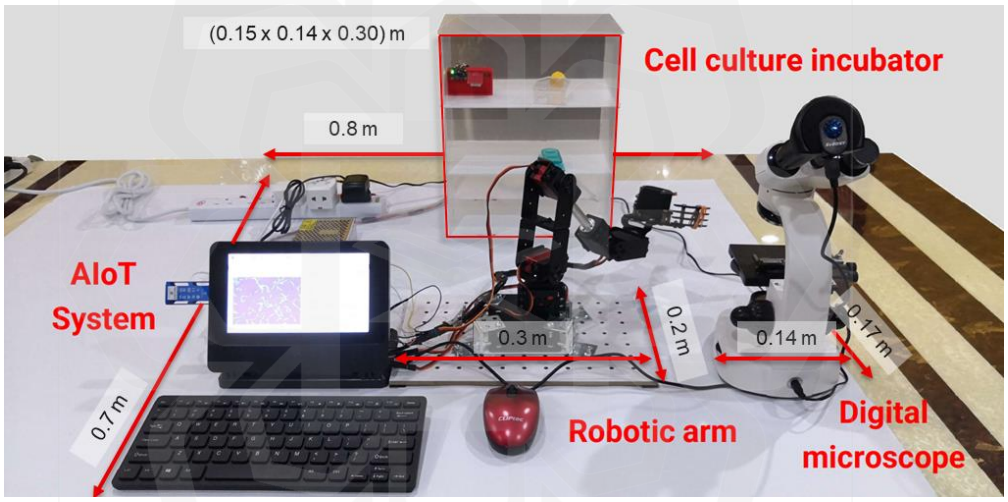


Figure 4.2 Hardware Setup for Deployment

The system uses Raspberry Pi 4 Model B with an Intel VPU USB which communicates with 16-channel 12-bit PWM Servo Driver HAT PCA9685 to control 6 servo motors acting as the joints of the robotic arm. As stated in the literature review, while the Raspberry Pi is a widely used and versatile board in this specific domain, it possesses lower computational power for deep learning models compared to other development boards. This deliberate choice is made to showcase the adaptability of our models even in challenging conditions with the lowest potential development board. To ensure that the system is able to integrate seamlessly and function effectively, a robotic arm, mostly chosen by existing studies in this field, was used. However, the selected robotic arm is an off-the-shelf with six 180° servo motors for

this study. The Servo Driver Hat is powered by the supported by an 6V-10A AC-DC switching power supply. The Raspberry Pi is also connected to VCNL4040 and DHT22 to report environmental conditions. All collected data, including segmented images and sensor outputs, are uploaded to Firebase and are accessible over Internet through developed mobile app in real-time. The detailed connection of the hardware and software system is illustrated in Figure 4.3. It is important to emphasize that this decision and onwards on the hardware and software selection was made for the purposes of merely realizing the design concept and demonstrating the feasibility of the system.

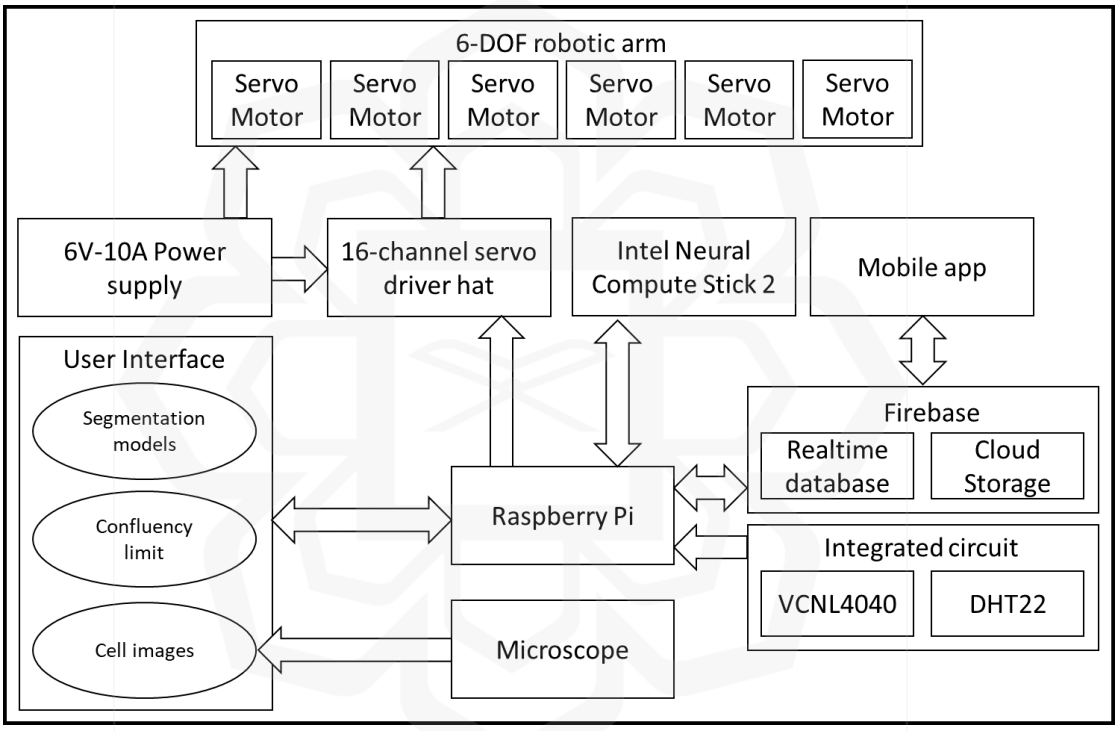


Figure 4.3 Hardware and Software Connection

4.3 AI-BASED IMAGING SYSTEM

Following the development of cell segmentation models, they will transition to a separate framework, distinct from the training framework, for better optimization and faster segmentation within the AI-based imaging system on the Raspberry Pi. Thus, this section will discuss the conversion methodology for implementing the models, alongside designing a user-friendly interface for the cell segmentation process and other relevant interactions within the entire system.

4.3.1 Tensorflow-OpenVINO Framework Conversion

In the preceding main section, several segmentation models have been proposed and assessed for their performance on the specified hardware. However, as mentioned earlier, they would be deployed on Raspberry Pi. The Raspberry Pi is solely utilized for inferencing, and it is not capable of training the model. To accelerate the performance of the image segmentation results on the Raspberry Pi, OpenVINO framework is utilized during deployment. Otherwise, all segmentation tasks can be slow even small models were used.

However, the use of the OpenVINO requires the segmentation to be applied on an Intel processor, given that the Raspberry Pi does not have one. This is the reason the Intel Neural Compute Stick (NCS2) has been incorporated. OpenVINO framework also necessitates the model to previously be trained in the Tensorflow framework and converted into a specific format, photobuf (.pb) format. The photobuf format is commonly used to bring the model into production for inferencing on other frameworks or over the web. The photobuf format is subsequently converted to Intermediate Representation (IR) for the OpenVINO framework. This conversion process is optimized to ensure efficient execution on the NCS2. The conversion steps are depicted in Figure 4.4.

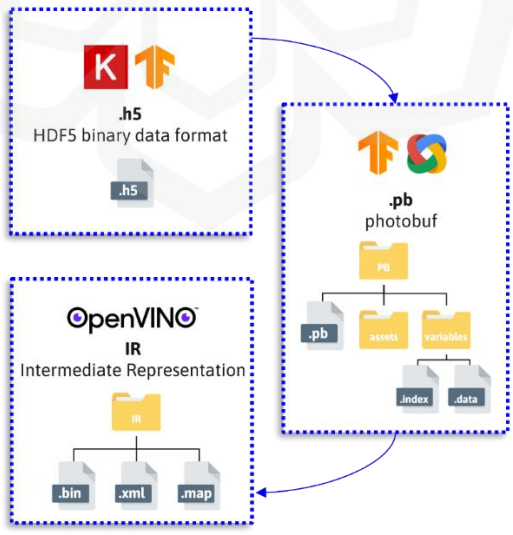


Figure 4.4 Model Format Conversion for OpenVINO Framework

4.3.2 Raspberry Pi Application Interface

To make the segmentation model accessible to non-experts, a simple graphical user interface (GUI) was developed using the PyQt5 library. As demonstrated in Figure 4.5, The user interface was designed to be easy to use and intuitive, with four main grids: an image browser, an image visualizer, a confluency indicator, and sensor outputs. For a conceptual design, the image browser allows the user to select and load the desired image or images for analysis. The image visualizer displays the selected image along with the segmentation results, enabling the user to visualize and evaluate the segmentation quality. The confluency indicator provides a numerical value that indicates the percentage of cell confluency in the selected image and allows a maximum confluency to be specified for the sample replacement. The sensor outputs provide real-time information on temperature, humidity, light, white, and proximity.

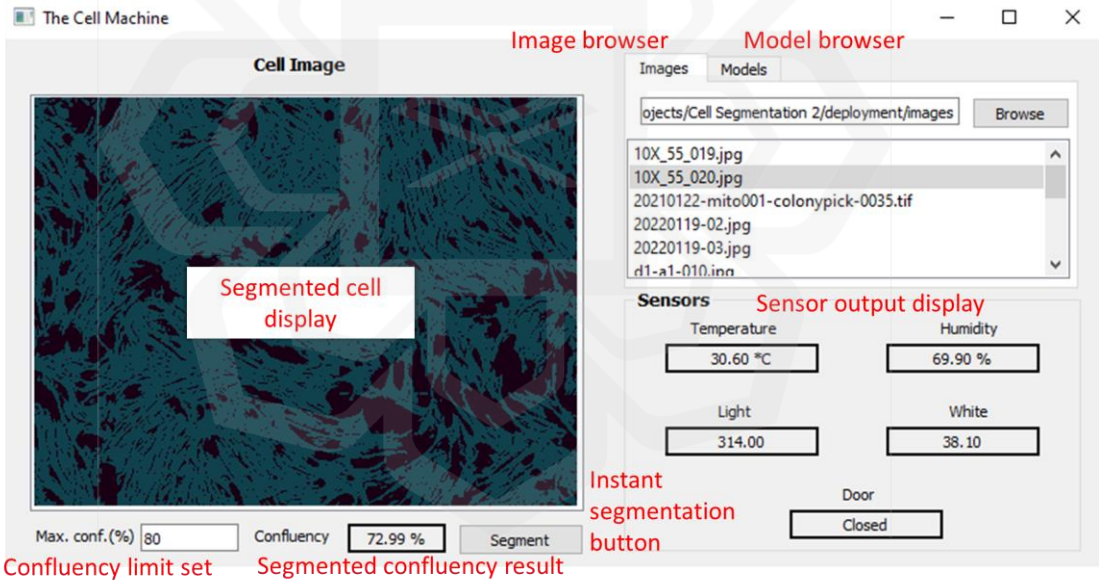


Figure 4.5 Illustration of System GUI

4.4 IOT-BASED CELL CULTURE MONITORING

This section encompasses the circuit design of relevant sensors for cell culture monitoring and the detailed framework for remote monitoring. This includes insights into the components of the online database and mobile application, illustrating how they facilitate the communication of output from the imaging system and the circuit.

4.4.1 Sensors for Cell Culture Incubator

Five parameters are considered important to monitor cell culture conditions within an incubator. It is essential to maintain optimal conditions within the incubator for the cells to grow and proliferate. To periodically measure these parameters, two sensors have been included and their detailed information is shown in Table 4.1. The first sensor, DHT22, is mainly to report environmental or climate conditions while the second sensor, VCNL4040, is to measure light intensity and act as limit switch for the door opening. It is important to emphasize that these sensors are specifically incorporated to detect significant deviations from optimal conditions. Given that most existing incubators come equipped with displays providing relevant information, the sensors can be validated in a known environment by placing them inside and observing the measurements as a reference.

Table 4.1 Sensors for Cell Condition Monitoring

No.	Sensors	Measured parameters	Specifications
1	DHT22 Sensor Module Breakout	• Temperature	• Measurement range: -40°C to 80°C • Accuracy: ±0.5°C • Resolution: 0.1°C
		• Humidity	• Relative measurement range: 0% to 100% • Accuracy: ±2% • Resolution: 0.1%
		• Operating voltage: 3.3 V to 5.5 V • Size: 15.5 mm x 12 mm x 5.5 mm	
2	Adafruit VCNL4040 Proximity and Lux Sensor - STEMMA QT / Qwiic	Two different light wavelengths: • Orange • White	• Measurement range: 0.0036 lux to 10,000 lux
		• Proximity (act as door exposure)	• Measurement range: Up to 200 mm
		• Operating voltage: 2.5 V to 3.6 V • Size: 3 mm x 3 mm x 0.75 mm	

As shown in Figure 4.6, the selected sensors were integrated onto a single donutboard circuit, which allowed for systematized data collection. To make the system more convenient, basic housing, as demonstrated in Figure 4.7 and Figure 4.8, was designed to facilitate easy installation and removal from the incubator, allowing for regular calibration and maintenance. The housing was designed to be compact, so as not to take up too much space inside the incubator for the cell culture and sample handling.

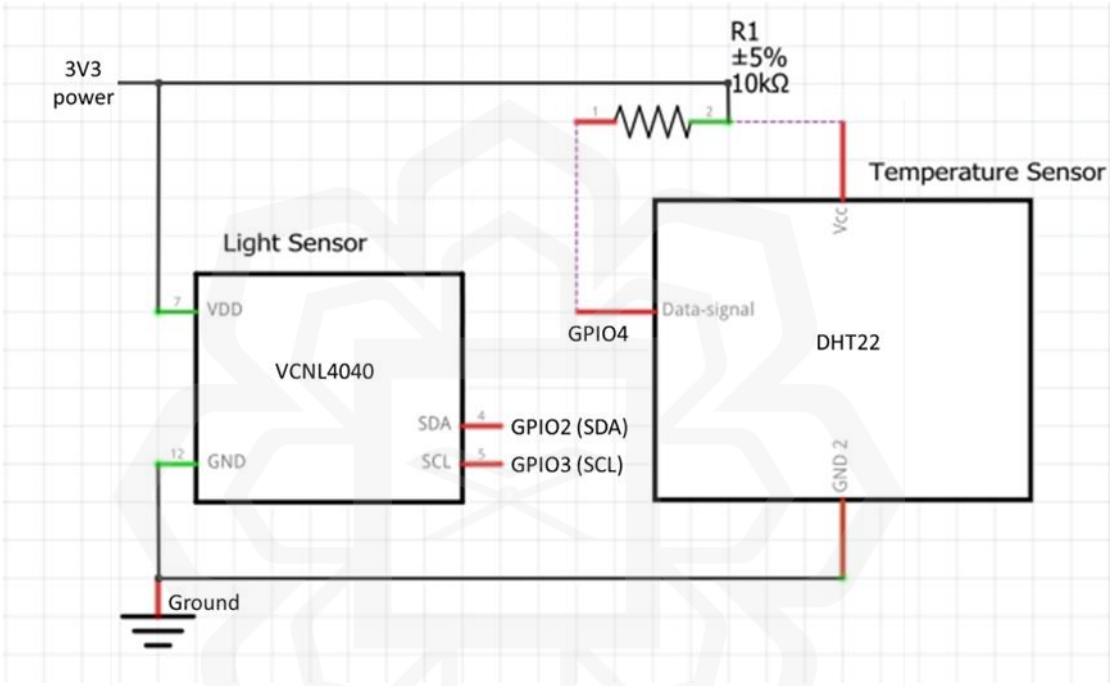


Figure 4.6 Circuit Schematic of the Integrated Sensors

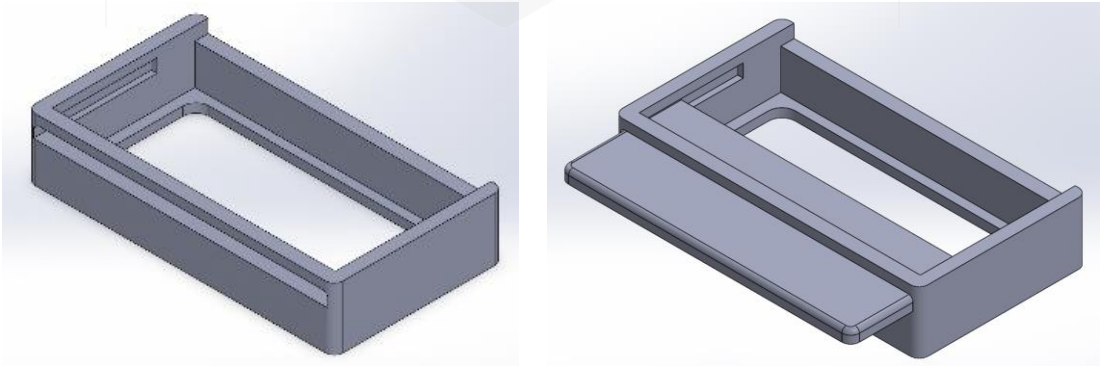


Figure 4.7 Prototype Case Design of the Circuit

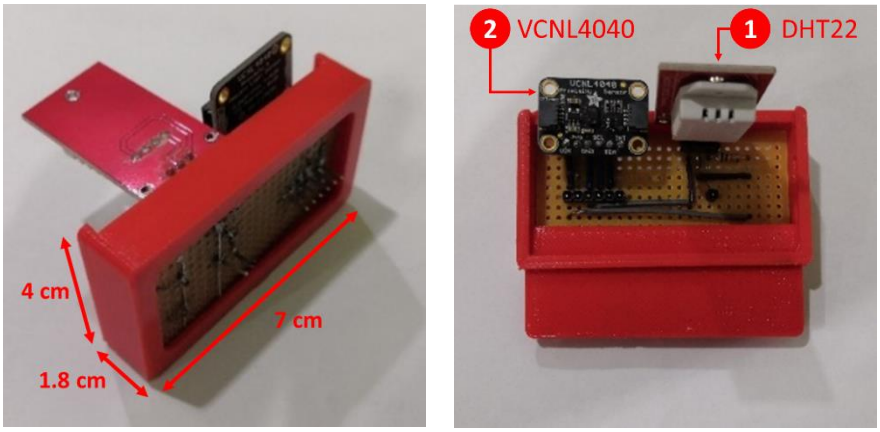


Figure 4.8 Final Production of Cell Incubator Sensors

4.4.2 Remote Monitoring Using Firebase Database and Android Studio

To enable remote monitoring of the cell culture system, all the sensor data and segmentation results were sent to Firebase, ensuring that the data logs are securely stored and easily accessible to authorized users from other IoT-based mediums such as mobile devices. This system serves four main purposes, including observing and notifying periodic segmented cells and their confluency values, tracking the system mode, authorizing cell transfers, and monitoring real-time cell culture conditions.

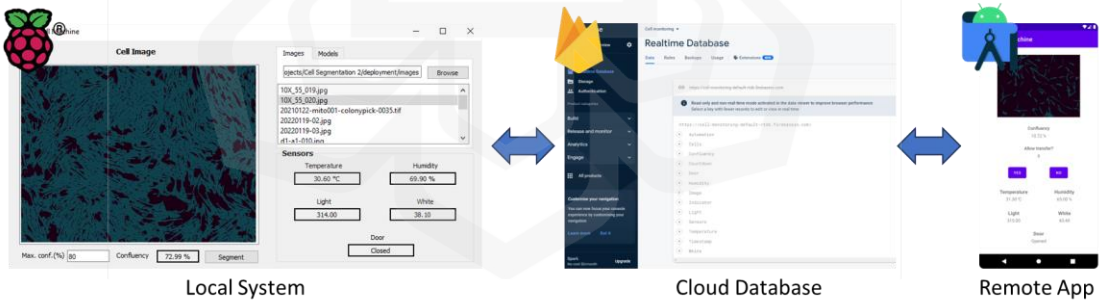
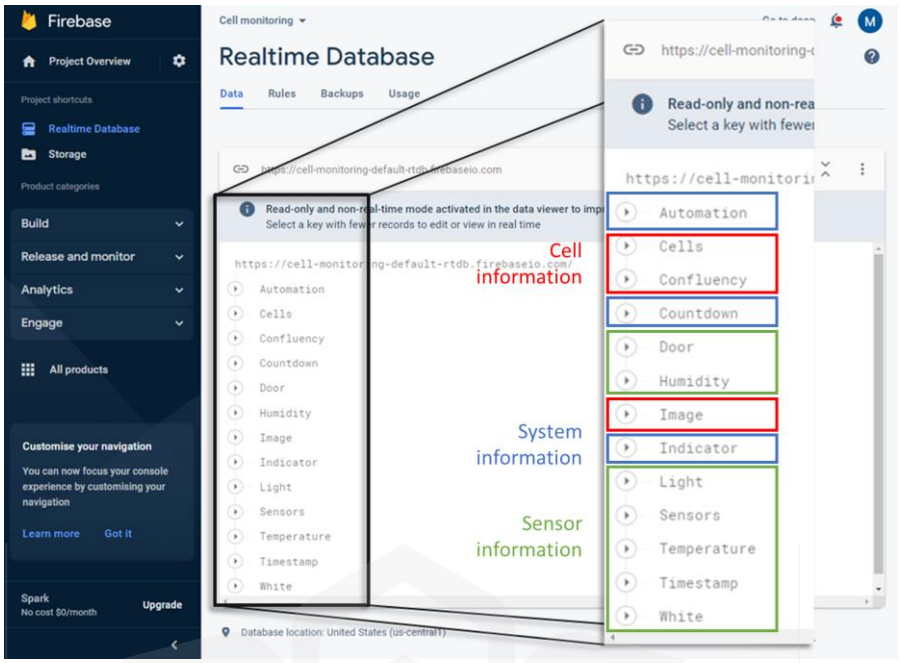


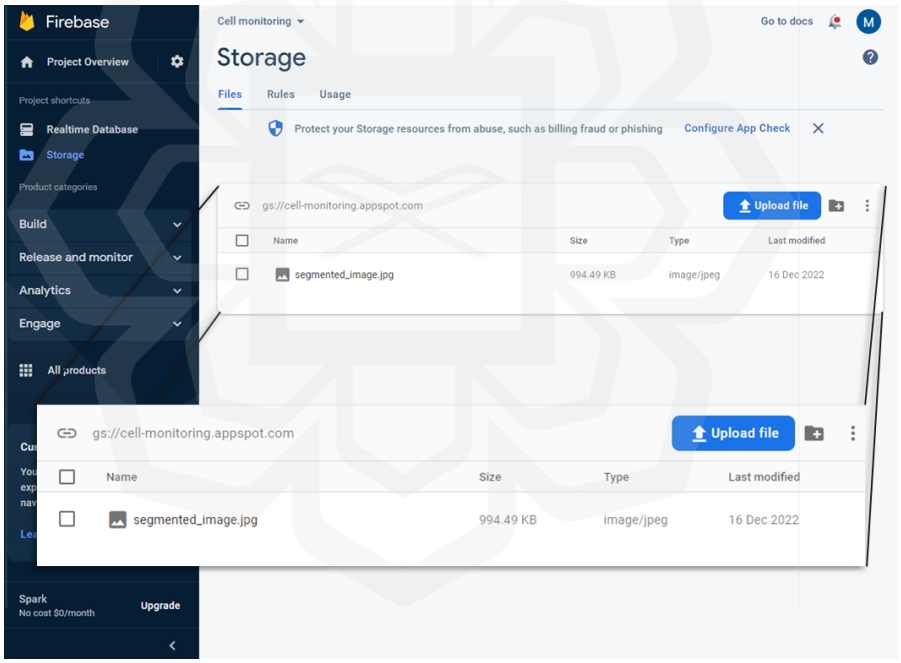
Figure 4.9 Remote Monitoring Framework

A. Firebase Console

Two Firebase services were used where Firebase Storage is used to store the predicted cell image and Firebase Realtime Database to store three types of information which cell information, system information, and sensor information, as depicted in Figure 4.10.



(a) Firebase Realtime Database



(b) Firebase Storage

Figure 4.10 Cloud Firebase Interfaces

As mentioned earlier, there are multiple types of data that need to be transmitted from the cell culture system to Firebase, but they do not need to be sent at the same periodic rate. To optimize data transmission, these data were transmitted to Firebase in compilations based on the system mode label, as detailed in Table 4.2.

Table 4.2 Data Format Sent to Cloud Firebase

System Mode Label		Parent	Child
Idle			
Segment_1		AUTOMATION	• Countdown • System Mode Indicator
		CELLS	• Segmented image • Confluency value
Handling	**_1**	AUTOMATION for continuing imaging	
	_2	AUTOMATION for replacing sample	
Sensor_1		SENSORS	• Temperature • Humidity • Light • White • Door exposure

The SENSORS data, which includes information from the temperature, humidity, and light sensors, were sent in real-time, as changes in these parameters can have an immediate impact on the cell culture environment. The AUTOMATION data, which includes information on system mode and the countdown for the user to authorize the sample replacement, were sent before and after a cell image was segmented. This allows users to track changes in the system mode and ensure that the system is operating as expected before and after segmentation. Moreover, the countdown feature ensures that the user has enough time to review the cell culture data shared by the CELLS data and decide about whether to replace the sample or continue with the current sample. The workflow of system mode will be further elaborated in a later section.

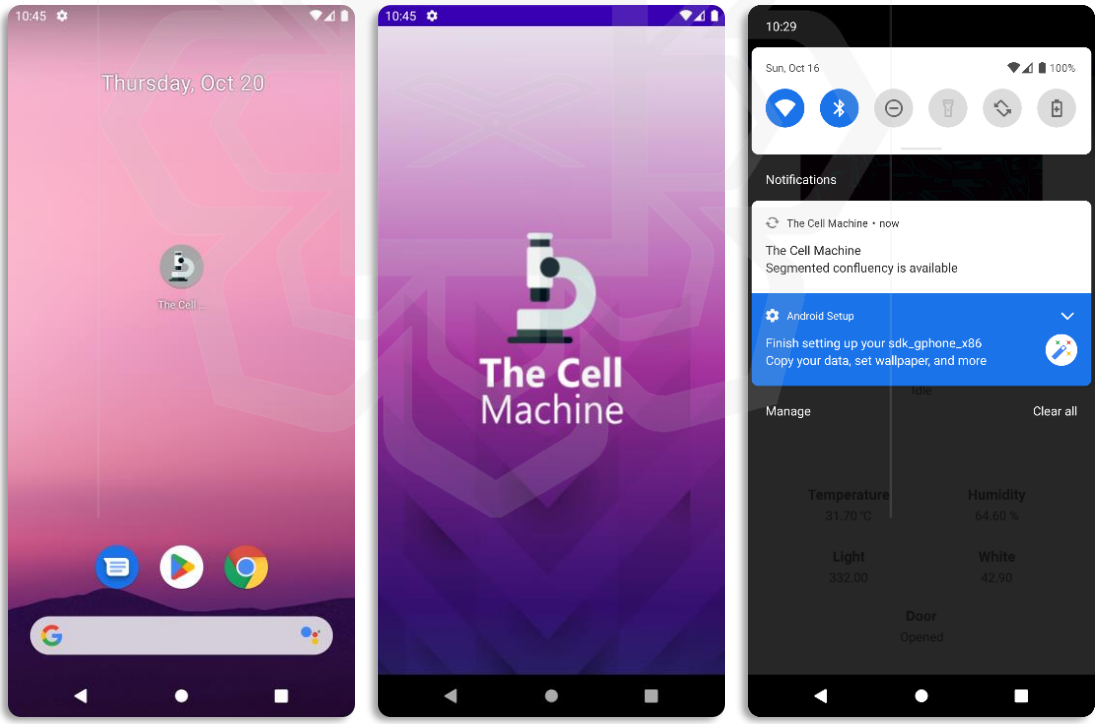
B. Android Application Interface

The remote monitoring system enables researchers to monitor and control their cell culture experiments from anywhere, at any time, using a mobile device. This provides greater flexibility and convenience, enabling users to respond to any issues or changes in real-time. All the information collected is communicated to the mobile app via Firebase. For this study, the mobile app was developed using Android Studio, a well-known platform for only creating Android applications. The detailed features provided by the remote app are described in Table 4.3. To visually represent the branding

elements and primary feature interfaces of the mobile app, refer to Figure 4.11 and Figure 4.12, respectively.

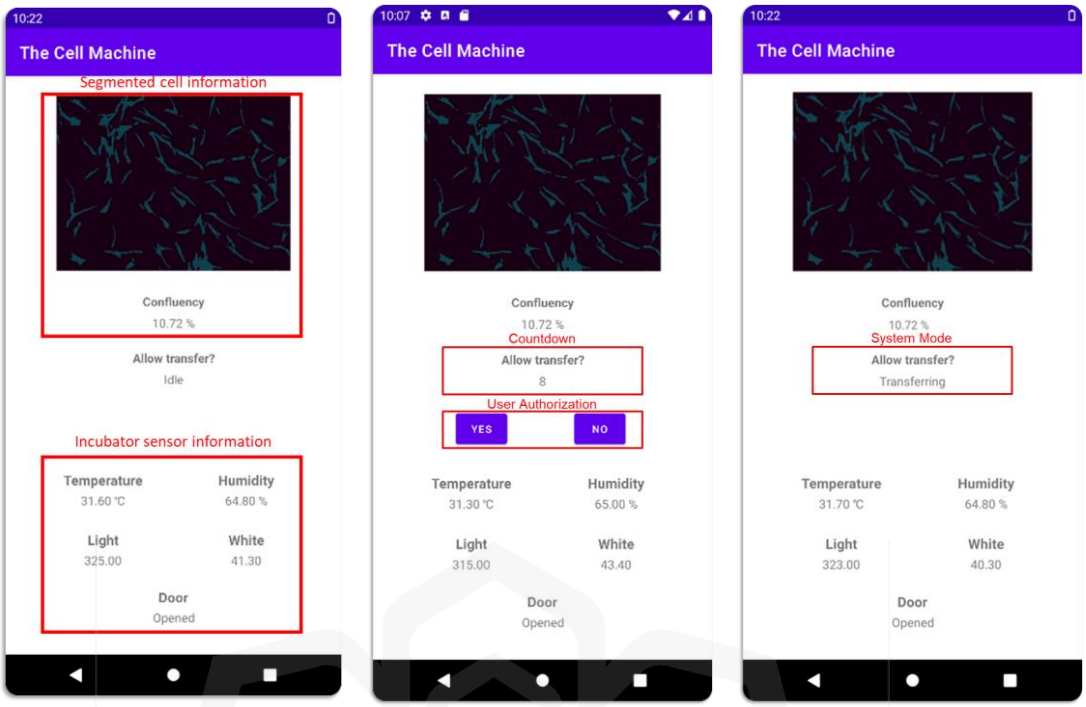
Table 4.3 List of Remote App Features

Features		**Descriptions**
Information Display	Cell Information	Display a recent segmented cell image and confluency
	System Mode	Display current system mode and manage countdown during user authorization phase
	Sensor	Display real-time incubator sensor outputs
User Authorization Button		Enable user to force cell transfer when the confluency is still below limit. Only appear when certain conditions met
Notification		Notify user whenever segmented cell information is transmitted



(a) Application Icon (b) Splash Screen (c) Notification

Figure 4.11 Mobile App Branding Elements



(a) Default Page (b) User Authorization (c) Sample Transferring

Figure 4.12 Mobile App Main Page

The design of the mobile app's interface is intentionally streamlined to provide users with a single-page view that concisely displays all the relevant information. The segmented image, system mode, countdown, and real-time sensor data are all displayed on this single page, providing users with a comprehensive view of the system status at a glance. In addition, the mobile app includes a notification system that alerts users when segmented confluency values are available for review and when they need to authorize the next procedure.

C. Detailed Workflow of System Mode

To ensure smooth communication and control over the system, the system mode is assigned indicator numbers ranging from 1 to 6 in the script. Furthermore, transmission of cell information mainly primarily revolves around two conditions, labelled as '1' and '2'. When indicator '1' is triggered, it indicates that the cell confluency has reached the limit and will transition to indicator '3'. At this stage, the cells will be directly transferred for passaging after the cells are segmented. On the contrary, indicator ‘2’ shows that the confluency has not reached the limit and require

further judgment. If the confluency less than half of the confluency limit, then, it will be directly stored back into the incubator. If the segmented confluency exceeds the half of the confluency limit, the further action will depend on the input from the authorized users remotely through mobile app, labelled as “special case” to determine next indicator number, as depicted in Figure 4.13.

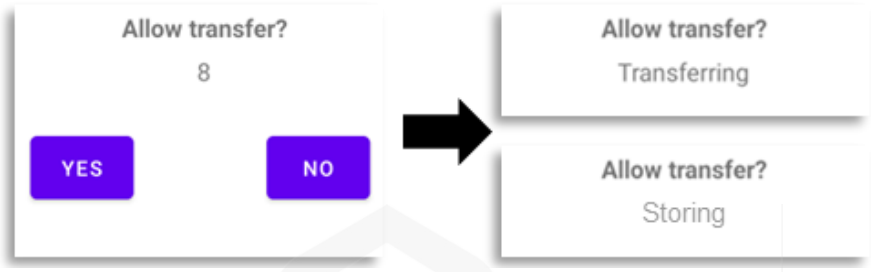


Figure 4.13 Waiting Period for User Authorization Interface

If there is no response from the user interface, the system will interpret the first event as an indication to continue storing the cells. The transition conditions of indicator numbers for cell information are summarized as follows:

$$Main\ Indications:$$
$$Indicator \begin{cases} 3 & if\ indicator == 1 \\ 5 & if\ indicator == 2\ and\ confluency < 0.5 * limit \end{cases}$$

$$Special\ Case:$$
$$Countdown \{ 1 - 10 \quad if\ indicator == 2\ and\ confluency > 0.5 * limit$$

$$User\ Authorization:$$
$$Indicator \begin{cases} 3 & if\ input == "YES"\ or\ input == " " \\ 5 & if\ input == "NO" \end{cases}$$

On the other hand, indicator numbers 3 to 6, denoted as Transferring, Idle, Storing, and Segmenting, primarily represent different aspects of system automation. Each indicator number corresponds to a specific system mode. The following is an elaboration of the system modes associated with each indicator number:

Table 4.4 Descriptions of Indicator Number for Automation System

Indicator Number	Descriptions
3	The system enters the "*Transferring*" mode, indicating that cell transfer is in progress. This mode is activated when the cell confluency reaches the limit, and the cells are being prepared for passaging after segmentation.
4	The system transitions into the "*Idle*" mode. This mode indicates that the system is not actively performing any specific actions or operations at the moment. It serves as a pause or resting state for the automation system. Real-time monitoring of cell incubator still active.
5	The system enters the "*Storing*" mode. This mode implies that the cells are being stored back into the incubator. It occurs when the cell confluency is considered low.
6	The system switches to the "*Segmenting*" mode. This mode is activated depending on user input or predefined system protocols. It transfers the cells from the incubator to the microscope to begin segmenting.

4.5 SAMPLE HANDLING WORKFLOW

After discussing the system mode indicators in the previous section, it is clear that the robotic arm primarily operates in three locations: the incubator, microscope, and an idling state. This section highlights how the robotic arm transitions between these locations, as illustrated in Figure 4.14.

Figure 4.14 Illustration of Arm Motion for Sample Handling

The off-the-shelf robotic arm consists of six separate servo motors. Typically, a Servo Motor HAT is employed to control these motors, although not necessarily all at once. In order to navigate around surrounding objects like lab equipment and furniture, particularly within confined spaces, the robotic arm requires a seamless motion trajectory. The initial program implements a sequential approach for servo motor control, but this results in increased trajectory time when moving between destinations. Moreover, the robotic arm requires to transfer cells within a petri dish without tilting it. Therefore, there is a need to implement a multithreading program that enables the servo motors to communicate with each other, enhancing coordination and synchronization and leading to improved trajectory efficiency. For the demonstration, the implementation primarily on servo 1 to servo 4, as these are the key motors involved in the robotic arm's motion.

4.6 CHAPTER SUMMARY

This chapter proposed a design of an AIoT based automated cell culture monitoring, integrating cell segmentation, remote monitoring, and sample handling systems. The hardware setup primarily leverages Raspberry Pi, Intel VPU, environmental sensors, and an off-the-shelf robotic arm.

The AI-based imaging system adopts OpenVINO framework, enhancing Raspberry Pi efficiency, as low power application, for deep learning models. The user-friendly interface, developed using PyQt5, enhances accessibility for non-experts during the cell segmentation process.

IoT-based cell culture monitoring includes circuit design for DHT22 and VCNL4040 sensors, along with a detailed framework for remote monitoring through Firebase and Android Studio. Firebase facilitates real-time data transmission, and the mobile app provides a single-page view of critical information.

The study establishes six system modes to seamlessly interact between cell imaging, remote monitoring, and sample automation. Additionally, the system uses multiple threads to control the different joints of the arm concurrently, ensuring safe sample transfer and preventing collision with other lab components. This chapter establishes the groundwork for subsequent system workflow assessment within a restricted space.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 GENERAL OVERVIEW

This chapter presents the results and discussion derived from the methodology proposed in Chapters 3 and 4. It provides a comprehensive analysis of the entire pipeline, encompassing the performance of preprocessing methods and segmentation model designs throughout the training and inference processes. Specifically on training process, the performance of each model will further be assessed in relation to complexity and loss. Additionally, given there are three distinct datasets, the study will also explore the comparative performance of training models individually and collectively.

Apart from the detailed assessment on the segmentation models, the deployment of segmentation models on Raspberry Pi, serving as a user-friendly imaging system, will also be subjected to performance evaluation. Next, the study observes the efficiency of the proposed remote monitoring system, examining communication delay time and data transmission rate. Similarly, the proposed sample handling system is assessed for workflow completion time and power consumption along the workflow.

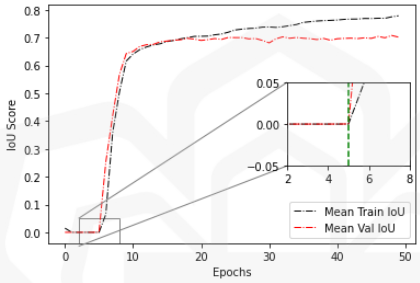
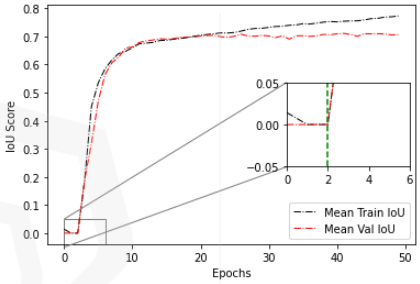
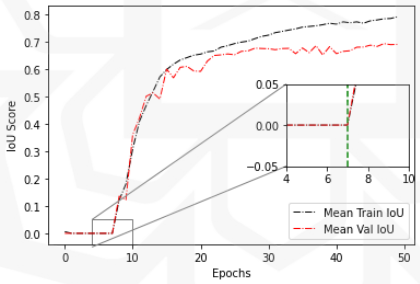
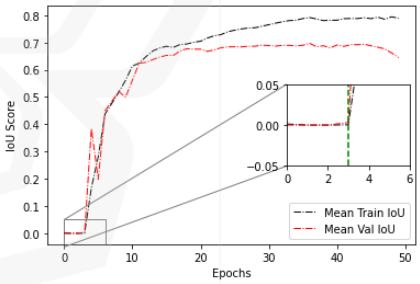
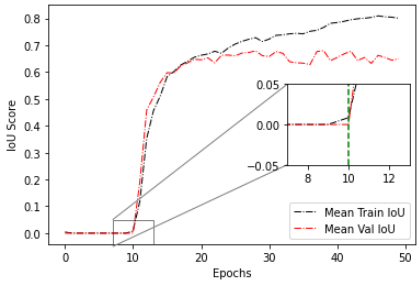
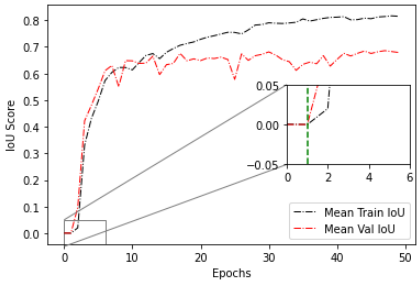
5.2 ANALYSIS OF PROPOSED IMAGE PREPROCESSING METHODS

This section highlights the effectiveness of proposed preprocessing methods on the collected data before training and identifies the conditions under which these methods perform optimally. This step should enhance the discriminative features of the object of interest within the image and concurrently reduce the computational demands of learning sample batches.

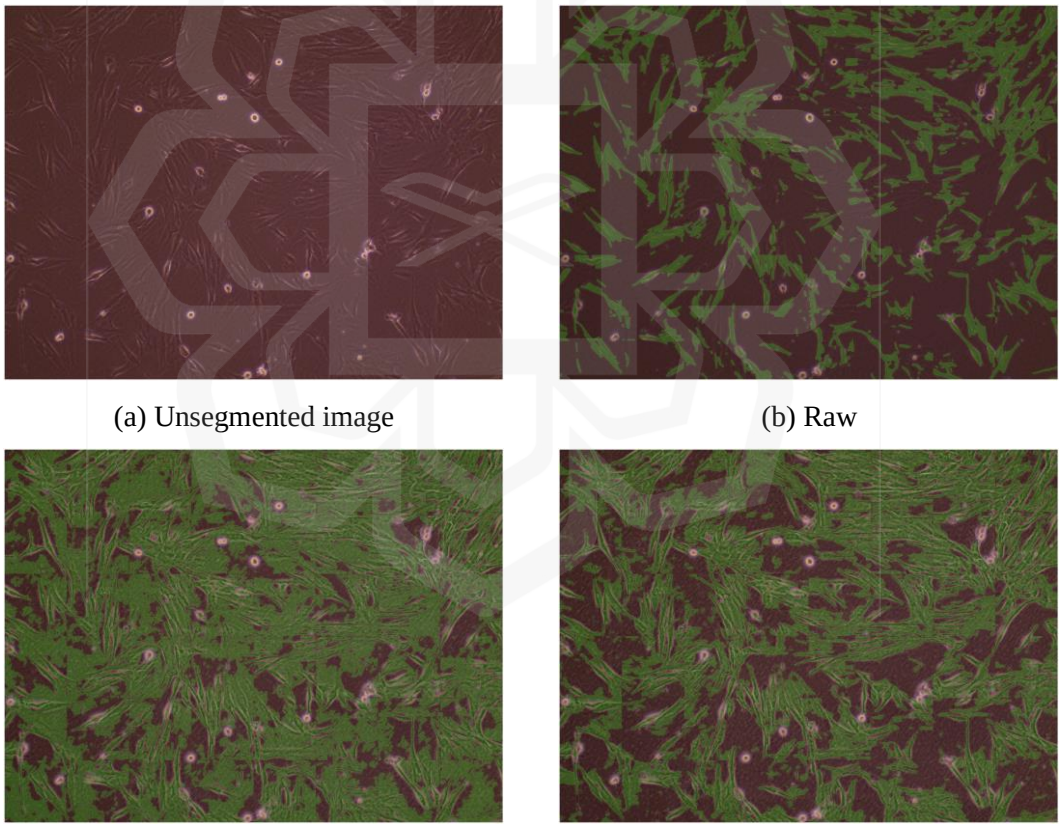
5.2.1 Enhancing Discriminative Features

Table 5.1 shows the effect of CLAHE on the training graphs for three datasets of different magnification levels, including the IoU score differences before and after applied.

Table 5.1 Analysis of CLAHE Implementation on Training Datasets

Magnification levels	**CLAHE**	
	Not applied	**Applied**
x4	 IoU = 0.708	 IoU = 0.711
	↑ 0.42 %	
x10	 IoU = 0.694	 IoU = 0.697
	↑ 0.43 %	
x20	 IoU = 0.679	 IoU = 0.685
	↑ 0.88 %	

According to the table above, for each dataset, the training graphs have better momentum when CLAHE is applied. The IoU score starts increasing earlier from zero as the epochs progress, allowing the model to converge faster and reduce training time. Specifically, the model can start learning nine epochs faster on the x20 dataset. Furthermore, the x20 dataset demonstrates a nearly 0.9% improvement in mean IoU score, while others show approximately half of that percentage. Considering both the learning graph and mean score, their improvement becomes more pronounced with higher magnification levels. Despite the variation, the method is worth applying universally, as it only executes within seconds in real-time. Apart from the statistical improvement, Figure 5.1 presents an example of segmentation result by baseline U-Net, where specific image processing settings are used.



(a) Unsegmented image (b) Raw

(b) CLAHE (c) CLAHE + TVD

Figure 5.1 The Implementation of TVD on High Noise CLAHE Image

While CLAHE effectively addressed the issue of under-segmentation in direct segmentation on the raw image, it introduced a new problem, over-segmentation, due to the high noise generated, particularly on background pixels. To address this, the TVD method is proposed in the methodology and applied to reduce the noise in the CLAHE image while preserving the edge features.

Based on the overall observations, it can be inferred that different image processing techniques are most effective for particular image conditions during segmentation, as shown in Table 5.2. These findings suggest that a tailored approach to image processing is necessary to achieve optimal results for segmentation tasks, rather than using a one-size-fits-all technique. Moreover, certain image conditions may require segmentation without the use of any image processing techniques at all.

Table 5.2 Image Conditions for Effective Image Processing Techniques

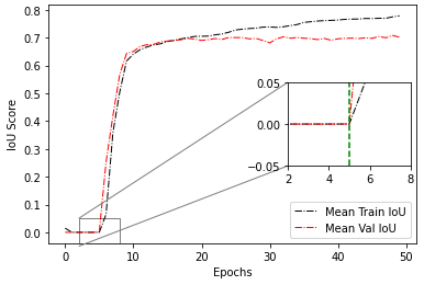
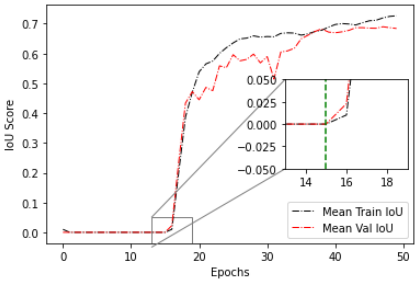
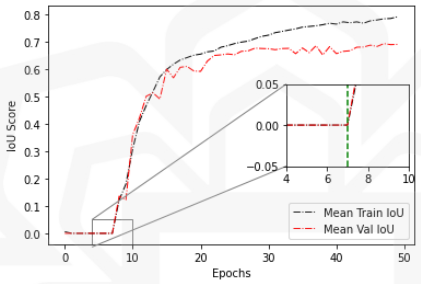
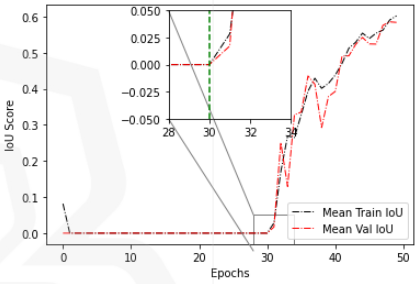
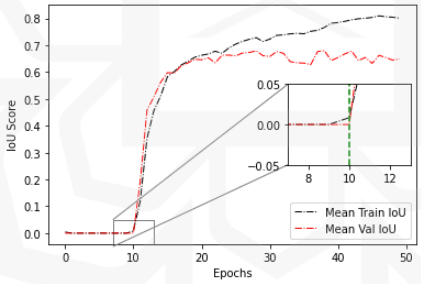
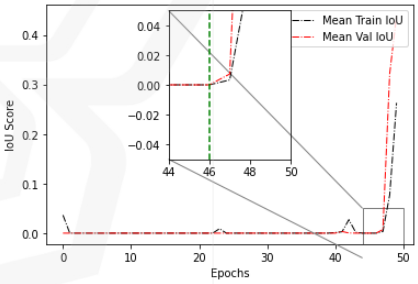
Image processing techniques	**Image conditions**
CLAHE	• Low magnification level images < x10 • High brightness images > 50
TVD	• High magnification level images ≥ x10 • High confluency images > 50 %

5.2.2 Reducing Computational Demands

Another method is image patching to cope with overflow computational demands on the available hardware for this study. As previously stated, the study considered two patch sizes, 256x256 and 516x516. Regardless of the chosen option, this step is deemed necessary unlike the previous preprocessing methods. Table 5.3 shows the effect of difference patch size applied during the training.

The results demonstrate that patches with dimension 256x256 have better ability to learn the dataset conditions through the training epochs and have higher mean IoU score than patches with dimension 516x516. Compared to previous methods, this has higher impact where the performance difference get nearly 50%. Therefore, further methodology will be conducted using the former patches, 256x256. Conclusively, same as previous methods, the performance becomes more pronounced with higher magnification levels.

Table 5.3 Analysis of Image Patching of Dimensions 256 and 512

Magnification levels	Patch dimensions	
	256x256	**512x512**
x4	 IoU = 0.708	 IoU = 0.693
x10	 IoU = 0.694	 IoU = 0.572
x20	 IoU = 0.679	 IoU = 0.436

5.3 ANALYSIS OF PROPOSED CELL SEGMENTATION MODELS

This section involves the analysis of model performance, examining their relationships with complexity and loss during individual and collective data training.

5.3.1 Model Complexity

Considering model complexity is crucial when the ultimate goal is deploying the model in a real-world setting. Regardless of high performance, it would be pointless if the model cannot be deployed into the available edge device such as Raspberry Pi. More complex models can have longer inference times, which can be problematic for real-time or latency-sensitive applications. There are several indicators that can be included in comparing the model complexity such as number of parameters, training time and inference time. Table 5.4 shows the model complexity comparisons of all models tested on the specified hardware.

Table 5.4 Model Complexity Comparisons

Model	**A**	**A_5**	**B**	**C**	**D**
	U-Net (baseline)	**U-Net (kernel 5)**	**Inception (kernel 3 & 5)**	**Dilation (rate =2)**	**+ Residual**
Phase	**0**		**1**	**2**	**3**
Number of layers	40	40	85	85	112
Number of trainable parameters	1,940,817	5,078,609	7,063,937	3,917,505	4,004,817
Minimum memory required (GB)	1.66	1.67	3.83	3.79	4.87
Model size (MB)	23	60	83	46	47
Mean training time (mins)	2.9	4.2	7.4	7.2	8
GPU inference time (ms/ patch)	40	52	56	54	58

Based on Table 5.4, it is apparent that the final proposed model, despite having fewer trainable parameters than the Inception model with various concurrent kernel sizes (Model B), has the highest minimum memory requirement. This may primarily be due to the fact that the final model has a greater number of layers, resulting in a higher memory demand for features. The minimum memory required is determined based on the features and parameters utilized in the model which can be defined as:

$$Minimum\ memory\ required\ (GB) = \frac{(BS\ \times f_m) + \mathrm{p}_m}{1024} \quad (19)$$

where BS is batch size (24), f_m is features' memory in megabytes (MB) and p_m parameters' memory in megabytes (MB).

It should be noted that the memory requirements presented in Table 5.4 are merely the minimum requirements necessary or initial precaution for running the models. In reality, training deep learning models involves several overheads, such as the amount of data and data complexity, that can significantly impact the memory requirements and overall computational cost.

5.3.2 Model Performance

To assess the performance of the segmentation model, three key aspects are evaluated: the learning adaptation over time is monitored by the training graph, the model's consistency is measured by evaluating the mean of its statistical score on test data, and areas where the model excels and where it needs improvement are identified by analyzing the precision of visually predicted segmentation. The first two are evaluated quantitatively while the last is observed qualitatively.

A. Quantitative Assessment

In Figure 5.2, the mean training graphs of the train and validation IoU score of the selected models are displayed, and it can be observed that the shaded region represents the deviation as five trials were conducted for each model and dataset with respect to the k-fold split. Moreover, the difference histogram of train and validation line graphs has been included to monitor the overfitting scenarios across the training epochs. The differences histogram was calculated throughout the end of epochs as follows:

$$Score\ differences = mean\ train\ iou - mean\ val\ iou \quad (20)$$

where overfitting is indicated by the increasing positive value, while underfitting is indicated by the increasing negative value, and a zero difference indicates the convergence of the training graphs.

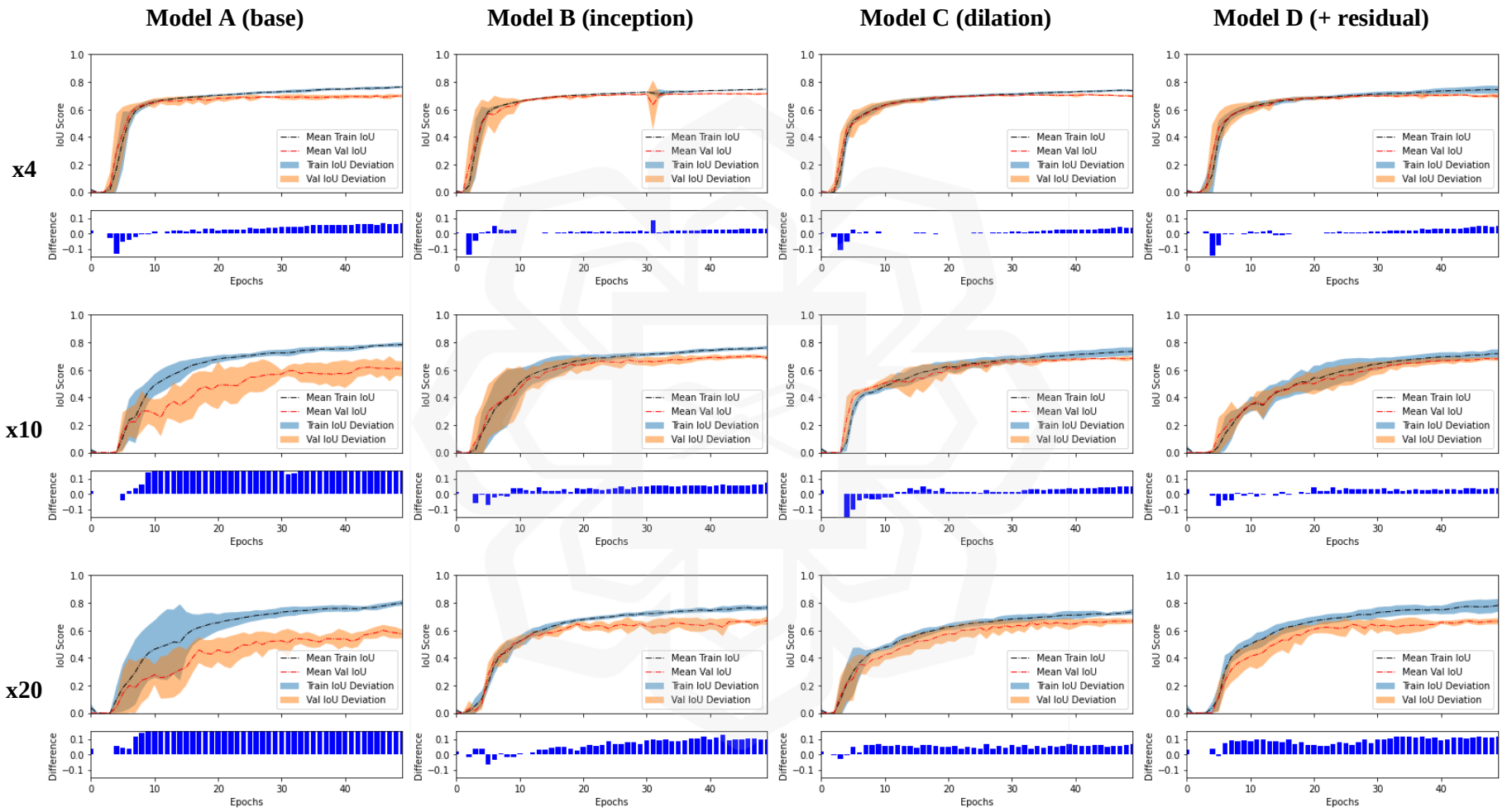


Figure 5.2 Training Graphs of The Selected Models

It also can be observed from Figure 5.2 that the lowest differences for every dataset throughout the epochs are indicated by Model C with the dilated path, followed by Model B, Model D, and lastly, the baseline Model A, which is suggested to have the highest degree of overfitting. The graphs also indicate that higher magnification levels result in training graphs that are harder to converge, with the x4 dataset being the easiest to train and the x20 dataset posing the most difficulty.

In terms of mean IoU scores, Table 5.5 and Table 5.6 present the scores and relative performance, respectively, assessed on the designated test data, as indicated in Figure 3.2. Model B with various concurrent kernel sizes yields the highest overall IoU score and relative performance towards the baseline Model A. As stated earlier, model C was introduced to reduce the complexity of Model D while retaining its larger receptive field. Then, Model C was extended to Model D by incorporating residual connections in order to preserve information on the skip pixels. However, contrary to expectations, statistically, Model C outperforms Model D despite the considerations.

Table 5.5 IoU Comparisons Between Models

Model type	**Intersection over Union (IoU)**		
	Training datasets based on magnification levels		
	x4	**x10**	**x20**
A	0.707±0.002	0.661±0.024	0.626±0.017
B	0.717±0.002	0.703±0.009	0.701±0.009
C	0.711±0.003	0.686±0.006	0.696±0.007
D	0.707±0.003	0.684±0.007	0.689±0.013

To ensure optimal IoU performance while minimizing memory usage, this study employed three phases of U-Net model refinement. As demonstrated in Table 5.6, Model C proved to be a promising alternative to Model B, which exhibited lower trainable parameters with only lags in terms of performance by a mere 1%. The correlation between the IoU scores of the models is computed using Equation (21) and is documented in Table 5.6.

$$Rel.performance\ (\%) = \frac{new\ model\ iou - previous\ model\ iou}{previous\ model\ iou} \times 100 \quad (21)$$

Table 5.6 Relative Performance Between Models

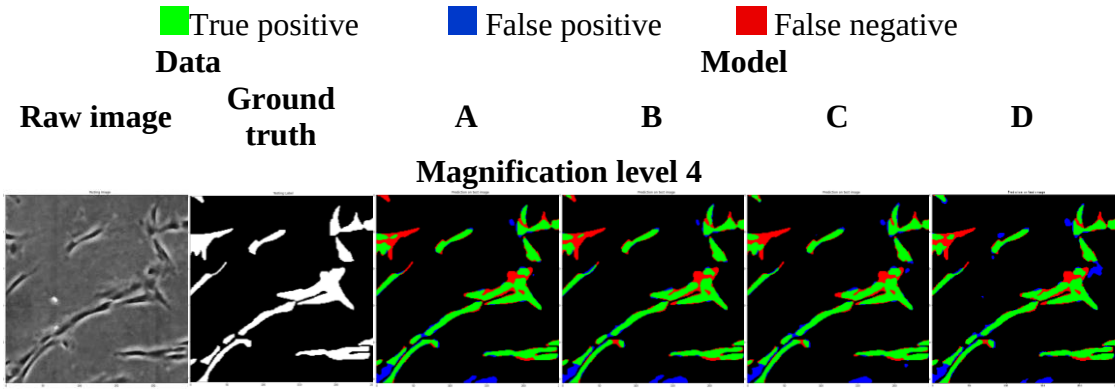
Relation	Relative performance (%)		
	Training datasets based on magnification levels		
	x4	x10	x20
B → A	+ 1.41 %	+ 6.35 %	+ 11.98 %
C → A	+ 0.57 %	+ 3.78 %	+ 11.18 %
C → B	- 0.84 %	- 0.22 %	- 0.71 %
D → A	0 %	+ 3.48 %	+ 10.06 %
D → B	- 1.39 %	- 2.70 %	- 1.71 %
D → C	- 0.56 %	- 0.29 %	- 1.00 %

In this subsection, the outlined results demonstrate that all refined models (B, C, and D) consistently enhance learning graphs and segmentation scores compared to the baseline model A. Moreover, based on the table above, a significant improvement in mean IoU score, exceeding 10%, is observed on the most complex dataset in this study, which consists of x20 images. However, a unique case is identified with model D, where it falls short of surpassing the performance of model A only on the x4 dataset, maintaining a mean IoU score of 0.707.

B. Qualitative Assessment

Previously, the performance metric for evaluating the segmentation models mostly involved the IoU score, which measures the overlap between the predicted pixels and the ground truth labels. In this study, another metrics that are also considered are precision and recall.

Based on the observations in Figure 5.3, it is found that the proposed models have varying combinations of IoU, precision and recall score, indicating that different strengths and weaknesses exist for each proposed model in terms of segmenting object of interests. It is worth noting that lower precision could lead to over-segmentation, where the model predicts more positive pixels than the actual ground truth while lower recall can indicate under-segmentation, where the model misses some of the actual positive pixels.



Average precision	81.90	81.54	80.30	81.14
Average recall	86.00	87.34	87.41	86.60

Magnification level 10

Average precision	79.50	79.63	79.51	79.28
Average recall	83.83	86.18	84.58	84.35

Magnification level 20

Average precision	74.88	74.95	73.60	74.68
Average recall	79.01	81.96	85.19	85.67

Figure 5.3 Average Precision of Test Patch Segmentation

C. Assessment Summary

Table 5.7 presents the rankings of the four different models in terms of IoU, precision and recall towards specific datasets, indicating the possibilities of three segmentation errors on cell predictions which are poor localization, over-segmentation, and under-segmentation, respectively. Based on the table, it is clear that Model B outperforms the other models across all datasets, achieving the highest overall performance in terms of IoU, precision and recall regardless of its complexity. A balanced performance of the individual model is indicated by the same ranking number obtained by the model under specific dataset, which those are shaded in table. However, a balanced performance of a model does not justify that it has the highest performance among other models unless it scores “1” for each of those metrics, as

shown by Model B on Dataset x10. Despite baseline Model A having relatively good precision on each dataset, it exhibits the poorest IoU and recall values among the models, indicating a higher possibility of poor localization and under-segmentation in its predictions. This proved that the refinements of baseline Model A in this study are relevant for an increased performance of other metrics.

Table 5.7 Overall Performance of Segmentation Models

Order	Model	x4			x10			x20			Best dataset for the model
		IoU	Pre.	Rec.	IoU	Pre.	Rec.	IoU	Pre.	Rec.	
↓	**A**	4	1	4	4	3	4	4	2	4	x4
	B	1	2	2	1	1	1	1	1	3	x10
	C	2	4	1	2	2	2	2	4	2	x10
	D	3	3	3	3	4	3	3	3	1	x20
Best model for the dataset		B			B			B			

where “1” refers to the lowest possibility of the errors and “4” refers to the highest.

5.3.3 Relationship Between Model Performance and Complexity

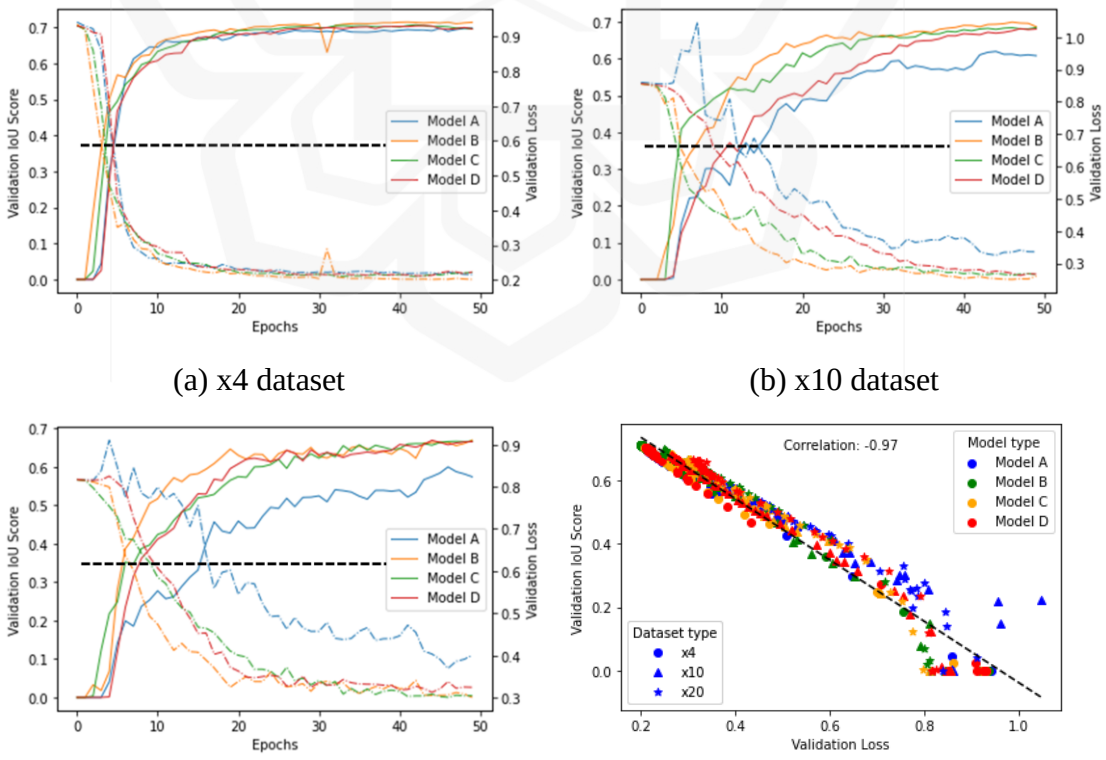
During deployment, the priority is to have a model that can perform the required task efficiently without consuming excessive resources. Consequently, based on Table 5.4, the degree of model complexity would be emphasized on its minimum memory required and inference time. To conclude, Table 5.8 presents an overview of the relationship between model performance and complexity based on the specific training hardware being used, concerning the effectiveness and efficiency of a deployment, respectively. While Model A has the highest performance, it ranks third in terms of complexity, making it less suitable for moderate deployment. Model B has the least complexity, but it holds the lowest performance rating. On the other hand, Model C offers a fine balance between performance and complexity, making it a suitable choice for moderate operation in deployment. This hypothesis serves as an initial observation in this chapter, acknowledging that further analysis regarding the actual adaptability of the models to the specific deployment environment will be discussed in the subsequent chapter.

Table 5.8 Model Performance and Complexity Relationship

Aspects	**Models**			
Performance (highest to lowest)	B	C	D	A
Complexity (least to highest)	A	C	B	D

5.3.4 Relationship Between Model Performance and Loss

The ideal scenario is to have a consistently high IoU score while the loss decreases over epochs. The performance and loss of the proposed models in terms of validation IoU score and loss, as depicted in Figure 5.4. As previously discussed, we found that the x4 dataset was the easiest to train compared to the other datasets. In this dataset, both the IoU score, and loss can exponentially increase and decrease, respectively, and have lowest magnitude of fluctuations, even for the reduced U-Net, model A. According to the graphs, the training process of all the proposed models tends to converge to similar results by the end of the epoch, regardless of their initial momentum.



(a) x4 dataset (b) x10 dataset

(c) x20 dataset (d) Compiled

Figure 5.4 Performance and Loss Comparison Plots

Based on the observation, the IoU score, and loss graphs show a pattern where the lines resemble a mirror image of each other, suggesting that they are correlated to each other. This correlation may indicate that the chosen loss functions effectively guide the model training process, facilitating convergence towards an optimal IoU score. However, it is worth noting that the pattern alone does not guarantee optimal model performance for specific segmentation and deployment tasks. Thus, other considerations, such as the overfitting rate between training and validation data, model consistency over multiple trials, and real-world visual segmentation, as shown in Figure 5.2, Table 5.5 and Figure 5.3, respectively, have been taken into account previously for a comprehensive analysis of the model capabilities. Nevertheless, Model B has demonstrated exceptional performance in segmenting images at different magnification levels while exhibiting a positive correlation with its loss.

5.3.5 Model Performance of Merged Dataset Training

Previously, despite among the segmentation models have correspondingly presented significant performance, they were trained using individual datasets, which meant that three different models of same architecture were required to segment cells at specific magnification levels. To address this issue, the study adopted a revised training approach using transfer learning on merged or collective datasets, as illustrated in Figure 5.5. This approach aims to ensure that each dataset receives an equal amount of training, instead of compiling the datasets at once, which may result in imbalanced training within a batch.

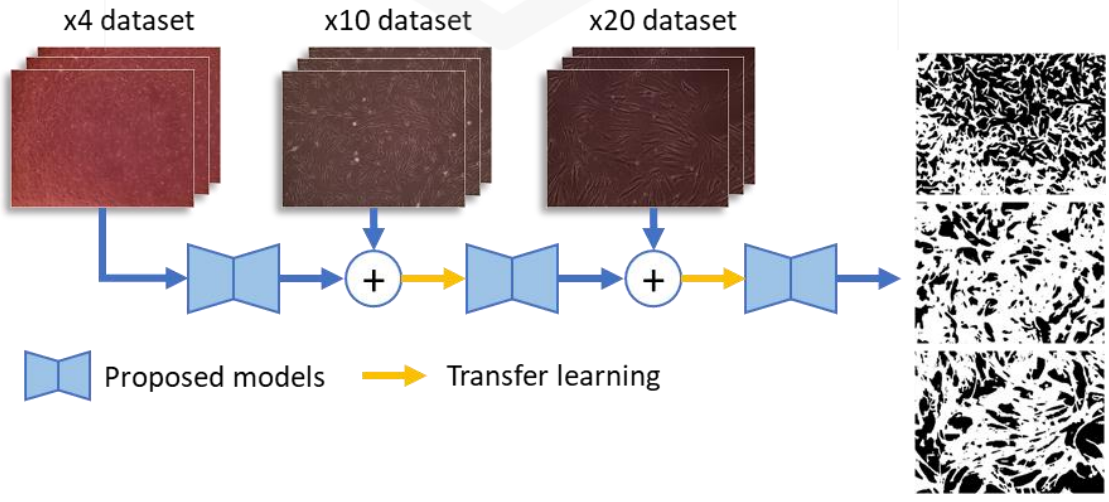


Figure 5.5 Transfer Learning for Merged Training

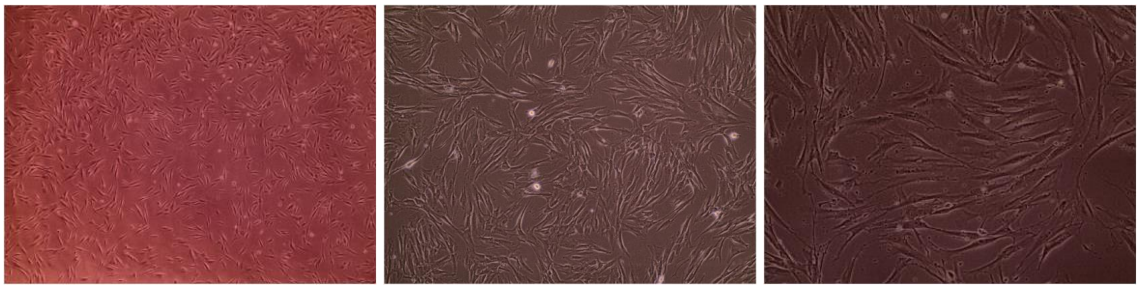
According to Table 5.9, this approach highlights that Model D consistently outperforms the other models across most datasets, particularly in terms of IoU score and precision. Model B, which showed promising performance during individual training, excels only with x20 images. Interestingly, Model C consistently demonstrates the best performance in minimizing under-segmentation, regardless of the dataset employed. Comparing these findings to the individual training results, the average relative IoU performance of this approach shows a drastic decrease of -35.22% for x4 images, a slight decrease of -7.32% for x10 images, and a modest improvement of 5.64% for x20 images, regardless of the model architecture. Overall, this approach mostly exhibits unfavorable performance in terms of statistical records, especially for the lower magnification level images.

Table 5.9 Results of Merged Dataset Training

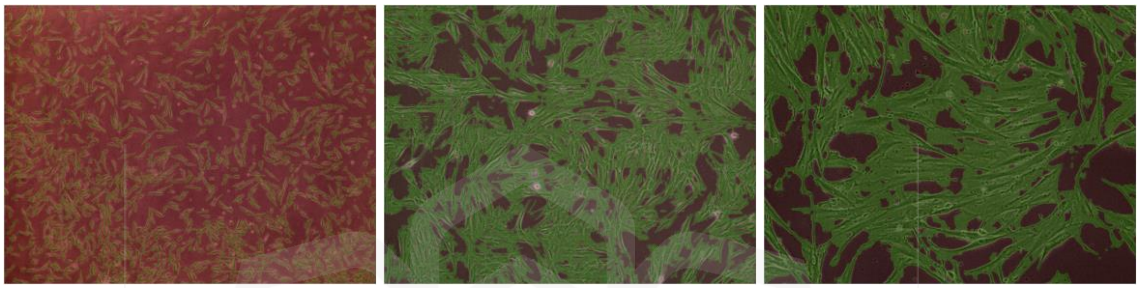
Dataset	**Metrics**	**Model**				**Average**
		A	**B**	**C**	**D**	
x4	**IoU**	0.433	0.472	0.453	0.483	0.460
	Precision	0.388	0.465	0.421	0.481	0.439
	Recall	0.894	0.876	0.910	0.829	0.877
x10	**IoU**	0.618	0.641	0.626	0.649	0.634
	Precision	0.650	0.654	0.631	0.667	0.651
	Recall	0.845	0.892	0.898	0.892	0.882
x20	**IoU**	0.709	0.718	0.722	0.716	0.716
	Precision	0.750	0.774	0.758	0.759	0.760
	Recall	0.845	0.831	0.874	0.851	0.850

Figure 5.6 showcases an example of the visual comparison between the segmentation results obtained from individual and merged training approaches, accompanied by confluency value predictions. While the visual segmentation of cells appears satisfactory, previous statistical observations suggest a high occurrence of over-segmentation due to the low precision metric observed in x4 images. As a result, the confluency value prediction deviates twice as much in the merged training model compared to the individual training model. However, the differences in confluency value predictions for other datasets remain tolerable. Nevertheless, upon observing the overall performance of collective data training, this study prefers to proceed with further analysis using individual data training, which has better universal performance.

Original Images

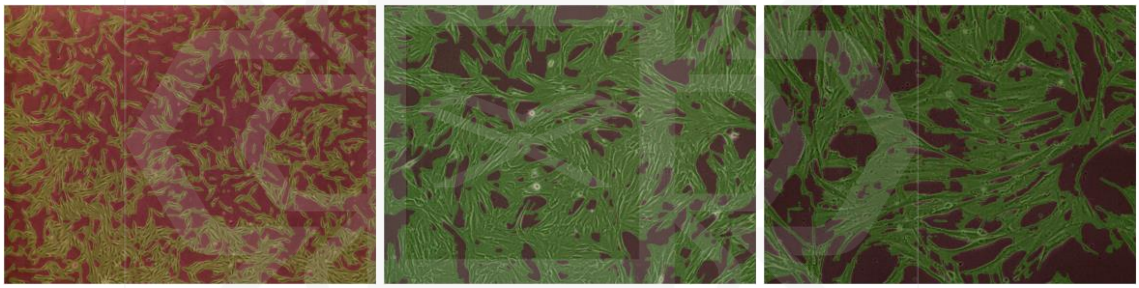


Segmentation Using Individual Training Model



Confluency: 22.03% Confluency: 62.34% Confluency: 64.99%

Segmentation Using Merged Training Model



Confluency: 47.13% Confluency: 71.37% Confluency: 59.57%

(a) x4 image (b) x10 image (c) x20 image

Figure 5.6 Segmentation Comparison Between Individual and Merged Training

5.4 SMOOTH BLENDING FOR PATCH-BASED SEGMENTATION

The original images are 2048x1536 pixels in size, and the model is trained using the selected patch size during training. Inputs used for training and for segmentation should have similar characteristics such as the dimensions. Thus, for predictions, it is also appropriate to employ patch-based segmentation instead of resizing the large and original image into the size of the training patches because it may result in a loss of detailed features and information. The original images will be patched first, and cells will be segmented by patches. After each segmented patch is completed, the patches

are unpatched to obtain the original images with true class labels as shown in Figure 3.18. Essentially, the predictions will return the probability of each pixel. Based on the experiments, pixels with values of 0.5 and above were considered as the true class and vice versa.

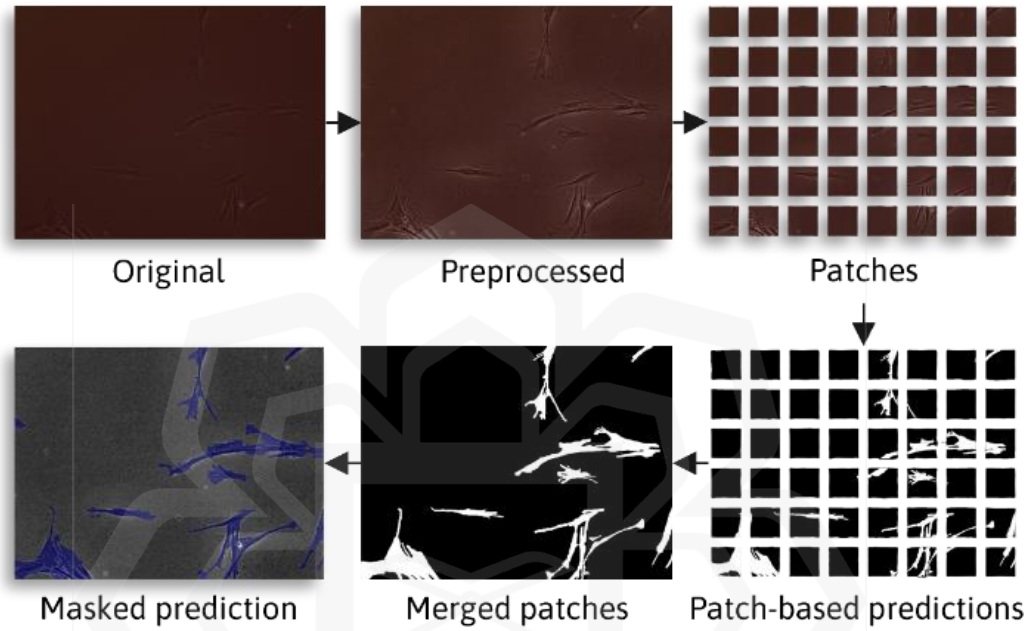


Figure 5.7 Proposed Segmentation Procedures

However, when the procedures were used, the segmented images have edge effects that cause discontinuity between neighboring tiles when they are combined, especially when the tiles have high cell confluency or fuzzy regions as illustrated in Figure 5.8. The effects are clearly visible when the segmentation results are plotted in binary. Consequently, for the patches to fit well in the original images, the segmentation requires smooth blending algorithm, particularly in cases of high confluence.

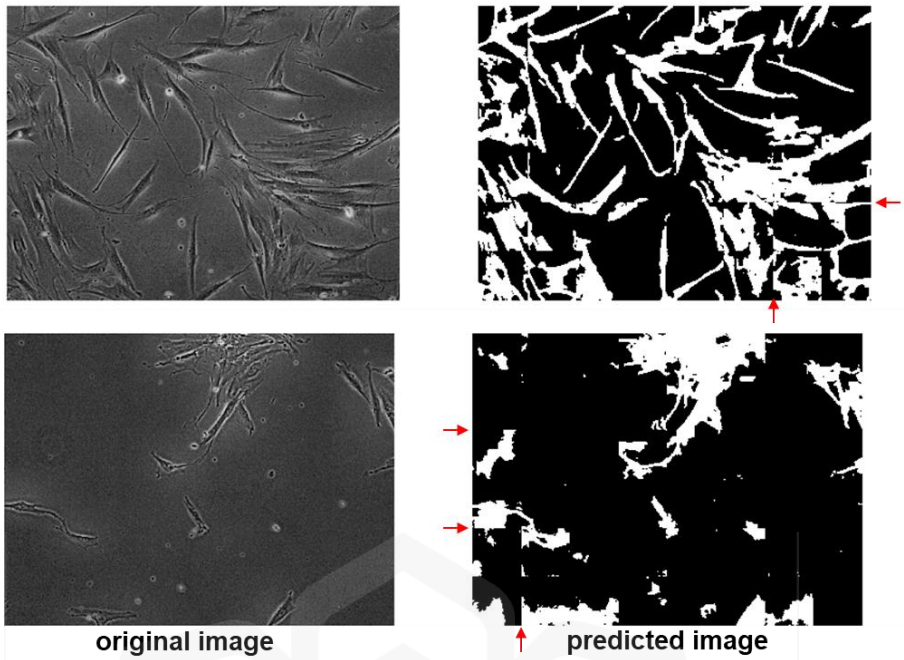


Figure 5.8 Edge Effects of Patch-based Segmentation

To address the edge effect at the border between neighboring patches in patch-based segmentation, overlapping patches were used during the prediction phase, and then apply a blending method to merge the overlapping regions. The procedures of the smooth blending are as shown in Figure 5.9.

Smooth blending algorithm

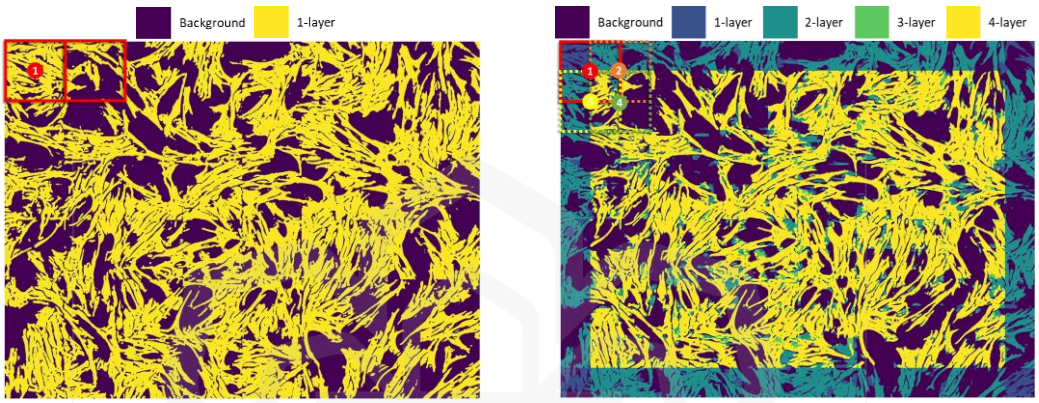
Input: 2048 x 1536 gray image

Output: 2048 x 1536 binary mask

1. Define the patch size (256) and stride (128).
2. Create an empty mask of 2048 x 1536 to store the segmented image (intensity mask shown in Figure 5.9(b)).
3. Iterate over each overlapping patch in the image using two nested for loops:
   a. Extract the current patch using array slicing.
   b. Perform the predictions (> 0.5) on the patch using the selected model.
   c. Blend the predicted patch with the mask using element-wise addition according to its original coordinates.
4. Normalize the intensity mask by dividing it by the maximum value of the mask array.
5. Apply a threshold (0.2) to convert the mask to a binary image.

Figure 5.9 Pseudocode of Smooth Blending Algorithm

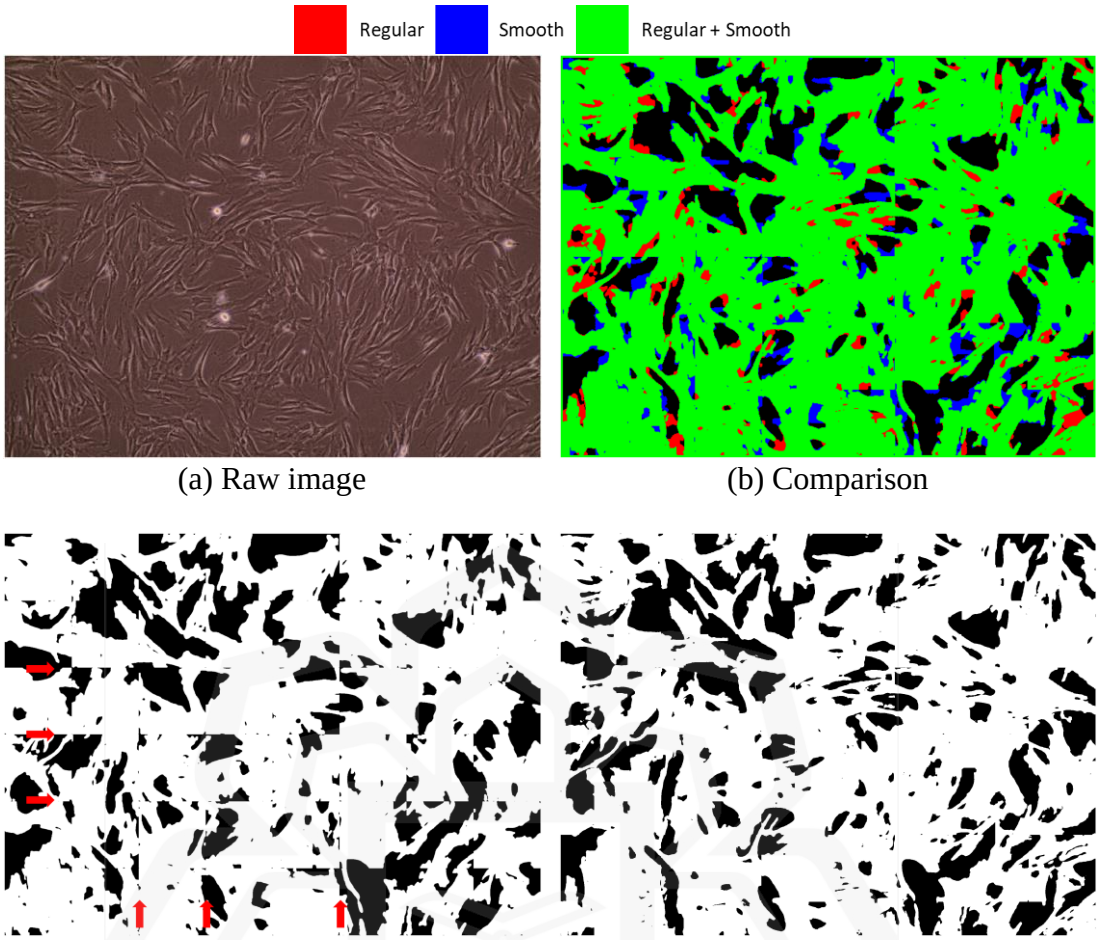
The comparison of the generated mask is illustrated in Figure 5.10. In contrast to regular segmentation, which produces a single-layer mask, the smooth blending algorithm generates a more comprehensive mask comprising four layers. These layers include the primary patch region itself, two edges adjacent to a corner of the patch and the corner region.



(a) Regular mask (b) Intensity mask

Figure 5.10 Mask Generation of Regular and Smooth Segmentation

The comparison of results can be observed in Figure 5.11(b), showing that the algorithm can effectively identify the significant feature across the neighboring patches, preventing over- and under-segmentation due to the patch-based segmentation. The final reconstructed image depicted in Figure 5.11(d) where the edge effects have been reduced, making the results more visually pleasing, compared to Figure 5.11(c). However, the utilization of overlapping patches leads to an increase in the number of patches per image to be segmented. Specifically, the number of patches rises from 48 to 165, making the segmentation or inferencing time approximately 3 times longer than presented in Table 5.4.



(a) Raw image (b) Comparison

Confluency: 73.66 % Confluency: 74.36 %

(c) Without smooth blending (d) With smooth blending

Figure 5.11 Regular to Smooth Blending Result Comparison

5.5 ANALYSIS OF MODEL VALIDATION

This section outlines two validation activities conducted to assess the practical utility of the developed models in real-world scenarios, emphasizing competence over traditional human evaluation and adaptation to natural environments. Unlike the previous section, which primarily assessed performance on confluency segmentation masks, this section shifts its focus to confluency value predictions, an essential aspect for end-users monitoring cell growth in quick and continuous working environments. Precision in confluency segmentation pixels, although visually pleasing, takes a secondary role.

5.5.1 Survey Analysis of Human-Computer Evaluations

This survey underscores two pivotal elements which are the responses received, and the outcomes derived from these responses. The outcomes will detail into the number of individuals participating in the survey, their level of involvement in the cell culture field, their precision and consistency compared to our segmentation models, and the factors influencing their estimations in predictions.

A. Responses

Overall, the survey gathered ten responses from individuals. Although the number of respondents is relatively low for traditional survey standards. However, the participants form a targeted sample, all possessing prior experience in the field rather than representing the general public.

According to Figure 5.12, the respondents were divided with approximately 60% above 30 years of age and 40% below 30 years of age, indicating a slight majority in the older age group. However, none are below 20 years old and above 40 years old. The majority group has spent more than 5 years working in cell culture, while only one respondent had less than 5 years of experience. Similarly, within the minority group, only one respondent, an undergraduate student, has less than 2 years of experience, while the others have 2 to 5 years of experience in cell culture.

Regardless of their age, all respondents are currently in academic field and majority of them had experience in cell culture for at least 2 years. 80% of the respondents were academicians and postgraduate students, where 50% of the academicians have experience above 10 years while the rest only experienced in cell culture between 5 to 10 years and only one postgraduate student has experience above 5 years while the rest only experienced in cell culture between 2 to 5 years.

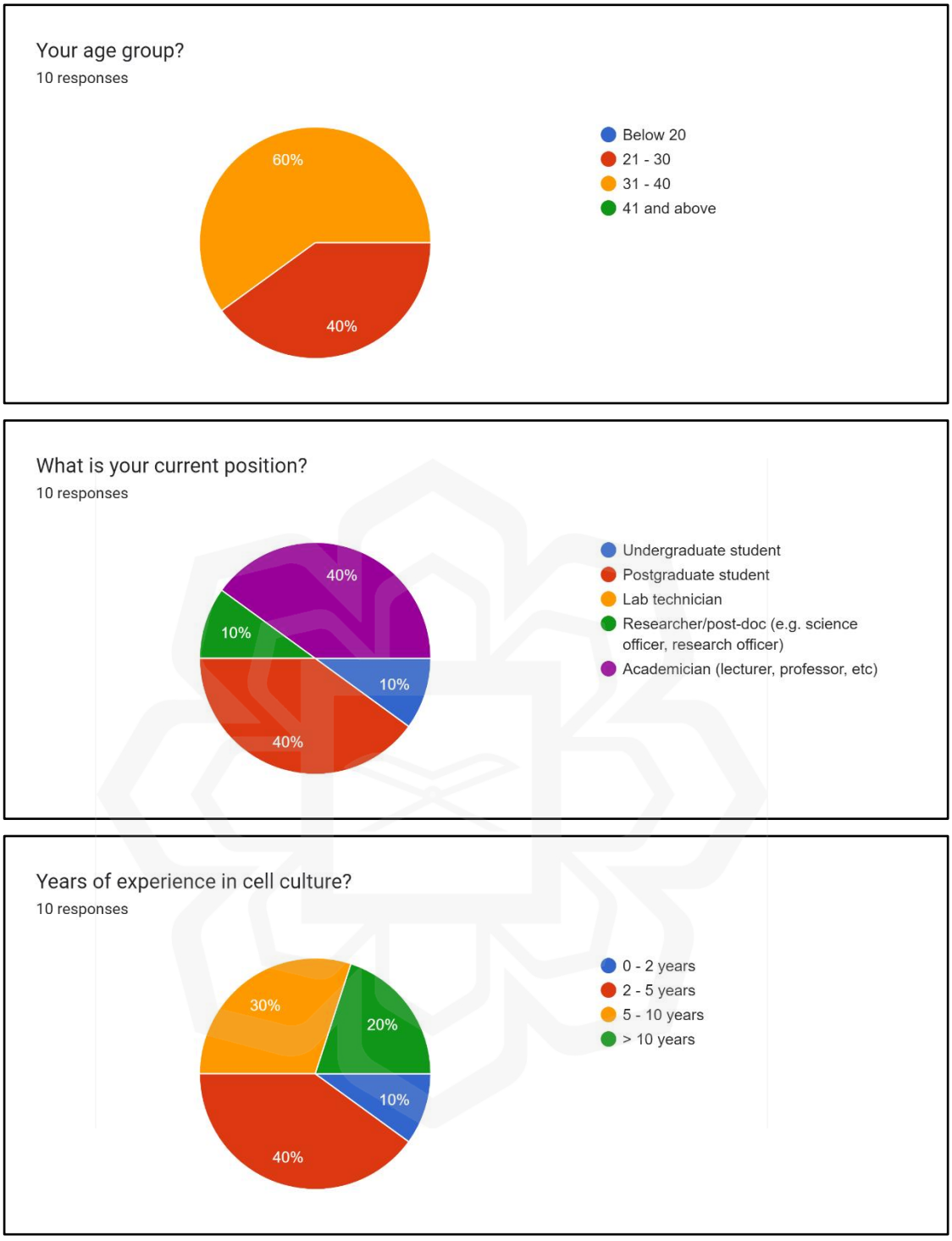


Figure 5.12 Background Information of Respondents

According to Figure 5.13, it is observed that the respondents were not totally in consensus when estimating the confluency value of the provided images. However, regardless of the true value, the highest level of consistency was observed in Image 2, where 7 out of 10 respondents were able to provide the same prediction value, while

the remaining respondents provided varying estimations. Another observation is that some respondents expressed doubt in estimating the confluency value. For example, Respondent 1 and Respondent 7 chose to provide a range of values, such as "40-50%" or "20-30%," rather than a solid value. Thus, the range values were averaged, resulting in values such as "45%" or "25%", allowing for a more relevant analysis of the data. The raw data of the survey is provided in Appendix B.

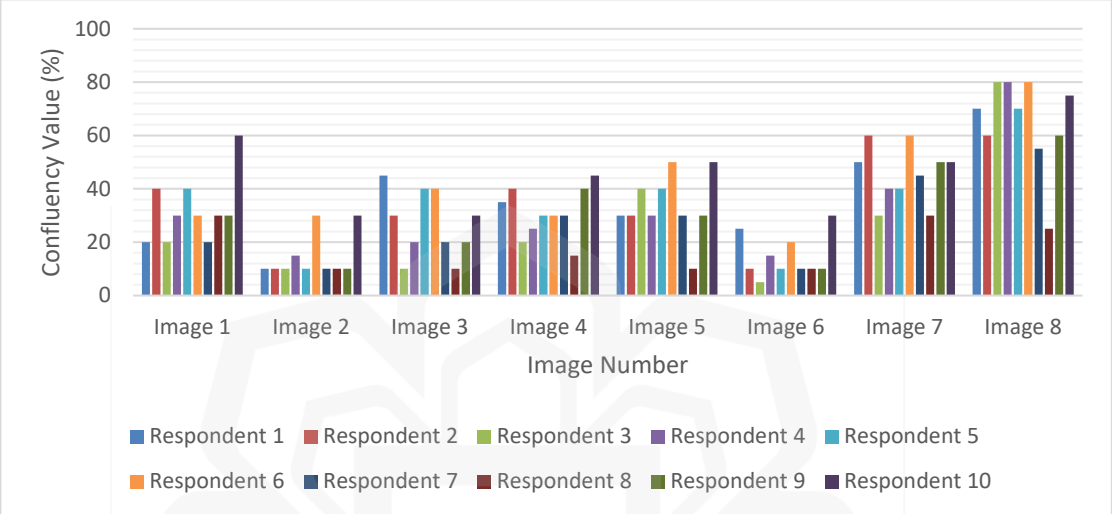


Figure 5.13 Confluency Predictions of Respondents

B. Findings and Discussion

Table 5.10 presents a comparison between the evaluations made by humans and the proposed segmentation model. As mentioned earlier, it shows that the human evaluations were inconsistent compared to the segmentation model. Moreover, the model segmented the cells more closely aligned with the ground truth labels. Relative segmentation differences of the selected model were compared to human observations, and it was found that the model outperformed human observations by 15% in terms of accuracy in estimating the true confluency value.

In terms of prediction difficulty, each magnification level presents a comparable level of challenge for manual segmentation, and there is no clear indication that any one dataset is more difficult to segment than the others, according to Table 5.10. It is also inferred that the difficulty to manually estimate the confluency value is not influenced by the confluency level. The relative differences of human evaluation could be higher or lower regardless of the ground truth's confluency level.

Table 5.10 Overall Respondents' Evaluation of Confluency Measurement

Image No.	Scale	Confluency measurement (%)			Relative differences (%)	
		Ground truth	Segmentation model	Human evaluation	Segmentation model	Human evaluation
1	x4	3.88	4.03	32.0±12.3	0.15	28.12
2	x4	2.55	2.49	15.5±8.3	0.06	12.95
3	x10	13.12	12.53	26.5±12.5	0.59	13.38
4	x10	15.96	15.15	31.0±9.4	0.81	15.04
5	x10	12.88	10.94	34.0±11.7	1.94	21.12
6	x20	10.08	12.33	14.5±8.0	2.25	4.42
7	x20	34.07	37.97	45.5±10.7	3.9	11.43
8	x20	26.91	40.71	65.5±16.9	13.8	38.59
Average ground truth deviations					2.94	18.13
Comparison of human to computer					- 15.19	

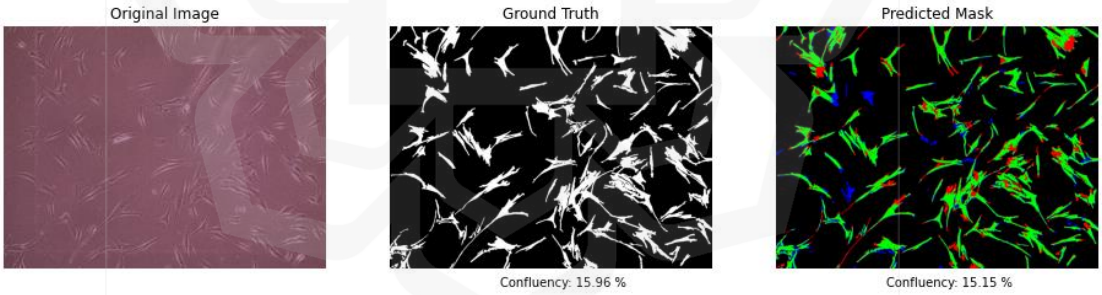
Table 5.11 revealed that the precision of subjective observations made by humans was not influenced by their years of experience in cell culture. This means that even individuals with many years of experience in the field did not necessarily perform better in predicting cell confluency than those with less experience. These findings highlight the need for more precise and consistent automated cell confluency segmentation. As demonstrated in the study, the proposed segmentation model outperformed human evaluation in terms of precision and consistency.

Table 5.11 Snippet of Respondents' Confluency Estimation from Image 3

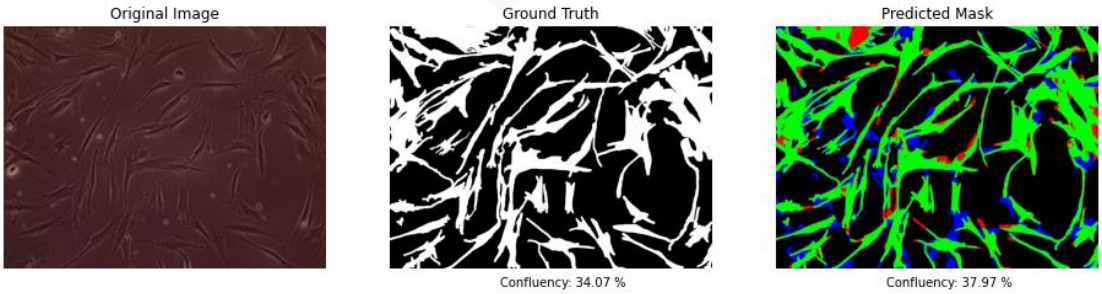
Age group	Current position	Years of experience in cell culture	Estimated confluency	Ground truth differences
31 - 40	Academician	> 10 years	20 %	6.88
31 - 40	Academician	> 10 years	30 %	16.88
31 - 40	Postgraduate student	5 - 10 years	40 - 50 %	26.88
31 - 40	Academician	5 - 10 years	40 %	26.88
31 - 40	Academician	5 - 10 years	20 %	6.88
21 - 30	Postgraduate student	2 - 5 years	30 %	16.88
21 - 30	Researcher/post-doc	2 - 5 years	10 %	3.12
21 - 30	Postgraduate student	2 - 5 years	20 %	6.88
31 - 40	Postgraduate student	2 - 5 years	40 %	26.88
21 - 30	Undergraduate student	0 - 2 years	10 %	3.12

Figure 5.14 presents a visual comparison between the ground truth and predicted mask of each dataset used for this analysis. The model used for the comparisons is Model B as it has the best segmentation performance for overall datasets in this study regardless of other aspects. The annotated images used in this analysis were separated from the segmentation modelling process and were conducted individually without the input of experts, which may have resulted in slight variations in the annotations compared to the true labels. However, despite the potential for error, the annotations were considered sufficient for use as a comparison indicator in the analysis.

(a) x4 image: 20220119-01



(b) x10 image: F1_10X_60_013



(c) x20 image: d1-c3-007

Figure 5.14 Survey Image Segmentations

5.5.2 Experiment Analysis on Natural Cell Cycle Growth Adaptation

All segmentation models in this study were utilized to characterize cell confluency in relation to the number of cells present. The resulting data was graphed and exponential trendlines were generated in Figure 5.15. The exponential trendlines were crafted to mimic the natural growth rate of cells in the real world, as depicted in Figure 1.2, where they grow exponentially due to high media availability and reach stationary growth at the end due to overcrowding and limited media. For this case study, R^2, known as the coefficient of determination, is a statistical measure that represents the proportion of the variance in the cell confluency that can be explained by the number of cells. It ranges from 0 to 1, with 1 indicating a perfect fit and vice versa. The fact that the R^2 value of the trendlines are closer to 1 further proved that the first hypothesis was satisfied. Thus, regardless of the performance of the segmentation models, they were still able to follow the natural growth of the cells.

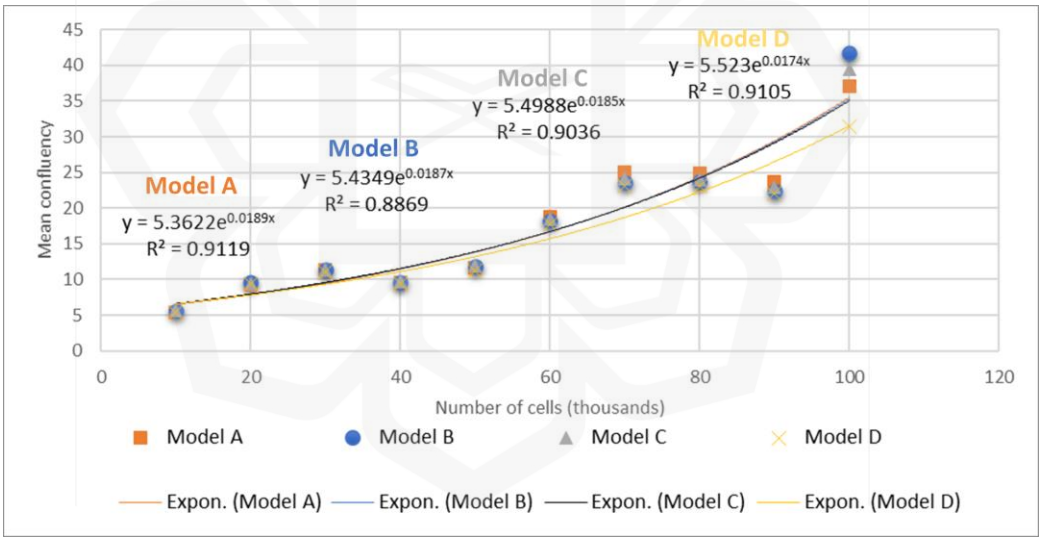
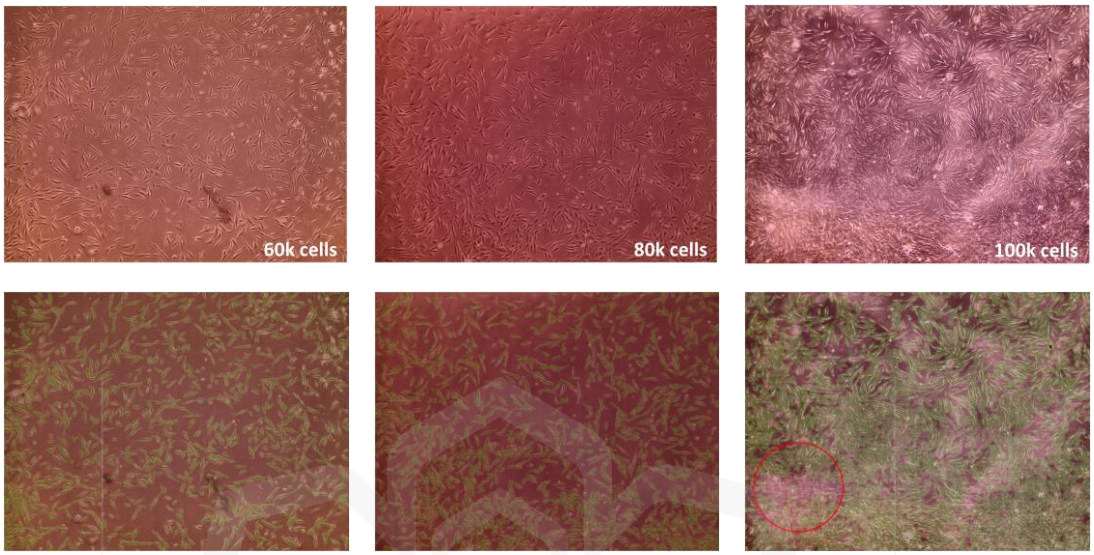


Figure 5.15 Model Segmentations Towards Cell Growth Nature

Another finding is that the higher the confluency in a petri dish, the harder it is to sharpen the microscope image. It is believed because the more cells there are in the field of view, the more complex and cluttered the image becomes, making it more difficult to distinguish individual cells or details as the light passing through the sample may be scattered or diffracted, resulting in a blurred or hazy image. Therefore, the higher the confluency, the sharpness of the microscope needs to be adjusted. The scenario is depicted in Figure 5.16. Consequently, the second hypothesis was proven

valid, as a higher confluency of cells leads to a loss of constant quality, making it more challenging to segment as other microscope settings are maintained.



Confluency: 17.86% Confluency: 22.03% Confluency: 55.58%

Figure 5.16 Image Quality of Various Cell Confluency

5.6 ANALYSIS ON DEPLOYED MODEL PERFORMANCE

As the models have been validated, they were deployed and evaluated. Two metrics are considered to evaluate the performance during deployment which are inference time and max power consumption by Raspberry Pi. Inference time refers to the time it takes for the model to generate segmented cell image, while power consumption is the amount of energy consumed only during segmentation process.

Table 5.12 provides a comparison of the average segmentation or inference time of different models on both a personal computer and a Raspberry Pi with a specific processing unit. One of the significant findings is that the OpenVINO framework is more efficient in reducing the inference time compared to the Keras framework on both platforms. However, using the Neural Compute Stick 2 (NCS2) only results in improved inference time on the Raspberry Pi as an edge device, not on the personal computer. The study also discovered that NCS2 cannot efficiently optimize dilation mechanism. This is noticeable because Model C always has lower inference time than Model B when using NCS2 while vice versa without using it and the only difference

between Model B and Model C is the dilation mechanism. This issue reduced the performance and complexity gaps of Model B shown in Table 5.8. Consequently, the hypothesis made in the previous chapter, considering Model B as a moderate consideration for deployment in this study, is deemed invalid.

Based on the above findings, it is concluded that Model B is the most efficient option for deployment. This is because Model B has the highest performance among all models and has manageable inference time when OpenVINO and NCS2 are employed.

Table 5.12 Model Performance Comparisons During Deployment

Platform	**Method**	**Device**	**Model**	**Inference time (secs)**	**Max power consumption (Watts)**
Raspberry Pi	OpenVINO	VPU (Intel Movidius Myriad X)	Model A	4.84 ±0.21	8.00
			Model B	8.91±0.12	8.40
			Model C	14.47±0.14	8.70
			Model D	15.07±0.43	8.70
	Keras	CPU (ARMv7 Processor rev 3)	Model A	28.58±0.37	8.00
			Model B	88.47±2.03	8.70
			Model C	64.84±0.35	8.40
			Model D	85.47±7.85	8.70
PC	OpenVINO	CPU (Intel i5-11400F)	Model A	0.62±0.08	75.60
			Model B	1.85±0.06	105.00
			Model C	1.12±0.08	92.40
			Model D	1.27±0.13	95.90
		VPU (Intel Movidius Myriad X)	Model A	3.70 ±0.02	63.70
			Model B	7.84±0.02	70.00
			Model C	13.32±0.01	87.50
			Model D	13.96±0.02	112.00
	Keras	CPU (Intel i5-11400F)	Model A	3.07±0.07	117.30
			Model B	5.07±0.21	141.80
			Model C	5.12±0.11	131.30
			Model D	5.44±0.14	168.00
		GPU (Nvidia RTX 3060)	Model A	2.21±0.34	105.00
			Model B	2.59±0.23	117.30
			Model C	2.52±0.25	112.10
			Model D	2.68±0.19	120.00

5.7 ANALYSIS ON CLIENT-SERVER COMMUNICATION

This analysis evaluates the data communication performance among sensors, microprocessor, online database, and mobile app, considering real-time, periodic, and log data within the workflow of the designed system mode.

5.7.1 Real-time Sensor Monitoring

Based on the serial monitor in Figure 5.17, it shows the readings of the incubator sensors, including their corresponding timestamp. By observing the timestamp, it is evident that the sensors are able to report the environmental and light conditions quickly as the time difference between consecutive readings is approximately a second. It indicates that the sensors can provide near real-time updates on the incubator conditions they are measuring.

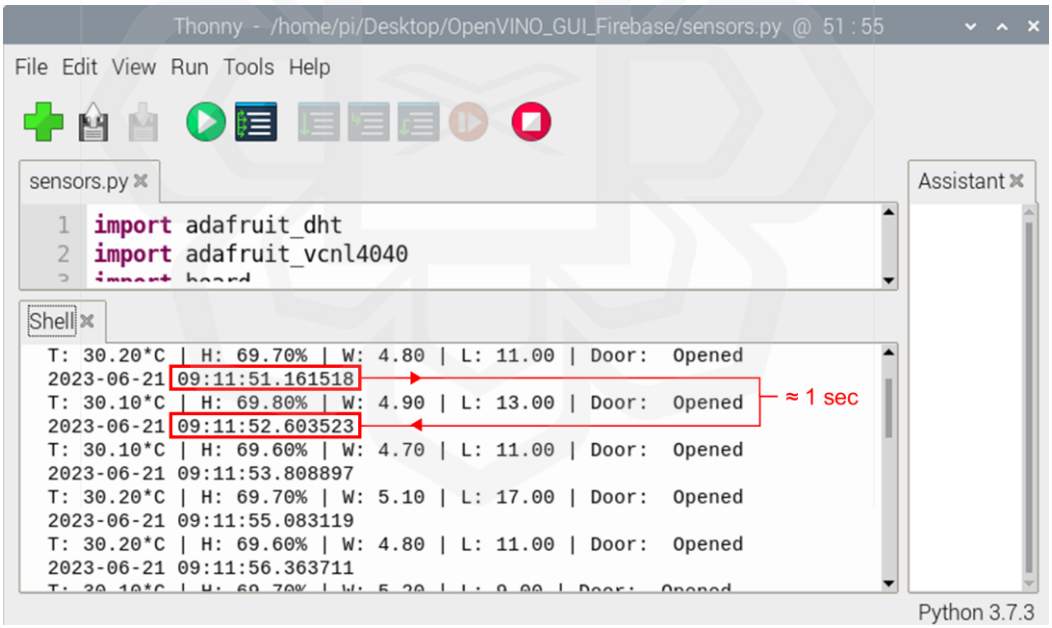


Figure 5.17 Sensor Data on Raspberry Pi Serial Monitor

The activity in the Firebase console, as depicted in Figure 5.18, showcases an ongoing data transfer process involving the sensors of an incubator. The local system, in this case, a Raspberry Pi, is responsible for collecting and transmitting sensor data to the Firebase Real-time Database.

The actively transmitted sensor data is highlighted in orange within the Firebase console, indicating its continuous and real-time synchronization between the Raspberry Pi and the Firebase database. In cases where a sensor reading remains unchanged from the previous reading, the corresponding variable is unhighlighted. This dynamic transfer allows for up-to-date monitoring of the incubator's environmental conditions.

Simultaneously, the Firebase console also presents a separate section (parent: Sensors) dedicated to logged data. These log entries encompass the collective real-time data received from the sensors and are represented by a blue highlight in the console as it is actively recorded. The log data is stored within the Firebase Real-time Database, providing a historical record of the sensor readings over time.

In brief, the highlights in the Firebase console provide a comprehensive overview of any transmitted data, enabling real-time monitoring as well as the ability to review and analyze past readings.

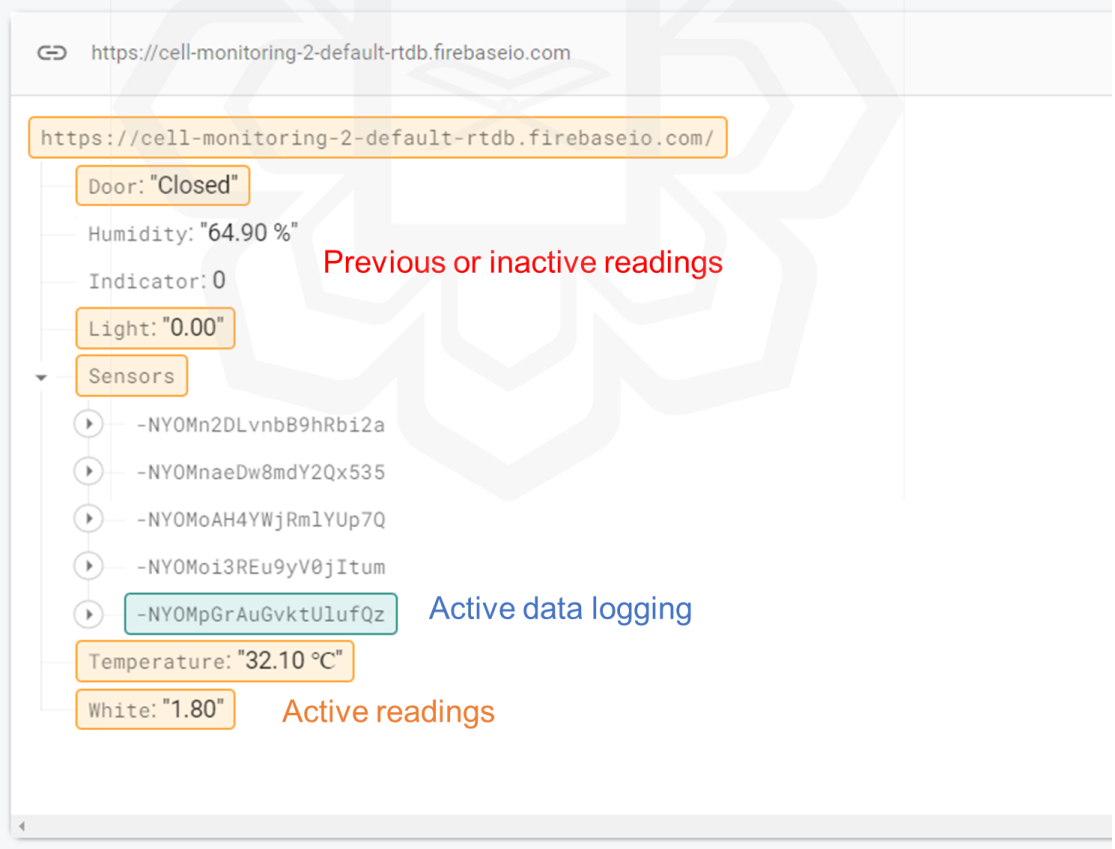


Figure 5.18 Active Sensor Data Transmit to Firebase Real-time Database

Figure 5.19 shows downloads of sensor data transfer over Internet connection to Firebase Realtime Database that account for all bytes downloaded from the database, including connection protocol and SSL encryption overhead, over one-minute intervals. Therefore, there may be occasional spikes in the graph within the hour, indicating additional bytes required for establishing and maintaining the connection, as well as encrypting the data to ensure secure transmission. According to the collected data, the real-time sensor monitoring system recorded a total download size of 6.19 MB over the course of an hour, with an average download rate of 133.93 kB per minute.

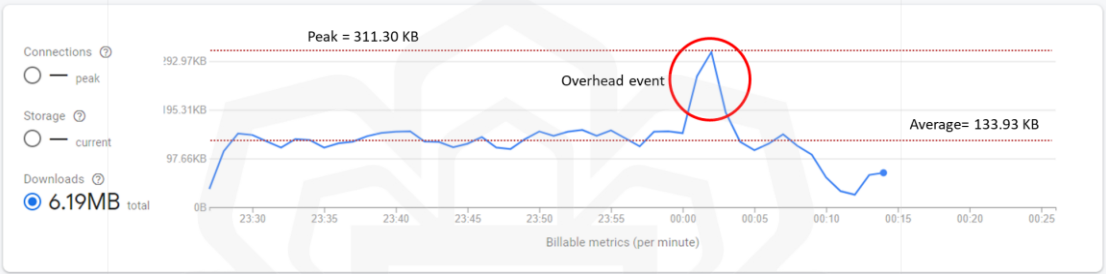


Figure 5.19 Sensor Data Download Rate Over One-Minute Interval for an Hour

5.7.2 Interval Cell Culture Monitoring and Handling

In contrast to the real-time sensor reading transmission, the update of cell information and system mode occurs at regular intervals. To monitor the interval updates, timestamps are included in the transmission. This distinction is important because sensor readings provide immediate reporting, while cell information and system mode updates follow a predefined schedule. the system's performance in terms of latency is highly dependent on the quality of the internet connection. As a result, there is no critical need for in-depth analysis of the data transfer rate to the Firebase platform. However, during testing phases, it has been observed that the system is capable of transferring data in real-time once the local system has obtained a complete result.

Figure 5.20 presents an ongoing data transfer process involving the cell information, including cell image label and cell confluency, and triggered the “special case”, involving a decision-making point for the user, where they are presented with an authorization prompt. The user has a limited timeframe of 10 seconds, as set during testing, to provide their authorization response.

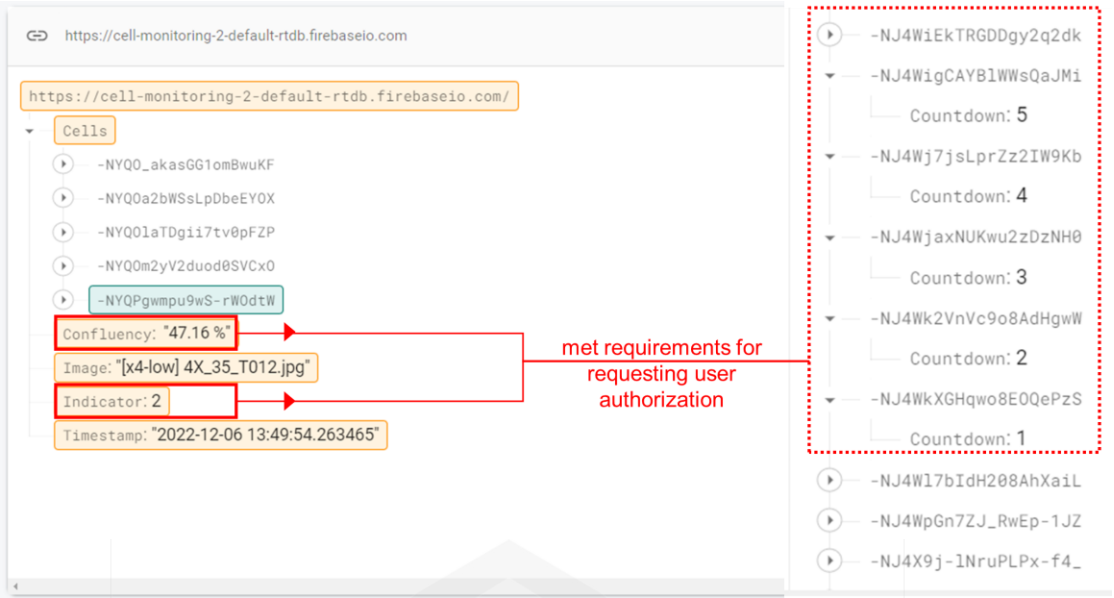
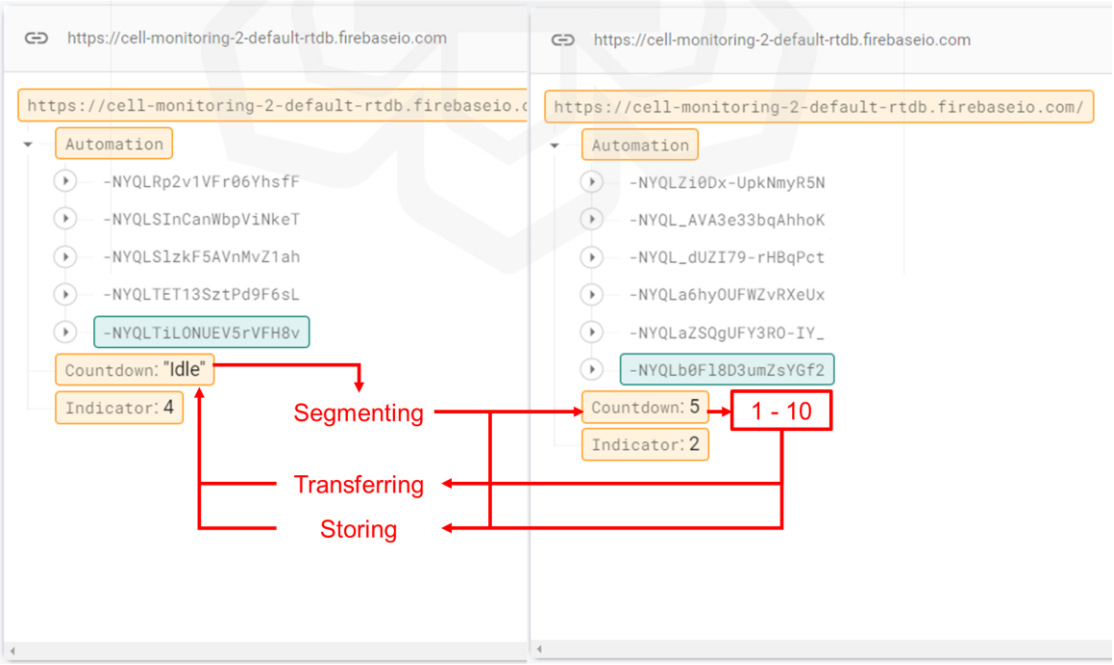


Figure 5.20 Transmission of Cell Information to Firebase Console

Figure 5.21 depicts an ongoing system mode transition within the AUTOMATION data and illustrates the possible transitions of “Countdown” variable within the Firebase console. The variable is one crucial element in this data transmission, which carries two types of information: the current system mode and a countdown related to the authorization process. This provides real-time insights into the ongoing operations and the timing of important events within the system.



(a) During operating system (b) During user authorization

Figure 5.21 Transmission of System Automation Information to Firebase Console

5.7.3 Log Data for Past Transmissions

Previously, the discussion mostly focused on the ongoing data transfer. Figure 5.22 presents a visual representation the log data encompassing a comprehensive record of all the transmitted data captured and stored through the Firebase system. The log data is accessible for identifying any errors or irregularities that may have arisen during the data transmission process.

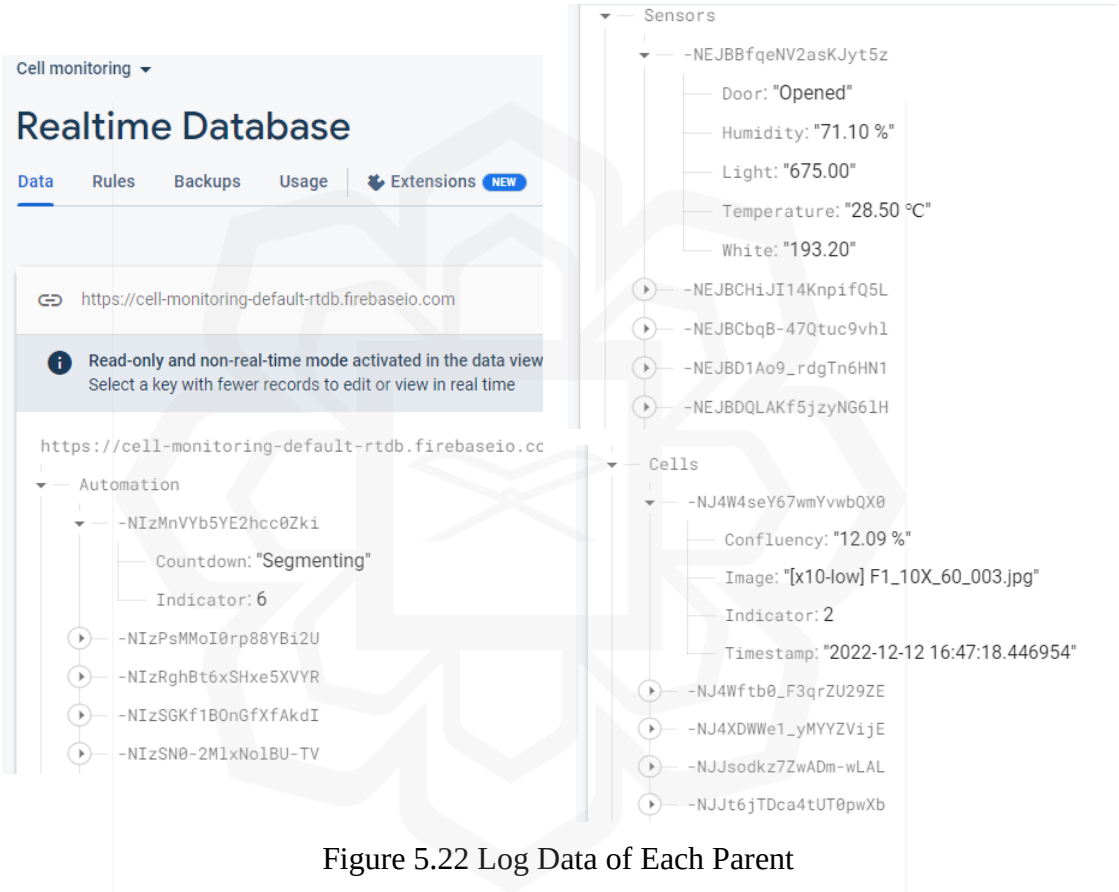


Figure 5.22 Log Data of Each Parent

5.8 ANALYSIS ON SYSTEM WORKFLOW AND POWER RATE

Integrating the system mode design and the implementation of multi-threading for automation, this analysis encompasses the efficiency of the system workflow and the power requirements associated with the specified motion sequence for a complete cycle, offering insights into its real-world applicability in a cell culture lab.

5.8.1 Efficiency of System Workflow

Figure 5.23 illustrates the demonstration workflow of the sample handling process with respect to Table 4.2, which involves transferring the sample between incubator and microscope for imaging using a robotic arm. A three-levels display box was included to represent the cell culture incubator for the demonstration. The top level indicates low confluency, which would be segmented again on the next periodic check, while the lower level indicates high confluency, which has already been segmented and is ready for the next cell culture procedures. Achieving a seamless workflow without collisions or sample spills within a confined space, the average workflow would take just under 1 minute, excluding the countdown period for user authorization.

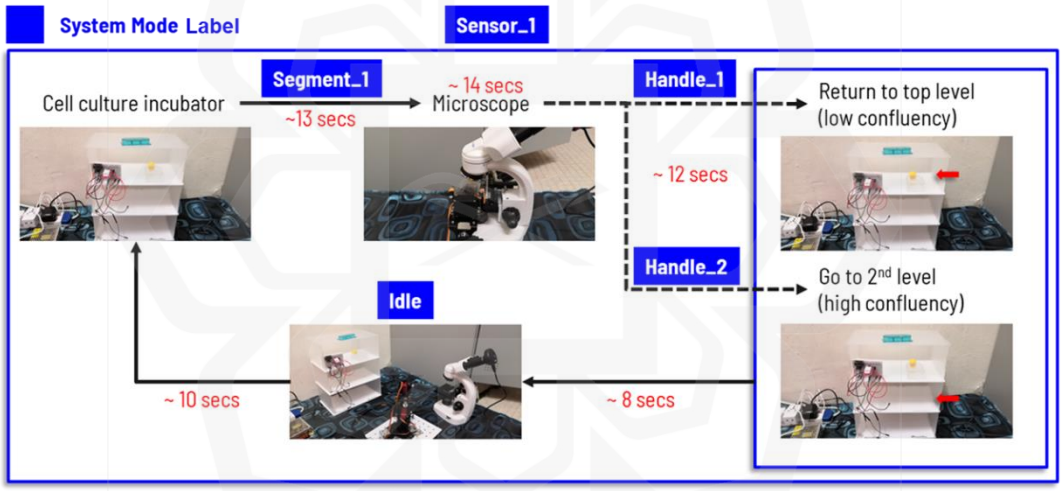


Figure 5.23 Sample Handling Workflow

5.8.2 Power Rate of Motion Set

As illustrated in Figure 5.23, the system workflow consists of four fundamental robotic motions. The power rate associated with each motion is shown in Figure 5.24. Specifically, the average power rate for transferring samples to the microscope is 12.02 W, for transferring samples to a low confluency level is 12.61 W, for transferring samples to a high confluency level is 13.03 W, and for positioning to the idle state is 10.48 W. Collectively, when considering the combination of all the

motions, the average power rate is 12.28 W, falling within the range typically suitable for home or laboratory appliances.

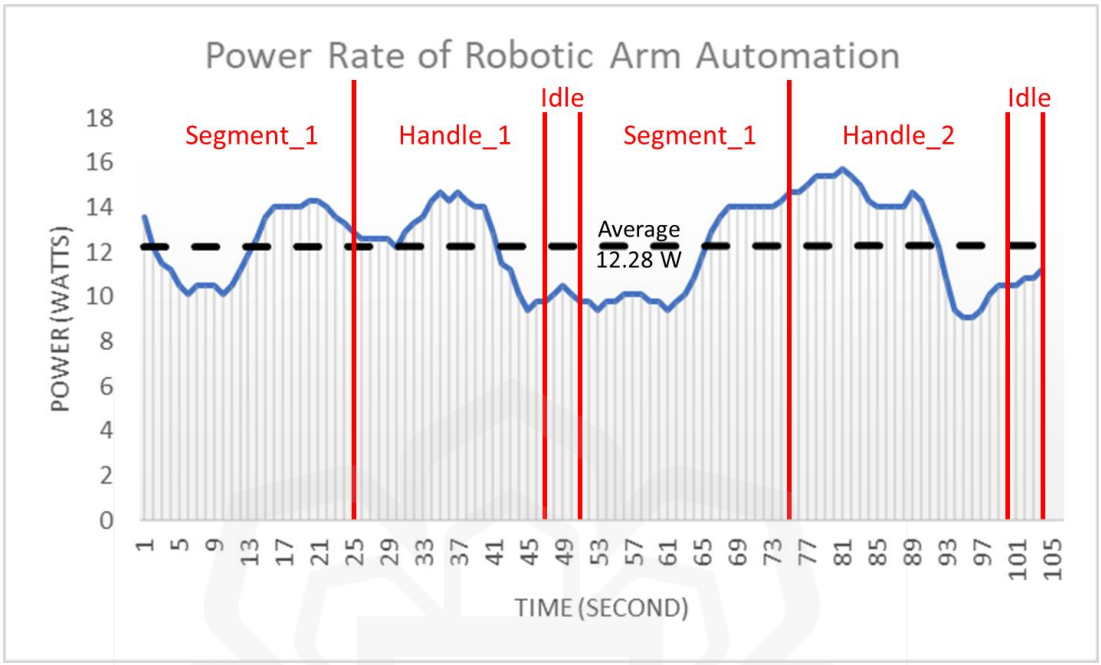
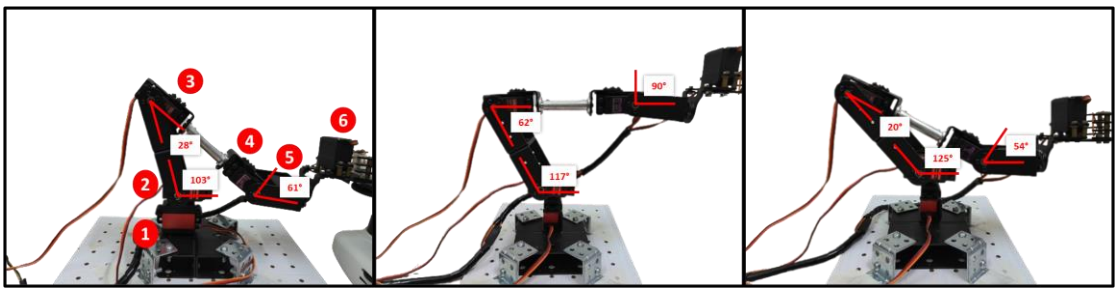


Figure 5.24 Power Rate of Robotic Arm Automation

One finding is that the robotic arm exhibits a higher power usage during sample transfers to lower levels compared to higher levels, as observed through the motions of Handle_1 and Handle_2. A possible reason for this observation could be attributed to the considerable difference in rotation of servo 2, 3 and 4, with some servos potentially reaching their rotation limit during the transition from the microscope to the specific levels within a restricted space, as portrayed in Figure 5.25.



(a) At microscope (b) At higher level (c) At lower level

Figure 5.25 Illustration of Servo Rotation During Transitions

5.9 CHAPTER SUMMARY

This chapter extensively discussed the model performance across the entire pipeline, encompassing deployment, integration with remote monitoring, and the sample handling system.

Initially, model C with a dilation mechanism was selected for its optimal performance with moderate complexity. However, after further analysis, model B with the inception mechanism was then selected based on its highest overall performance, considering the aspects of model complexity during the deployment on OpenVINO framework and Intel VPU. The study also found that Model B is indeed the superior choice for adapting to individual datasets. However, when it came to the merged dataset, Model D outperformed Model B, particularly on images with lower magnification levels by 1.79 % in terms of IoU score. Nevertheless, all refined models worked better than the baseline model A, outperforming human manual evaluation, and accurately representing cell growth cycle nearly the end of its exponential phase.

The communication framework between the local system, cloud database, and user interface was also detailed, including its transmission rate, the arrangement of transmitted data in the database, and the reception of user input. The study showed that the system can operate near real-time environment depending on the internet connection. Furthermore, for a complete day of operation, the system only needs downloading data of less than 200MB, where the users can view all the data in real-time as well as access it for later analysis.

With regards to the automation system, the study emphasized the significance of multi-threading in arm motion, providing insights into its implementation and outlining expected power requirements for specific motions. Given a fixed experiment space, the system can complete a full cycle in approximately 1 minute with an average power rate of 12.28 W, ensuring dish parallelism and avoiding collisions.

In summary, the developed design is relevant for real-world applications that require precision, consistency, efficiency, and safety.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

In this study, a complete pipeline for fibroblast segmentation pipeline was successfully designed, assessed, and validated starting from data preprocessing until the post-segmentation. The model developed from the pipeline worked well for further deployment to automate cell culture monitoring. Thus, all objectives were achieved. The following subsections discuss the findings obtained through the study to address specific objectives.

6.1.1 Pipeline and Model Development

Before the modelling, the characteristics of the datasets were studied to determine the appropriate inputs for the training process. Two preprocessing techniques were employed which are CLAHE and TVD for improving the visibility of object of interest and removing excessive noise, respectively. The effectiveness of the techniques was observed across various magnification and confluency levels and their conditions were highlighted. Three phases of refinement have been successfully carried out on the U-Net architecture, resulting three models were developed, and the approaches include the inception mechanism, dilation, and residual, chosen according to the main characteristic of the datasets which requiring the adaption of multi-scale and irregular object dimensions.

6.1.2 Model Assessment

A successful development of a comprehensive evaluation framework was achieved, where three aspects were utilized to assess the performance of the proposed models, namely their learning graphs, test scores, and visual segmentations. The overall highest performance on the three aspects was yielded by Model B, which employs the

inception mechanism of multiple kernel sizes, as shown by the results on overall datasets. Firstly, the overfitting rate of its learning graphs was the lowest. It was also discovered that x20 images are the most difficult to learn as they possess the most complex features as observed. Regardless of various training settings, x20 image training still results in the highest overfitting rate. Secondly, Model B exhibited the highest relative increment in test scores compared to other models with an overall IoU value of 0.707. Thirdly, the visual segmentation results indicated the strongest relationship between IoU and precision, as summarized in Table 5.7. It is noticeable that regardless of bad learning rate, all magnification levels, that were visually segmented, is evidently acceptable compared to the ground truth labels, as demonstrated in Figure 5.14. Based on the analysis framework, it was also discovered that IoU and precision of a model could oppose each other on specific datasets.

During the study of model complexity, Model B has the highest trainable parameters, but Model D proved to have higher minimum memory required for about 1 GB in difference. This was due to the fact that Model D had higher feature weights, which were attributed to its larger number of layers as inspected. Moreover, it was found that a higher number of trainable parameters did not necessarily result in increased performance, as evidenced by Model C and Model D, unless the architecture was relevant to the datasets. It is well-known that deep learning is a black box in nature as the reasoning behind refinements may be relevant but may not necessarily be attributed to the dataset training.

6.1.3 Model Validation

An online survey and lab experiment were successfully conducted to compare and validate the performance of the segmentation models against subjectively segmented by the people in which have various biology backgrounds. One significant finding is that the human evaluation was not influenced by their years of experience in cell culture which urges more the needs of automated cell segmentation tools for precise and consistent measurement, allowing more efficient screening and production. Moreover, the selected model showed that it outperformed manual segmentation in estimating the confluency value by 15% more accuracy. The segmentation models were also tested on real-world scenarios where the further the cell monitoring

proceeded, the higher the possibility of cell imaging quality and complexity increases. Regardless of the segmentation precision, all models were able to track the cell growth exponentially between the lag phase and stationary phase, as depicted in Figure 1.2. Another finding is that the models were unable to precisely segment the cells towards the end of the exponential phase. This is because the cells tend to get overcrowded in limited space, leading to uneven lighting, resulting in under-segmentation issues, as shown in Figure 5.16. However, this issue is trivial for a deployment as it occurs where the cells were near to being passaged or transferred.

6.1.4 Model Deployment

Last but not least, the analysis on the deployment strategies. A design of an automated cell culture monitoring system was successfully demonstrated, which includes AIoT-based monitoring and communication system, and sample handling automation. Based on the findings in section 5.6, it concluded that Model B is the most relevant for the deployment of monitoring system as it has highest performance among others and showed lowest inferencing time when using OpenVINO framework and NCS2. Overall, one cycle of the system only took under 1 minute, considering users do not delay the authorization. The system used internet connection to communicate, thus, the latency of the system is highly dependent on it. As all data shown in Table 4.2, are communicated to the Cloud Firebase, the workflow of sample handling could stop if the system mode is not updated through the internet connection. Chapter 5 have discussed in detail on how the data can be managed through the Firebase, allowing the seamless interaction between the local system and remote app online. The application of Raspberry Pi and OpenVINO in this specific field is relatively low, thereby constituting an added value resulting from this study. It also demonstrates their capability to integrate seamlessly with other desired systems through cost-effective implementations.

6.2 FUTURE WORK

As of now, the segmentation models are mainly trained on individual datasets. Consequently, three models were generated for each architecture, corresponding to the three different magnification levels, rather than one model fitting all datasets. Training on merged datasets was conducted, however, it resulted in poor performance, particularly on x4 images, due to the dominance of x10 and x20 images in generalizing the significant features. To address this issue, an image classification system can be developed to classify the magnification levels of images, which can then be used to select the appropriate models for segmentation. The scope of this study is limited to binary segmentation and analysis of how the proposed models perform on each individual dataset which has satisfied the objectives of this study. However, it does not provide a comprehensive analysis of how the models perform when used simultaneously on multiple datasets.

In terms of the deployment strategies, remote monitoring using mobile apps may be also developed for IOS phone to increase its feasibility towards various users. Another possible improvement is that the data logs can be directly accessed through mobile apps as of now, it is only applicable from Firebase. Moreover, the ability to generate documentation of periodic results should be recommended for quick evaluation of all stacked results. Other considerations for the automation system are to have an adaptive trajectory for varying environments for market scaling and to develop another system mode for emergency situations for user and sample safety, and data losses.

REFERENCES

Aida, S., Okugawa, J., Fujisaka, S., Kasai, T., & Kameda, H. (2020). Deep learning of cancer stem cell morphology using conditional generative adversarial networks. Biomolecules, vol. 10, no. 6, 931.

Al-Dulaimi, K., Banks, J., Nugyen, K., Al-Sabaawi, A., Tomeo-Reyes, I., & Chandran, V. (2021). [Segmentation of white blood cell, nucleus and cytoplasm in digital haematology microscope images: a review-challenges, current and future potential techniques]. IEEE Reviews in Biomedical Engineering, vol. 14, 290-306.

Anjum, M. A., Amin, J., Sharif, M., Khan, H. U., Malik, M. S. A., & Kadry, S. (2020). Deep semantic segmentation and multi-class skin lesion classification based on convolutional neural network. IEEE Access, vol. 8, 129668-129678.

Aspert, T., Hentsch, D., & Charvin, G. (2022). DetecDiv, a generalist deep-learning platform for automated cell division tracking and survival analysis. ELife, vol. 11, e79519.

Baudin, P. V., Ly, V. T., Pansodtee, P., Jung, E. A., Currie, R., Hoffman, R., Willsey, H. R., Pollen, A. A., Nowakowski, T. J., Haussler, D., Mostajo-Radji, M. A., Salama, S. R., & Teodorescu, M. (2022). Low-cost cloud-based remote microscopy for biological sciences. Internet of Things, vol. 18, 100454.

Belevich, I., Joensuu, M., Kumar, D., Vihinen, H., & Jokitalo, E. (2016). Microscopy image browser: A platform for segmentation and analysis of multidimensional datasets. PLoS Biology, vol. 14, no. 1, e1002340.

Benčević, M., Galić, I., Habijan, M., & Babin, D. (2021). Training on polar image transformations improves biomedical image segmentation. IEEE Access, vol. 9, 133365-133375.

Campiche, R., Trevisan, S., Séroul, P., Rawlings, A. V., Adnet, C., Imfeld, D., & Voegeli, R. (2019). Appearance of aging signs in differently pigmented facial skin by a novel imaging system. Journal of Cosmetic Dermatology, vol. 18, no. 2, 614-627.

Cao, J., Zhao, Z., & Yan, H. (2018, October). Accurate cell segmentation based on biological morphology features. Paper presented at International Conference on Systems, Man, and Cybernetics organized by IEEE, Japan, pp. 3380-3383.

Chambolle, A. (2004). An algorithm for total variation minimization and applications. Journal of Mathematical Imaging and Vision, vol. 20, 89-97.

Chen, S., Yang, H. A. N., Fu, J., Mei, W., Ren, S., Liu, Y., Zhu, Z., Liu, L., Li, H., & Chen, H. (2019). U-Net Plus: Deep semantic segmentation for esophagus and esophageal cancer in computed tomography images. IEEE Access, vol. 7, 82867-82877.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., & Wei, Y. (2017, October). Deformable convolutional networks. Paper presented at International Conference on Computer Vision organized by IEEE, Italy, pp. 764-773.

Din, N. U., & Yu, J. (2021). Unsupervised deep learning method for cell segmentation. bioRxiv.

Doulgkeroglou, M.-N., Di Nubila, A., Niessing, B., König, N., Schmitt, R. H., Damen, J., Szilvassy, S. J., Chang, W., Csontos, L., Louis, S., Kugelmeier, P., Ronfard, V., Bayon, Y., & Zeugolis, D. I. (2020). Automation, monitoring, and standardization of cell product manufacturing. Frontiers in Bioengineering and Biotechnology, vol. 8.

Eschweiler, D., Spina, T. V., Choudhury, R. C., Meyerowitz, E., Cunha, A., & Stegmaier, J. (2019, April). CNN-based preprocessing to optimize watershed-based cell segmentation in 3D confocal microscopy images. Paper presented at International Symposium on Biomedical Imaging organized by IEEE, Italy, pp. 223-227.

Gadosey, P. K., Li, Y., Agyekum, E. A., Zhang, T., Liu, Z., Yamak, P. T., & Essaf, F. (2020). SD-UNet: Stripping down U-Net for segmentation of biomedical images on platforms with low computational budgets. Diagnostics, vol. 10, no. 2, 110.

Gallois, B., & Candelier, R. (2021). FastTrack: An open-source software for tracking varying numbers of deformable objects. PLOS Computational Biology, vol. 17, no. 2, e1008697.

Ghahremani, P., Marino, J., Dodds, R., & Nadeem, S. (2022, June). DeepLIIF: An online platform for quantification of clinical pathology slides. Paper presented at Conference on Computer Vision and Pattern Recognition organized by IEEE Computer Society, USA, pp. 21399-21405.

Gholami, P., Ahmadi-Pajouh, M. A., Abolftahi, N., Hamarneh, G., Kayvanrad, M. (2018). Segmentation and measurement of chronic wounds for bioprinting. IEEE Journal of Biomedical and Health Informatics, vol. 22, no. 4, 1269-1277.

Gu, Z., Cheng, J., Fu, H., Zhou, K., Hao, H., Zhao, Y., Zhang, T., Gao, S., & Liu, J. (2019). CE-Net: Context encoder network for 2D medical image segmentation. IEEE Transactions on Medical Imaging, vol. 38, no. 10, 2281-2292.

He, K., Zhang, X., Ren, S., & Sun, J. (2016, June). Deep residual learning for image recognition. Paper presented at Conference on Computer Vision and Pattern Recognition organized by IEEE, USA, pp. 770-778.

Hollandi, R., Moshkov, N., Paavolainen, L., Tasnadi, E., Piccinini, F., & Horvath, P. (2022). Nucleus segmentation: Towards automated solutions. Trends in Cell Biology, vol. 32, no. 4, 295-310.

Hu, H., Zheng, Y., Zhou, Q., Xiao, J., Chen, S., & Guan, Q. (2019, November). MC-UNet: Multi-scale convolution UNet for bladder cancer cell segmentation in phase-contrast microscopy images. Paper presented at International Conference on Bioinformatics and Biomedicine organized by IEEE, USA, pp. 1197-1199.

Hu, T., Xu, S., Wei, L., Zhang, X., & Wang, X. (2021). CellTracker: An automated toolbox for single-cell segmentation and tracking of time-lapse microscopy images. Bioinformatics, vol. 37, no. 2, 285-287.

Huang, C., Ding, H., & Liu, C. (2020). Segmentation of cell images based on improved deep learning approach. IEEE Access, vol. 8, 110189-110202.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017, July). Densely connected convolutional networks. Paper presented at Conference on Computer Vision and Pattern Recognition organized by IEEE, USA, pp. 2261-2269.

Huang, J., Yang, G., Li, B., He, Y., & Liang, Y. (2021). Segmentation of cervical cell images based on generative adversarial networks. IEEE Access, vol. 9, 115415-115428.

Ibtehaz, N., & Rahman, M. S. (2020). MultiResUNet: Rethinking the U-Net architecture for multimodal biomedical image segmentation. Neural Networks, vol. 121, 74-87.

Iqbal, A., Sharif, M., Khan, M. A., et al. (2022). FF-UNet: A U-shaped deep convolutional neural network for multimodal biomedical image segmentation. Cognitive Computation, vol. 14, no. 7, 1287-1302.

Jia, D., Li, Z., & Zhang, C. (2020). A parametric optimization oriented, AFSA based random forest algorithm: Application to the detection of cervical epithelial cells. IEEE Access, vol. 8, 64891-64905.

Kachi, K., Iwahori, Y., Usami, H., Fukui, S., Wang, A., & Bhuyan, M. K. (2020, September). Pore detection from human skin image using U-Net. Paper presented at International Congress on Advanced Applied Informatics organized by International Institute of Applied Informatics, Japan, pp. 463-466.

Kane, K. I. W., Moreno, E. L., Hachi, S., Walter, M., Jarazo, J., Oliveira, M. A. P., Hankemeier, T., Vulto, P., Schwamborn, J. C., Thoma, M., & Fleming, R. M. T. (2019). Automated microfluidic cell culture of stem cell-derived dopaminergic neurons. Scientific Reports, vol. 9, no. 1, 1796.

Kaur, A., Midha, S., Giri, S., & Mohanty, S. (2019). Functional skin grafts: Where biomaterials meet stem cells. Stem Cells International, 1286054.

Khan, A., Kim, H., & Chua, L. (2021). PMED-Net: Pyramid based multi-scale encoder-decoder network for medical image segmentation. IEEE Access, vol. 9, 55988-55998.

Khan, M. Z., Gajendran, M. K., Lee, Y., & Khan, M. A. (2021). [Deep neural architectures for medical image semantic segmentation: Review]. IEEE Access, vol. 9, 83002-83024.

Kist, A. M., & Döllinger, M. (2020). Efficient biomedical image segmentation on EdgeTPUs at point of care. IEEE Access, vol. 8, 139356-139366.

Konishi, K., Mimura, M., Nonaka, T., Sase, I., Nishioka, H., & Suga, M. (2019). Practical method of cell segmentation in electron microscope image stack using deep convolutional neural network. Microscopy (Oxford), vol. 68, no. 4, 338-341.

Konishi, S., Hashimoto, T., Nakabuchi, T., Ozeki, T., & Kajita, H. (2021). Cell and tissue system capable of automated culture, stimulation, and monitor with the aim of feedback control of organs-on-a-chip. Scientific Reports, vol. 11, no. 1, 2999.

Korzynska, A., Roszkowiak, L., Zak, J., & Siemion, K. (2021). [A review of current systems for annotation of cell and tissue images in digital pathology]. Biocybernetics and Biomedical Engineering, vol. 41, no. 4, 1436-1453.

Kumar, N., Verma, R., Anand, D., Zhou, Y., Onder, O. F., Tsougenis, E., Chen, H., Heng, P. A., Li, J., Hu, Z., Wang, Y., Koohbanani, N. A., Jahanifar, M., Tajeddin, N. Z., Gooya, A., Rajpoot, N., Ren, X., Zhou, S., Wang, Q., … Sethi, A. (2020). A multi-organ nucleus segmentation challenge. IEEE Transactions on Medical Imaging, vol. 39, no. 5, 1380-1391.

Lei, T., Wang, R., Zhang, Y., Wan, Y., Liu, C., & Nandi, A. K. (2022). DefED-Net: deformable encoder-decoder network for liver and liver tumor segmentation. IEEE Transactions on Radiation and Plasma Medical Sciences, vol. 6, no. 1, 68-78.

Li, L., Dissanayake, B., Omotezako, T., Zhong, Y., Zhang, Q., Cai, R., Zheng, Q., Sng, D., Lin, W., Wang, Y., & Kot, A. C. (2023). Evaluating the efficacy of skincare product: A realistic short-term facial pore simulation. Electronic Imaging, vol. 35, no. 7, 276-1-276–6.

Lo Vercio, L. D., Green, R. M., Robertson, S., Guo, S., Dauter, A., Marchini, M., Vidal-García, M., Zhao, X., Mahika, A., Marcucio, R. S., Hallgrímsson, B., & Forkert, N. D. (2022). Segmentation of tissues and proliferating cells in light-sheet microscopy images of mouse embryos using convolutional neural networks. IEEE Access, vol. 10, 105084-105100.

Loewke, N. O., Pai, S., Cordeiro, C., Black, D., King, B. L., Contag, C. H., Chen, B., Baer, T. M., & Solgaard, O. (2018). Automated cell segmentation for quantitative phase microscopy. IEEE Transactions on Medical Imaging, vol. 37, no. 4, 929-940.

Lugagne, J. B., Lin, H., & Dunlop, M. J. (2020). DeLTA: Automated cell segmentation, tracking, and lineage reconstruction using deep learning. PLoS Computational Biology, vol. 16, no. 4, e1007673.

Malik, H., Idris, A. S., Toha, S. F., Idris, I. M., Daud, M. F., & Azmi, N. L. (2023a). A review of open-source image analysis tools for mammalian cell culture: algorithms, features and implementations. PeerJ Computer Science, vol. 9, e1364.

Malik, H., Idris, A. S., Toha, S. F., Idris, I. M., Daud, M. F., & Tokhi, M. O. (2023b). Deploying patch-Based segmentation pipeline for fibroblast cell images at varying magnifications. IEEE Access, vol. 11, 98171-98181.

Matsuzaka, Y., Kusakawa, S., Uesawa, Y., Sato, Y., & Satoh, M. (2021). Deep learning-based in vitro detection method for cellular impurities in human cell-processed therapeutic products. Applied Sciences, vol. 11, no. 20, 9755.

Minh, T. N., Sinn, M., Lam, H. T., & Wistuba, M. (2018). Automated image data preprocessing with deep reinforcement learning. ArXiv.

Mota, S. M., Rogers, R. E., Haskell, A. W., McNeill, E. P., Kaunas, R., Gregory, C. A., Giger, M. L., Maitland, K. C. (2021). Automated mesenchymal stem cell segmentation and machine learning-based phenotype classification using morphometric and textural analysis. Journal of Medical Imaging (Bellingham), vol. 8, no. 1, 014503.

Naidu, S., Quadros, A., Natekar, A., Parvatkar, P., Kumar, K., & Aswale, S. (2021). Medical image enhancement based on statistical and image processing techniques. International Journal of Engineering Research and Technology, vol. 10, 509-515.

Ng, H. P., Ong, S. H., Foong, K. W. C., Goh, P. S., & Nowinski, W. L. (2006, March). Medical image segmentation using k-means clustering and improved watershed algorithm. Paper presented at Southwest Symposium on Image Analysis and Interpretation organized by IEEE, USA, pp. 61-65.

Nofallah, S., Li, B., Mokhtari, M., Wu, W., Knezevich, S., May, C. J., Chang, O. H., Elmore, J. G., & Shapiro, L. G. (2022). Improving the diagnosis of skin biopsies using tissue segmentation. Diagnostics (Basel), vol. 12, no. 7, 1713.

O' Connor, O. M., Alnahhas, R. N., Lugagne, J. B., & Dunlop, M. J. (2022). DeLTA 2.0: A deep learning pipeline for quantifying single-cell spatial and temporal dynamics. PLoS Computational Biology, vol. 18, no. 1, e1009797.

O' Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Velasco-Hernández, G. A., Krpalkova, L., Riordan, D., & Walsh, J. (2019). Deep learning vs. traditional computer vision. ArXiv.

Oktay, O., Schlemper, J., Le Folgoc, L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N. Y., Kainz, B., Glocker, B., & Rueckert, D. (2018). Attention U-Net: Learning where to look for the pancreas. ArXiv.

Oskal, K. R. J., Risdal, M., Janssen, E. A. M., Undersrud, E. S., & Gulsrud, T. O. (2019). A U-Net based approach to epidermal tissue segmentation in whole slide histopathological images. SN Applied Sciences, vol. 1, no. 7, 672.

Pachitariu, M., & Stringer, C. (2022). Cellpose 2.0: How to train your own model. Nature Methods, vol. 19, no. 12, 1634-1641.

Pérez, P., Huertas, G., Olmo, A., Maldonado-Jacobi, A., Serrano, J. A., Martín, M. E., Daza, P., & Yúfera, A. (2018). Remote cell growth sensing using self-sustained bio-oscillations. Sensors, vol. 18, no. 8, 2550.

Punn, N. S., & Agarwal, S. (2022). [Modality specific U-Net variants for biomedical image segmentation: A survey]. Artificial Intelligence Review, vol. 55, no. 7, 5845–5889.

Ramachandram, D., Ramirez-GarciaLuna, J. L., Fraser, R. D. J., Martínez-Jiménez, M. A., Arriaga-Caballero, J. E., & Allport, J. (2022). Fully automated wound tissue segmentation using deep learning on mobile devices: Cohort study. JMIR Mhealth Uhealth, vol. 10, no. 4, e36977.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. ArXiv.

Scherr, T., Seiffarth, J., Wollenhaupt, B., Neumann, O., Schilling, M. P., Kohlheyer, D., Scharr, H., Nöh, K., & Mikut, R. (2022). microbeSEG: A deep learning software tool with OMERO data management for efficient and accurate cell segmentation. PLoS One, vol. 17, no. 11, e0277601.

Schlemper, J., Oktay, O., Schaap, M., Heinrich, M., Kainz, B., Glocker, B., & Rueckert, D. (2019). Attention gated networks: Learning to leverage salient regions in medical images. Medical Image Analysis, vol. 53, 197-207.

Sekou, T. B., Hidane, M., Olivier, J., & Cardot, H. (2019). From patch to image segmentation using fully convolutional networks - Application to retinal images. ArXiv.

Shamhan, M., Idris, A. S., Toha, S. F., Daud, M. F., Idris, I. M., & Malik, H. (2023). An automated approach for fibroblast cell confluency characterisation and sample handling using AIoT for bio-research and bio-manufacturing. Cogent Engineering, vol. 10, no. 1.

Shen, T., & Wang, Y. (2018, October). Medical image segmentation based on improved watershed algorithm. Paper presented at Conference on Advanced Information Technology, Electronic and Automation Control organized by IEEE, China, pp. 1695-1698.

Siddique, N., Paheding, S., Elkin, C. P., & Devabhaktuni, V. (2021). [U-Net and its variants for medical image segmentation: A review of theory and applications]. IEEE Access, vol. 9, 82031-82057.

Su, R., Zhang, D., Liu, J., & Cheng, C. (2021). MSU-Net: Multi-scale U-Net for 2D medical image segmentation. Frontiers in Genetics, vol. 12, 639930.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015, June). Going deeper with convolutions. Paper presented at Conference on Computer Vision and Pattern Recognition organized by IEEE, USA, pp. 1-9.

Tong, X., Wei, J., Sun, B., Su, S., Zuo, Z., & Wu, P. (2021). ASCU-Net: Attention gate, spatial and channel attention U-Net for skin lesion segmentation. Diagnostics, vol. 11, no. 3, 501.

Vaze, S., Xie, W., & Namburete, A. I. L. (2020). Low-memory CNNs enabling real-time ultrasound segmentation towards mobile deployment. IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 4, 1059-1069.

Voitiuk, K., Geng, J., Keefe, M. G., Parks, D. F., Sanso, S. E., Hawthorne, N., Freeman, D. B., Currie, R., Mostajo-Radji, M. A., Pollen, A. A., Nowakowski, T. J., Salama, S. R., Teodorescu, M., & Haussler, D. (2021). Light-weight electrophysiology hardware and software platform for cloud-based neural recording experiments. Journal of Neural Engineering, vol. 18, no. 6, 066004.

Wang, C., Anisuzzaman, D. M., Williamson, V., Dhar, M. K., Rostami, B., Niezgoda, J., Gopalakrishnan, S., & Yu, Z. (2020). Fully automatic wound segmentation with deep convolutional neural networks. Scientific Reports, vol. 10, no. 1, 21897.

Wang, S., Wang, X., Shen, Y., He, B., Zhao, X., Cheung, P. W.-H., Cheung, J. P. Y., Luk, K. D.-K., & Hu, Y. (2022). An ensemble-based densely-connected deep learning system for assessment of skeletal maturity. IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 52, no. 1, 426-437.

Weigert, M., Schmidt, U., Boothe, T., Müller, A., Dibrov, A., Jain, A., Wilhelm, B., Schmidt, D., Broaddus, C., Culley, S., Rocha-Martins, M., Segovia-Miranda, F., Norden, C., Henriques, R., Zerial, M., Solimena, M., Rink, J., Tomancak, P., Royer, L., … Myers, E. W. (2018). Content-aware image restoration: pushing the limits of fluorescence microscopy. Nature Methods, vol. 15, no. 12, 1090-1097.

Wu, W., Mehta, S., Nofallah, S., Knezevich, S., May, C. J., Chang, O. H., Elmore, J. G., & Shapiro, L. G. (2021). Scale-aware transformers for diagnosing melanocytic lesions. IEEE Access, vol. 9, 163526-163541.

Zargari, A., Lodewijk, G. A., Neudorf, C. W., Araghbidikashani, K., Mashhadi, N., Rubio, S., Hinck, L., & Shariati, S. A. (2021). DeepSea: An efficient deep learning model for automated cell segmentation and tracking. bioRxiv.

Zhang, M., Li, X., Xu, M., & Li, Q. (2020). Automated semantic segmentation of red blood cells for sickle cell disease. IEEE Journal of Biomedical and Health Informatics, vol. 24, no. 11, 3095-3102.

Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., & Liang, J. (2018). UNet++: A nested U-Net architecture for medical image segmentation. Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision, vol. 11045, 3-11.

Zuiderveld, K. (1994). Contrast limited adaptive histogram equalization. In P. S. Heckbert (Ed.). *Graphics Gems IV*. Academic Press Professional, Inc.

APPENDIX A

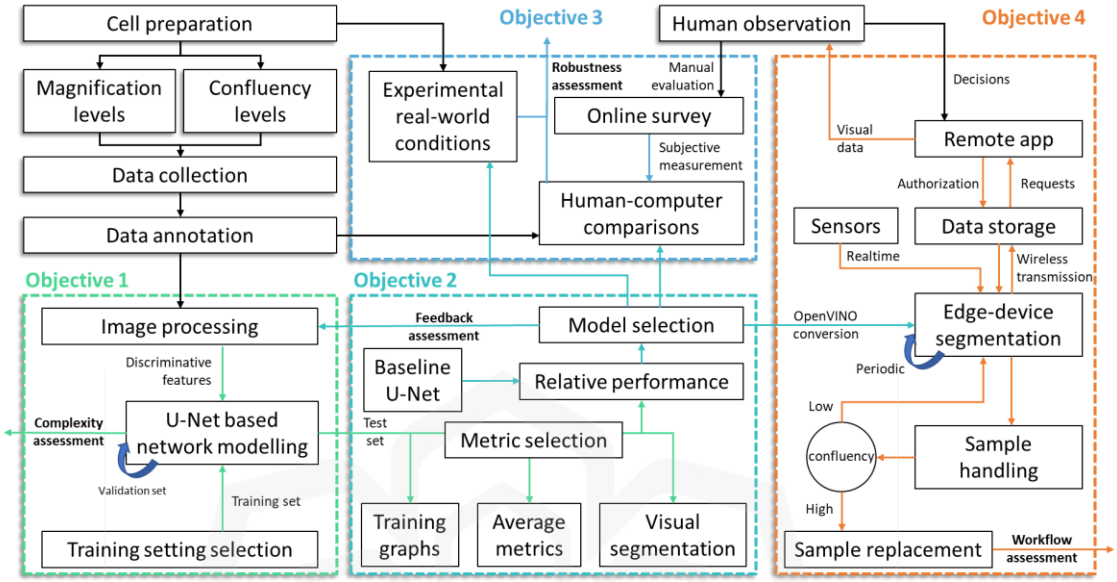


Figure 1 Research Framework

APPENDIX B

Your age group?	What is your current position?	Years of experience in cell culture?	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8
31 - 40	Postgraduate student	5 - 10 years	20%	10%	20-30%	50%	70%	40-50%	30-40	30%
21 - 30	Postgraduate student	2 - 5 years	40%	20%	10%	60%	60%	30%	40%	30%
21 - 30	Researcher/post-doc (e.g. science officer, research officer)	2 - 5 years	20%	10%	5%	30%	80%	10%	20%	40%
21 - 30	Postgraduate student	2 - 5 years	30	15	15	40	80	20	25	30
31 - 40	Postgraduate student	2 - 5 years	40%	10%	10%	40%	70%	40%	30%	40%
31 - 40	Academician (lecturer, professor, etc)	5 - 10 years	30	30	20	60	80	40	30	50
31 - 40	Academician (lecturer, professor, etc)	> 10 years	20%	10%	10%	40-50%	50-60%	20%	30%	30%
21 - 30	Undergraduate student	0 - 2 years	30%	10%	10%	30%	25%	10%	15%	10%
31 - 40	Academician (lecturer, professor, etc)	5 - 10 years	30	10	10	50	60	20	40	30
31 - 40	Academician (lecturer, professor, etc)	> 10 years	60	30	30	50	75	30	45	50

Figure 1 Survey Raw Data