INTEGRATING SEGMENTATION SOFTWARE AND BLIP-2 FOR AUTOMATED CELLULAR DATA ANALYSIS IN A WEB-BASED APPLICATION

by

Chyi Ricketts

Signature Work, in partial fulfillment of the
Duke Kunshan University Undergraduate Degree Program

March 9th, 2025

Signature Work Program
Duke Kunshan University

APPROVALS

Mentor: Linfeng Huang, Division of Natural and Applied Sciences

---

CONTENTS

Abstract	ii
Acknowledgements	iii
List of Figures	iv
List of Tables	v
1 Introduction	1
2 Material and Methods	11
3 Results	14
4 Discussion	26
5 Conclusion	28
Narrative	29
References	32
Additional Material	36
Appendix A	36
Appendix B	37
Appendix C	37

ABSTRACT

This study focuses on optimizing the efficiency of image analysis in scientific research through three main deliverables: an assessment of existing cellular segmentation software, an original web-based application for segmentation visualization, and a fine-tuned vision language model, BLIP-2. Several segmentation software were assessed through a benchmark analysis of segmentation accuracy following a literature review and a qualitative analysis to ultimately select Cellpose for integration with the web-based application. This user-friendly web graphical user interface (GUI) responds to user input to view image files by selecting pre-identified regions of interest (ROI) and corresponding extraction of features. This study establishes a recommended pipeline for segmenting masks with Cellpose for pixel-wise, accurate segmentation and highlighting ROIs with tools provided by the web-application. The fine-tuned BLIP-2 model utilizes its multimodal translational capabilities to enable subsequent automated captioning on a large set of images. This transformation into language data will allow any large language model (LLM) to identify patterns within the image set. This comprehensive pipeline traces the transformation of raw image data into processed language formats, leveraging the strengths of deep learning algorithms. This will provide biological research scientists with an efficient method for identifying trends in complex datasets without the need of unpacking data with complex software coding.

Keywords:

Cellular Segmentation, Deep Learning, Data Visualization, Image Analysis, Web Application

ACKNOWLEDGEMENTS

Special thanks to Professor Linfeng Huang and the Wang Cai Biochemistry Lab at DKU for their mentoring and resources throughout this project. Also, to Priyom Adhyapok for her guidance and inspiration for this project during my time at the Bagnat Lab and Di Talia Lab. Lastly, thank you to the Student Research Scholars Program and DKU Undergraduate Studies.

LIST OF FIGURES

Figure 1: Thresholding and Watershed 3

Figure 2: Image Pre-Processing with Gaussian Blur 4

Figure 3: Shallow and Deep Learning Annotations 6

Figure 4: Comparison of Segmentation Results 17

Figure 5: Manual Feature Selection in Ilastik 18

Figure 6: Diameters as a Varied Parameter in Cellpose 19

Figure 7: Impact of Fine-tune Training in Cellpose 19

Figure 8: Visualization Tool in Jupyter Notebook 21

Figure 9: User-input Web Application 22

Figure 10: Image Analysis Pipeline Flowchart 24

LIST OF TABLES

Table 1: Qualitative Cell Segmentation Software Comparison 14
Table 2: F1 Scoring on Selected Models 16

Chapter 1

INTRODUCTION

Most people rely heavily on their vision to interact with and understand the world. In scientific research, complex data can be made more accessible and intuitive by visual representation. In this way, it can improve comprehension and serve as a powerful tool that allows reasoning about processes by observing their occurrence or effects. In biological research, for example, where many processes are occurring at a molecular level, visualization becomes crucial. One of the most foundational forms of visual data acquisition in biological research is the use of biomarkers and microscopy to illuminate targeted regions within a sample[1]. A wide range of techniques for biomarkers, including fluorescent dyes, fluorescent probes[2], or fluorophores, can be directly stained onto the sample, selectively bound through the use of antibody-antigen interactions, or have been genetically modified to be produced by the cell itself [3]. This highly versatile tool can be applied anywhere from fundamental biological processes to specific interactions within cellular dynamics to personalized medicine and more.

Microscopy and Computational Techniques

With the widespread use of biomarkers comes the necessity to develop microscopy technology to visualize increasingly complex images, and without fail, microscopy has advanced at an incredible rate within the past few decades, enabling accessible high-resolution imaging and live tracking of markers within living cells[4]. Additionally, the range of available microscopy options has expanded. For example, the visual characteristics of the same sample can vary significantly when prepared and imaged through a confocal microscope versus a transmission electron microscope (TEM).

As the technology for biological research has rapidly developed, today's research scientists can easily obtain thousands of high-quality images within a few hours using automated screening machines, some even utilizing AI in the process. Extracting these complex and rich datasets has become extremely time-consuming, requiring experts to look through each image, either

annotating the individual elements or picking out trends to understand the significance of an overflow of visual data. To relieve the bottleneck that has fallen onto data processing, the number of automated image analysis pipelines has surged dramatically, releasing programs and software from simple image editing to deep learning models requiring dedicated GPUs. Today, bioinformatics and computational biology, the study of creating algorithms and software to process high amounts of nucleotide, protein, visual, and other data, form their own field within the larger context of scientific research.

Introduction to Cellular Segmentation

While computational techniques in biology encompass a wide range, cellular segmentation retains a crucial role in image analysis. Cellular segmentation, a facet of image segmentation, is the process of sorting pixels into various groups, commonly referred to as regions of interest (ROIs) in scientific terminology[5]. This technique is currently seen being used in various fields such as self-driving vehicle vision[6] and medical imaging analysis[7]. The first hurdle to overcome when segmenting any image with computational techniques for any field is to allow the computer to identify the ROI[8]. While it may sound simple, especially with a trained eye, the process of partitioning a digital image into individually labeled pixels to differentiate objects, boundaries, and background noise can be tricky, as it is a problem that is still being refined today. This process, known as segmentation, allows the computer to isolate instances of an object to analyze, for instance, the size, shape, or other identifying characteristics. In recent decades, cellular segmentation has evolved drastically from human annotation to simple techniques, and more recently, to machine learning; in just the past few years, deep learning has continued to transform the field. The following literature review will provide a comprehensive background on the growth of cellular segmentation over the years and its consistent contribution to bioimage analysis.

Classical Segmentation Algorithms

The most fundamental method of automatic segmentation, thresholding, involves separating pixels into foreground and background by intensity. Figure 1 shows the basic application of Otsu's method[9] and the Watershed algorithm[10,11]. Otsu's method is a global thresholding method that calculates the optimal single-intensity threshold to separate pixels into either foreground or background. It uses an equation that exhaustively searches for the threshold that minimizes the intra-class variance. This method works best with spaced-out data of a bimodal nature, visualized as a histogram with two intense peaks, to isolate ROIs. Plenty of methods rely on this determined threshold for further processing. For example, combining Otsu's method with the Watershed algorithm can take the binary segmentation results from Otsu's method and classify the pixels of individual cells into separate classes, significantly improving results on clustered cells or cells with overlapping borders. The Watershed algorithm, named after its function of treating an image as a topographical map of intensity and 'flooding' the area, can separate the boundaries between cells by splitting the local maxima. It is important to note that cells with overlapping borders are not able to be uniquely identified as individual cells with Otsu's method and seen as one 'conjoined' cell, shown with red arrows in Figure 1. While the Watershed algorithm can differentiate cells, it is highly reliant on data presenting a consistent gradient toward its borders and fails to properly segment cells with uncertain borders. Its computational speed and simplicity make it ideal for data like nuclei segmentation in fluorescence microscopy images.

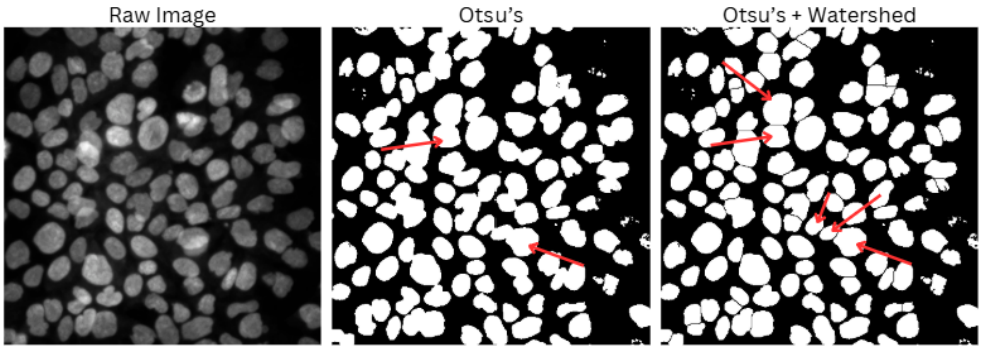


Figure 1: Comparison of Otu's method for thresholding and the watershed algorithm on hoechst 33342 (ThermoFisher, H3570) stained, clustered Hela cells. Red arrows indicate cells labeled as 1 ROI with Out's method and multiple ROIs with the Watershed Algorithm. Figure made with ImageJ

As with Otsu's method, the Watershed algorithm is not ideal for images with similar grey means with varying textures such as images with a large amount of noise. A wide variety of traditional image editing techniques can assist in reducing the impact of additional noise in images through pre-processing. ImageJ[12,13], also known as Fiji[14], has been a staple in scientific image analysis for decades since its development as NIH Image in 1987. By specializing in traditional forms of image analysis, it remains one of the most widely used tools, offering essential features such as pixel measurements, color scaling, and image cropping. These simple tools will always remain integral to image analysis pipelines along with various scientific operations for pre-processing of images, including Gaussian blur, Hessian matrix, edge smoothing, and a wide range of noise reduction and diffusion filtering techniques. These pre-processing techniques can assist in reducing the impact of additional noise in images to enhance the capabilities of algorithms such as Watershed. For example, Gaussian blur[15] is an image-blurring filter typically applied in two dimensions to produce a surface where the original points become contours of its immediate concentric circle. While it reduces noise, blurring also diminishes image details, which may hinder certain image processing tasks and feature extraction. Still, Gaussian blurs, when used in the appropriate context, have proved to increase the effectiveness of computer vision programs[14] and are commonly used as a pre-processing measure in many image analysis contexts, especially segmentation (Figure 2). ImageJ additionally offers easily accessible plug-ins for Otsu's and other methods of thresholding, the Watershed algorithm, and more complex segmentation techniques.

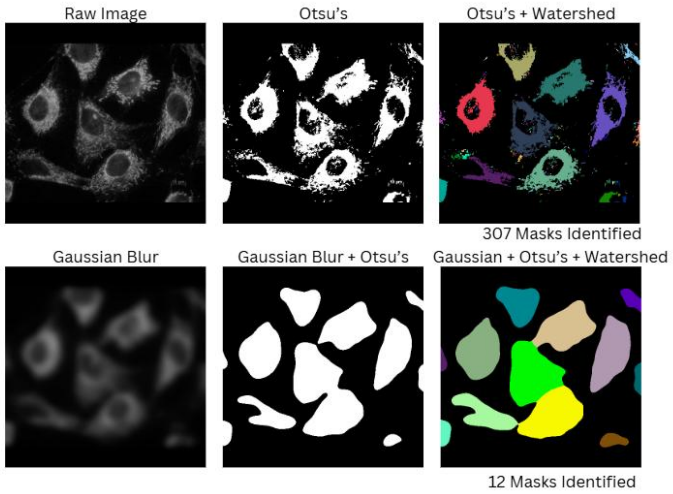


Figure 2: The effects of Gaussian blur on cellular segmentation on Fibroblasts stained with Alexa Fluor 594 and Alexa Fluor 488 (https://www.cellimagelibrary.org/images/248). Figure made with ImageJ

Even with pre-processing strategies, challenges of overfitting, variation, and time consumption challenge the efficiency of cellular segmentation and requires the development of more effective tools. Luckily, the scientific community has taken great strides to aid the advancement of image analysis by providing resources and opportunities to researchers worldwide. One of the most innovative efforts in the field is the Grand Challenges in Bioimage Analysis[16]. This initiative provides a dataset, typically with a set of ground truth values, released to the public to analyze with their own methods. The most effective methods are ranked and published. These types of events not only encourage participation and innovation but also allow for collaboration and transparency to learn from one another.

Machine Learning Cell Segmentation Software

The past couple of years have featured a domination of machine learning (ML) and deep learning (DL) based algorithms, which are proving to be increasingly more accurate and accessible. Artificial intelligence (AI) is a broad term that encompasses machine learning and deep learning. AI has started to flood our daily lives, changing elements from how we work to how we communicate. While colloquial uses of artificial intelligence are thrown around, it is often used interchangeably with machine learning and deep learning, causing confusion. Essentially, the umbrella term of AI refers to the ability of machines to mimic human learning and intelligence. Within AI, ML refers to the ability to analyze input data to recognize patterns, often optimizing the processes to replicate that output data. In this paper, deep learning is defined as a subset of machine learning that relies on neural networks to make decisions in a logical fashion.

ML dates back to the mid-20th century, but roots in practical algorithms only took hold in the 1980s and 1990s with developments such as support vector machines and ensemble methods. In 2001, a commonly used machine learning technique called Random Forest[17] had a great impact on cellular segmentation. It works by creating a multitude of random decision trees through training data. This data goes through randomized sampling of datasets, called bootstrapping, and randomizing selection of features, known as feature selection. These processes allow a smaller quantity of training data to be less sensitized to the training data and thereby generalized for testing data. This collection of random decision trees ultimately learns scattered portions of the training

data and stores data about its pattern to filter testing data and aggregate each tree's output to make decisions about its features, such as segmentation.

Illastik[18] is an example of software utilizing ML Random Forest for cellular segmentation. Instead of relying on neural networks, it builds an optimized path through bootstrapping and feature selection to predict the boundaries of an object. This method within Ilastik aims to alleviate some of the time-consuming processes of annotating truth data for deep learning algorithms by featuring interactive training of simply drawing on annotations over a portion of pixels within two or more different groupings, or 'shallow learning'[19,20]. When employing this training data, only labeled pixels are considered. At the same time, the rest are ignored, greatly contrasting the annotations required for deep learning truth values in which every pixel must belong to a given class (Figure 3). Random forest, as well as other shallow learning techniques, will randomize and build pattern pathways based off the limited training data to reverse engineer the optimal computational pathway to arrive at the corresponding segmentation.

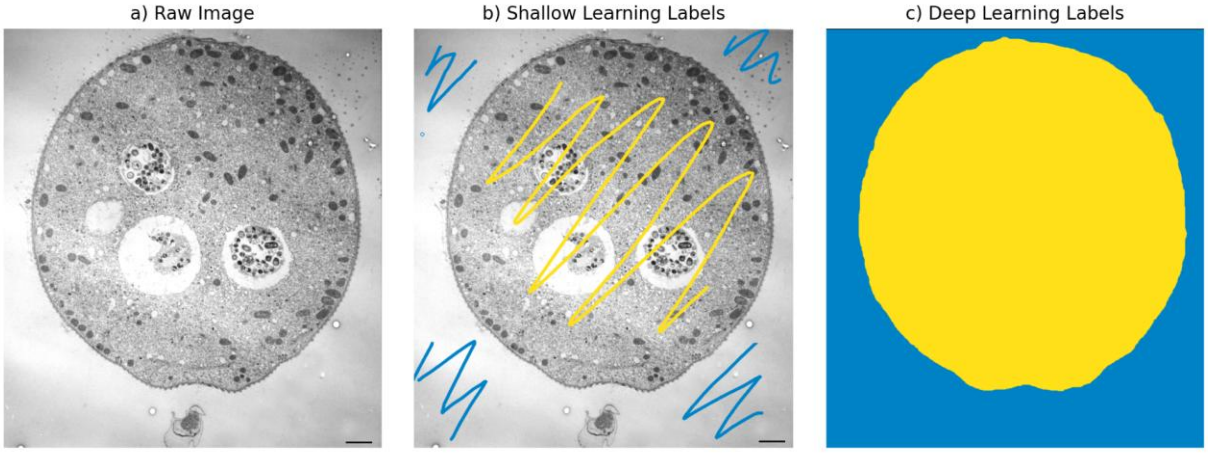


Figure 3: Illustrating the difference between shallow learning and deep learning labels
(a) Raw image, (b) Shallow learning labels prepared in Ilastik, (c) Deep learning labels prepared in Cellpose on a serially sectioned contracted Vorticella convallaria cell taken on a Transmission Electron Microscope (TEM) from the Public Domain of cellimagelibrary.org

Deep Learning Cell Segmentation Software

Deep learning neural networks come in basic building blocks of nodes, layers, and weights/biases to form a variety of neural network types, including the more well-known convolutional neural network (CNN), dense/fully connected neural networks (DNN), and many more. Combinations of neural network layers are optimized for different tasks and can vary by field. The use of neural networks generally outperforms ML techniques when set up with optimal structural architecture and training. Though, DL algorithms require a far larger and more precise set of data annotations, where every pixel is labeled with a group (Figure 3). This process can be extremely tedious and time-consuming as it requires the user to carefully trace the borders of hundreds or thousands of cells or nuclei before the model becomes usable. Typically, DL models require millions to billions of data points to learn complex patterns. For example, GPT-4 uses over 300 Billion text tokens, or snippets of text[21].

Several popular DL cellular segmentation software today revolves around a U-Net[22] framework such as Cellpose[23] and StarDist[24]. The release of U-Net in 2015 was monumental for the application of image processing. It featured a U-shaped encoder and decoder made fully with convolutional neural network (CNN) layers. U-Net uses an encoder that receives an image and reduces the information by half while doubling the channels for each layer to produce a decoded, informational version of the image. Afterward, it passes through a decoder that, inversely, reduces the channels by half while doubling the information passed by skip connections that link the decoder. It can produce significantly more accurate results with a smaller testing dataset and deal with more complex shapes than other methods[22]. For instance, Cellpose is only trained on around 70,000 segmented cells and only recommends an additional 250 labeled images to induce effective fine-tuning[23].

Cellpose was published as a generalist cellular segmentation algorithm with the aim of using its pre-trained models to segment cell boundaries and nuclei of all shapes and sizes. Using a Convolutional Neural Network (CNN) with a U-Net architecture, it can overcome images with more noise or poor lighting with the addition of pre-processing and other features. While the base models have been trained on a diverse range of images, it includes an interactive feature to solidify its accreditation as a generalist algorithm by allowing a researcher to expand the original training set to create models that are more specialized for their specific needs.

StarDist[24] took a very interesting approach to the difficulty of recognizing all cells within the wide range of microscopy images. It focuses on segmentation with the basis of star-convex polygons, meaning a shape where all the edges may be 'seen' or connected linearly to any point within the shape. This blob-like shape restriction includes most, if not all, cell shapes, which help filter out noise or unrelated artifacts. StarDist outputs a distance transform map and a probability map for each pixel. When encountering overlapping cell masks, the outcome with the highest probability is kept, greatly increasing the program's ability to identify adjacent cells accurately. StarDist operates as a python package that can be imported, and its model is designed to be directly used through python IDEs (Integrated Development Environment). Many have recognized the difficulty of requiring biological scientists to integrate coding into their image analysis pathways since most have little expertise in that field. To account for this, several graphical user interfaces (GUIs) have been released to provide platforms to access these computationally heavy software. For example, QuPath[25] provides a platform for several of the aforementioned software and specializes in histology samples. Other platforms, including Icy[26] and CellProfiler[27], often integrate other useful tools, such as cell tracking and batch processing, in addition to the integration of StarDist and Cellpose models. These platforms can be extremely useful in increasing the accessibility of computational methods and are being continuously updated with collaborative plugins.

Unlike Cellpose and StarDist, DeepCell[28] is a web-based interface, allowing researchers to access it more easily without downloading heavy software. DeepCell focuses on multi-class segmentation and excels at segmenting dense, clustered cells in both fluorescence images by using a modified watershed algorithm called DeepWatershed to create and learn from boundary probability maps. The difficulty of these three deep learning programs is that they require a certain amount of coding knowledge to either use the interface or process the data it outputs.

Additional Segmentation and DL Models

Segmentation is applied in many fields outside of biological data and there are many alternative platforms for general segmentation. Segment Anything[30] (SAM) is a foundation model for image segmentation developed by Meta AI. As its name suggests, SAM is designed to segment all sorts of images without task-specific fine-tuning, known as zero-shot generalization. Unlike the cellular segmentation specialized models, SAM accepts user prompts such as points and bounding boxes to guide segmentation. SAM's strength comes from combining deep learning techniques, including a vision transformer (ViT) and multi-layer perceptron (MLPs), to understand spatial relationships within images and convert user inputs into embeddable information. Its adaptability makes it applicable as a base model to many fields requiring image analysis, such as medical imagining analysis, cellular segmentation, object tracking in self-driving vehicles, and more.

For more general ML and DL, a company called Hugging Face provides a repository of thousands of open-source models, including Random Forest, GPT-based, BERT-based, Vision Transformers (ViT), SAM ,and many more. It works directly through a Python library, which can be easily accessed through any integrated development environment (IDE). Additionally, they provide tools that can be directly used to train and fine-tune their models with any compatible data.

Challenges with DL Segmentation

As more labs are adopting DL software into their image processing pipelines, they face challenges in balancing the time required for annotating training data and customizing parameters with potential accuracy. With image analysis workflows becoming increasingly complex, research scientists without a software engineering background find it increasingly difficult to understand how their data is processed and interpreted even with GUIs. These programs typically offer custom calibration of hundreds of parameters for each image, most of which are difficult to grasp the direct consequences of. Even more so, many parameters are hidden within the algorithm itself, allowing the computer to automate decision-making through convolutional filters. These models also require meticulous hand annotation to provide a large set of training data with predetermined classes of every pixel. This ground truth data must be expertly annotated, requiring huge amounts of time before a model dissects your labels and can accurately segment your targeted specimen.

Additionally, a challenge that often arises when training DL models is the risk of overfitting, especially when specialized image types dominate the labeled data. Overfitting in DL models occurs when a model performs well on the training data but poorly on unseen data as a direct result of manually labeled ground truth values being highly saturated with specific image types. The models will memorize patterns in the training data that may not be immediately apparent, essentially leading the model to learn shortcuts rather than genuine patterns. To avoid this, the model should be trained on a varied set of data.

Other publicly accessible resources that follow a similar sentiment to Grand Challenges aid in these challenges. Bioimage.io[32] is a community-driven platform designed for sharing and accessing pre-trained deep-learning models for bioimage analysis on new and existing platforms. Sharing pre-trained models is an invaluable resource that can increase the accuracy of output data and save valuable time. This method, also known as transfer learning, is important for the cumulative learning of deep learning models.

Study Motivation

This project aims to streamline the process of segmentation and post-segmentation analysis. First, by analyzing the strengths of existing cellular segmentation programs through a quantitative benchmark analysis. Second, by creating an external web application to more efficiently view and interpret complex image data without requiring additional software coding skills. Ultimately, the goal is to provide a more accessible and time-efficient method for scientists to extract meaningful insights from large-scale image datasets, thereby alleviating the strain of tedious research processes.

Chapter 2

MATERIALS AND METHODS

Training and Testing Data Cell Lines

HeLa cells were cultured in MEM medium (CELL RESEARCH, ZQ-300) and incubated at 37 degrees Celsius in a 5% CO2 humidified incubator.

Fluorescent Staining

Cells were stained with a combination of fluorescent dyes, including Hoechst 33342 (ThermoFisher, H3570), Calcein AM, Propidium Iodide (PI), BODIPY, WGA (Wheat Germ Agglutinin), Phalloidin, Concanavalin A (ConA), and SYTO Dyes.

Group 1: hoechst, calcein AM, PI
Groups 2 and 3: hoechst, BODIPY
Group 4: hoechst, WGA, Phalloidin, conA
Group 5: hoechst, WGA, Phalloidin, SYTO

Microscopy

Experimental groups were imaged on 96-well plates with a CellInsight CX7 High Content Screening (HCS) platform (ThermoFisher Scientific)

F1 Scoring

F1 Scoring, or the Dice-Sørensen coefficient, is a statistic that is commonly used in gauging the similarity of two samples by the following equation:

$$DSC = \frac{2TP}{2TP + FP + FN}$$

Where TP, FP, and FN are the true positives, false positives, and false negatives, respectively. F1 scoring is utilized to measure the overlap between the truth and test mask sets using their respective pixel data sets.

Otsu's Method of Thresholding

Otsu's method of thresholding is calculated by exhaustively searching for the optimal threshold for an image by minimizing intra-class variance using the following equation:

$$\sigma_{\omega}^{2}(t) = \omega_{0}(t)\sigma_{0}^{2}(t) + \omega_{1}(t)\sigma_{1}^{2}(t)$$

Where weights w_0 and w_1 are the probabilities of the two classes separated by a threshold, t, and σ_0^2 and σ_1^2 represent the variance of the two classes. The calculated threshold is essentially the greatest weighted sum of variances of the two classes.

Cellpose

Cellpose Version 2.0 used with Python 3.10

The base model used for base model segmentation and for self-trained model base was 'nuclei'

Selected channel was [0,0], segmenting based on the grayscale image

Diameters were automatically selected when training the image unless otherwise noted.

StarDist

The StarDist version used was 2D versatile (H&E nuclei) that was trained on images from the MoNuSeg 2018 training data and the TNBC dataset from Navlor et al. (2018).

Ilastik

The Illastik software version used was ilastik-1.4.1rc2

Each batch of images underwent one training regiment with Pixel Classification with Shallow Training Labels and automatic feature selection amongst available options with Color/Intensity, Edge, and Texture. The remaining images were processed in Batch Processing and exported as a numpy file containing Simple Segmentation data.

Segment Anything (SAM)

The SegmentAnything model type: vit_h

SAM Automatic Mask Generator and SAM Predictor

Albumentations

Python package containing built-in augmentations for horizontal flip, vertical flip, random rotation, brightness and contrast adjustment, color jitter, elastic transformation, grid distortion, gaussian noise, gaussian blur, and CLACHE (local contrast)

Other packages used:

Numpy, Matplotlib, Flask, Render

BLIP-2

BLIP-2 (Bootstrapping Language-Image Pretraining) is a vision language model
The model is used through HuggingFace and trained with Parameter Efficient Fine-Tuning (PEFT) techniques using the following HuggingFace libraries: transformers, datasets, peft, and bitsandbytes[42]

Chapter 3

RESULTS

A literature review was first conducted on several of the available software and evaluated based on qualitative analysis to narrow down the selection for a deeper quantitative benchmarking. In Table 1, the most popular software are summarized based on a literature review and simple analysis. Categorizing the software by year released, algorithm type (traditional, ML, and DL), inclusion of pre-trained models and self-trained models, and features.

Table 1. Qualitative assessment of 8 available segmentation platforms

Software	Year	Algorithm Type	Open Source	Pre-trained Models	Self-trained Models	Features
Fiji/ImageJ	1987	Traditional (thresholding, watershed, etc.)	Yes	No **	No **	Classical segmentation with plugins
CellProfiler	2005	Traditional and ML	Yes	No **	Yes **	High-throughput, batch processing, and modular workflows
Ilastik	2011	ML (Random Forest)	Yes	No **	Yes	Interactive pixel classification, object segmentation
Aivia	2015	DL with AI assistance	No	Yes	Yes	Advanced 3D segmentation, tracking, visualization
DeepCell	2017	DL using U-Net (CNN-based)	Yes	Yes	Yes	Pretrained models for nuclear/cell segmentation
StarDist	2018	DL using U-Net (Star-convex polygons)	Yes	Yes **	Yes **	Optimized for star-convex objects
Cellpose	2020	DL using U-Net (flow-based CNN)	Yes	Yes	Yes	Pretrained models, scalable for different cell types
Segment Anything	2023	ViT with Promptable DL	Yes	Yes	No	Foundation model for general object segmentation

** stars denote that it is only or additionally available through plugins or other linkable software

The available cellular segmentation software ranges from traditional segmentation to ML and DL techniques. The specific types of algorithms used by each software are detailed in Table 1, but many also offer additional tools in the form of wrappers and plug-ins that keep them relevant and applicable to a wide range of research. Many of these plug-ins are developed with the help of independent developers and contributors who expand on existing software by adding desired features, a process made possible by open-source accessibility. These developments expand on the available bank of tools for all researchers which aids in the general advancement of the field.

Amongst the software analyzed, only Aivia is closed source since it is only available as a commercial product. According to its description, it utilizes AI, including deep learning-based segmentation, to track and segment objects with a machine learning pipeline, but the exact algorithms are not disclosed.

Accuracy Benchmarking of Segmentation Software

When selecting software to move into quantitative analysis, a diversity of algorithms was considered, along with differences in the availability of pre- and self- trained models and specialized features. The three selected software, Ilastik, Cellpose, and SAM, represent ML techniques, DL CNNs optimized for generalization of cell images, and a zero-shot generalization model built for any image.

A qualitative benchmark analysis on segmentation accuracy was carried out using two sets of 49 images, groups 2 and 4, with varying combinations of fluorescent staining (see Methods for more detail). Ilastik does not provide pre-trained nuclei models; therefore, the segmented cells were analyzed with Ilastik's pixel classification workflow using automatic feature selection amongst available options with Color/Intensity, Edge, and Texture. Each of the two batches of images was trained on the first image of the set, with the rest processed with batch selection. Pre-trained models were used for the quantification of Cellpose. Specifically, Cellpose's 'nuclei' model was used directly for the pre-trained model and used as a base for the fine-tuned model. The fine-tuned model was trained with an additional two sets of 49 images, groups 3 and 5 (see Methods for more detail). Parameters were set to a general default with diameters automatically estimated, channels as greyscale, flow threshold, cell probability threshold, and minimum size as

factory settings. Conversely, SAM is a base model for segmentation and was not trained specifically for these images, only the pre-train vit-h model was used with SAM's Automatic Mask Generator. Table 2 summarizes the data for the quantitative comparison of these three software: Ilastik, SAM, and Cellpose.

Table 2: F1 scoring on pre-trained models and self-trained models for 3 software

	Ilastik	**Cellpose**	**SAM**
Algorithm Type	ML (Random Forest)	DL using UNet (flow-based CNN)	ViT with Promptable DL
Ease of Use	Easy	Medium	Difficult
GUI	Yes	Yes	No
F1 score with pre-trained model	N/A	**0.67826**	**0.7379**
F1 score with fine-tuned model	**0.7708**	**0.8773**	N/A

Ilastik version 1.4.1rc2 using pixel classification with automatic feature selection of 7 features
Cellpose version 2.0 using base model "nuclei" and training on 96 images from Groups 3 and 4 (methods)
SAM version vit_h using SamAutomaticMaskGenerator package for whole image segmentation

The F1 score is calculated using pixel-wise data, comparing the hand-annotated truth masks to the predicted masks of each software. Figure 4 shows a preview of one of the images annotated across each of the three software. ROIs identified in Ilastik are generally accurate with an F1 score of 0.7708; however, it overestimates the boundaries resulting in segmented masks that are larger than the expected truth masks. Cellpose produces masks that closely align to the ground truth masks. Even though individual masks yield a higher F1 score, crucial masks around the borders of the image are missing, averaging to an F1 score of 0.8773. The SAM model performs accurately but adds unnecessary additional masks in some locations, leading to an average F1 score of 0.7379.

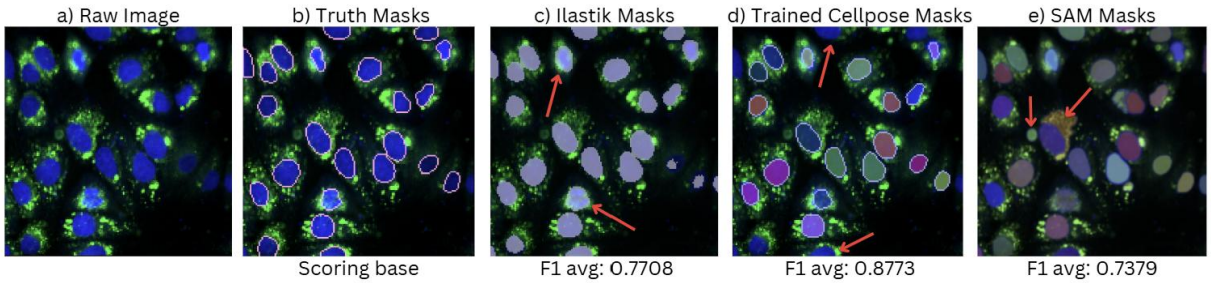


Figure 4: Comparison of Segmentation results of (a) raw image with (b) truth masks (c) Ilastik, (d) Fine-tuned/Trained Cellpose , and (e) SegmentAnything (SAM). Referenced details are marked with red arrows

Baseline Benchmarking

This data quantifies a baseline benchmark analysis of segmentation accuracy across Ilastik, Cellpose, and SAM. While this data portrays each model's ability to segment cells within these images accurately, it is not an encompassing truth for the accuracy of these models. Factors such as variation in cell type and its fluorescent labeling, as well as ground truth labeling and parameter selection, can affect these models. The cells used in this study were HeLa cells stained with various fluorescent dyes (see Methods for more detail) and an overview of the images are available in Appendix A. Customized parameters in each of the models can account for variations such as the selected diameter in Cellpose and SAM's selection of points and bounding boxes.

Though cell segmentation, especially with ML and DL techniques, is highly sensitive to parameter selection, the parameters for this study were set to default values to allow for greater reproducibility across different image sets, models, and with external studies. This approach was selected because it focuses on consistency and comparability. For instance, in Figure 5, a comparative analysis of Ilastik segmentation is performed on an image with automatic feature selection and manual feature selection. There is noticeable variation and quantitative difference in F1 score across the two images. Additionally, Figure 6 shows a comparison between manually selected cell diameters in Cellpose. An underestimation or overestimation can impact the effectiveness of even self-trained models. These results show the importance of parameter optimization when aiming for optimal segmentation performance in accuracy and highly recommended for any cellular segmentation pipeline. But, of course, manual feature selection can result in increased or decreased accuracy based on the experience of the user. This continues to highlight the importance of using default settings to create a standardized baseline.

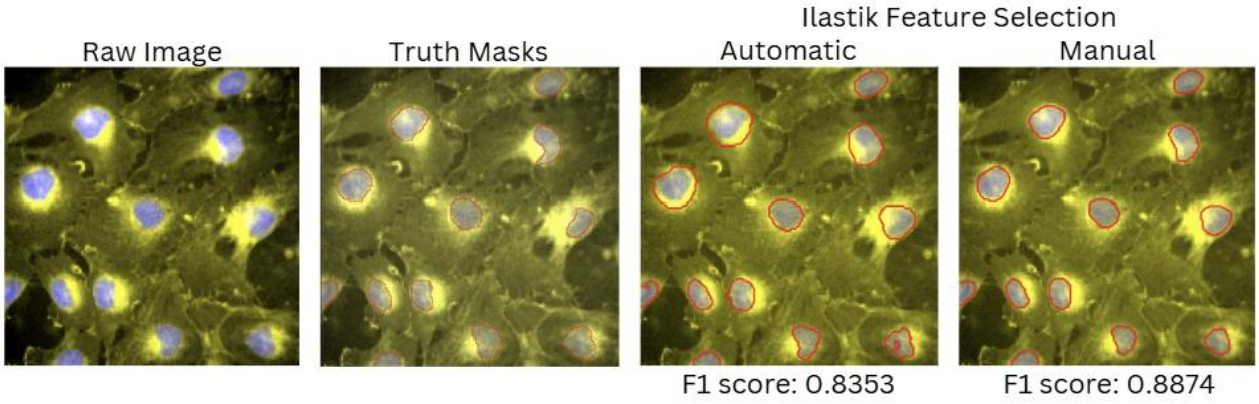


Figure 5: Automatic and Manual Feature Selection Impact on Ilastik Segmentation
Group 4 cells (stained with hoechst, WGA, Phalloidin, conA)

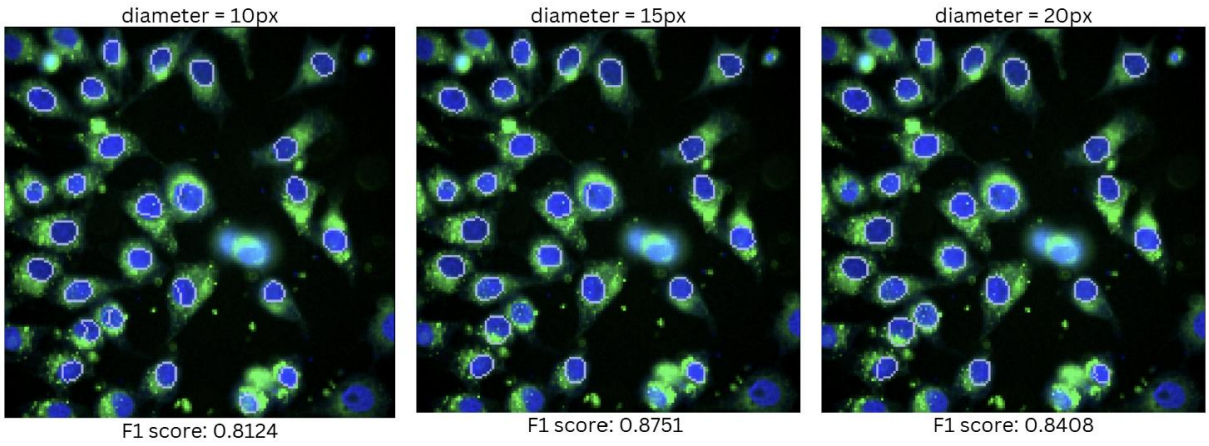


Figure 6: Comparison of diameters as a parameter in Cellpose

Group 3 cells (stained with hoechst and BODIPY)

The greatest observed difference in accuracy is between Cellpose's pre-trained nuclei model and the fine-tuned model trained on additional masks. Using the base nuclei model, an F1 score of just 0.67826 was obtained. In the example image (Figure 6), the pre-trained model misses and over segments cells. With fine-tuning with truth masks on similar cells, a significant increase in accuracy was seen, scoring 0.8773. The segmentation accuracy is even greater than that of SAM's model due to the pixel-wise detail along the identified borders. This analysis highlights the importance of additional training on Cellpose image analysis pipelines.

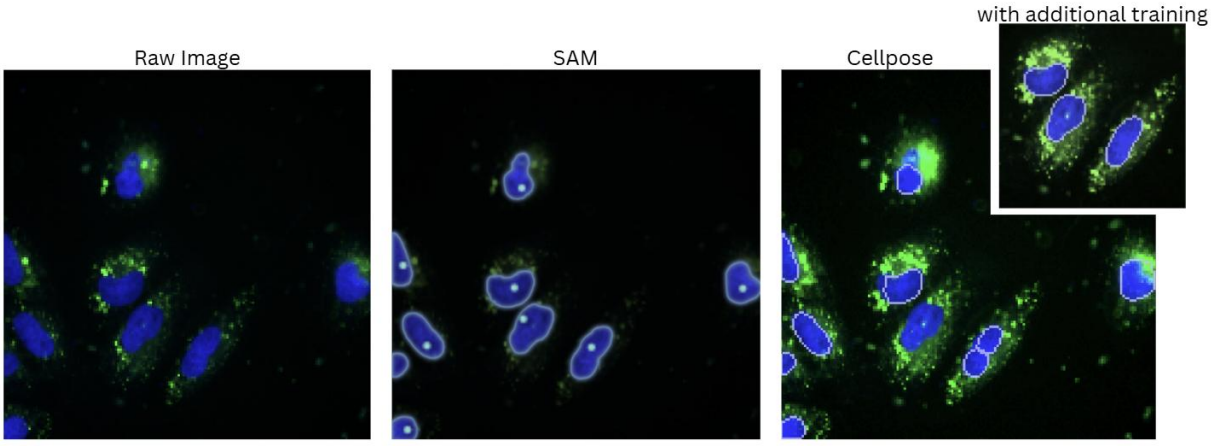


Figure 7: Comparison of SAM with Cellpose pre-trained model and Cellpose fine-tuned model

Group 2 cells (stained with hoechst and BODIPY)

When performing the quantification benchmark analysis, I observed the difficulty in using each of the three software packages. Ilastik and Cellpose both have excellent GUI, allowing for a clear selection of features and parameters. Ilastik, being an ML Random Forest based algorithm, had a much faster compute time, allowing it to provide a live view of segmentation updates while performing shallow learning annotations. With Cellpose, training a new model and running models on large sets of images was computationally more taxing, but overall simple for the user to interpret and decide. Segment Anything, on the other hand, does not have its own GUI, but instead has a website providing a demonstration with all uses of its functions based as python packages to be used within an IDE. All three software provide great segmentation capabilities, but none provide post-segmentation analysis, leaving the output files to be unpacked through software code by the user.

Web Application for Visualization

A web graphical user interface (GUI), or web-based application, was created to aid post-segmentation analysis of Cellpose output data by allowing easy customization of visualization without necessitating writing code to extract the data. Cellpose output data was chosen to be the basis of my cellular segmentation visualization web GUI because while it performs well in segmentation, it does not offer any feature extraction analysis after segmentation. This makes it difficult to view the data outside of the software without downloading and setting up additional software tools. Within Cellpose, a user can only view the masks (without numbered labels) and the most basic of mask statistics, such as the number of masks. Analyzing data through Python is tedious, especially when filtering data and identifying the mask numbers that correspond to the cells within the original image. This web-based application aims to streamline the process of visualizing and extracting data from the output of completed cellular segmentation.

The web-based application was implemented through HTML, CSS, and JavaScript with a backend in Python 3.9.21 through Flask 3.0 and deployed through Render Web Service. The website and code within the GitHub repository can be found in Appendix B. The code has also been adapted for use in Jupyter Notebook which allows user editing of certain features within the code (Figure 8).

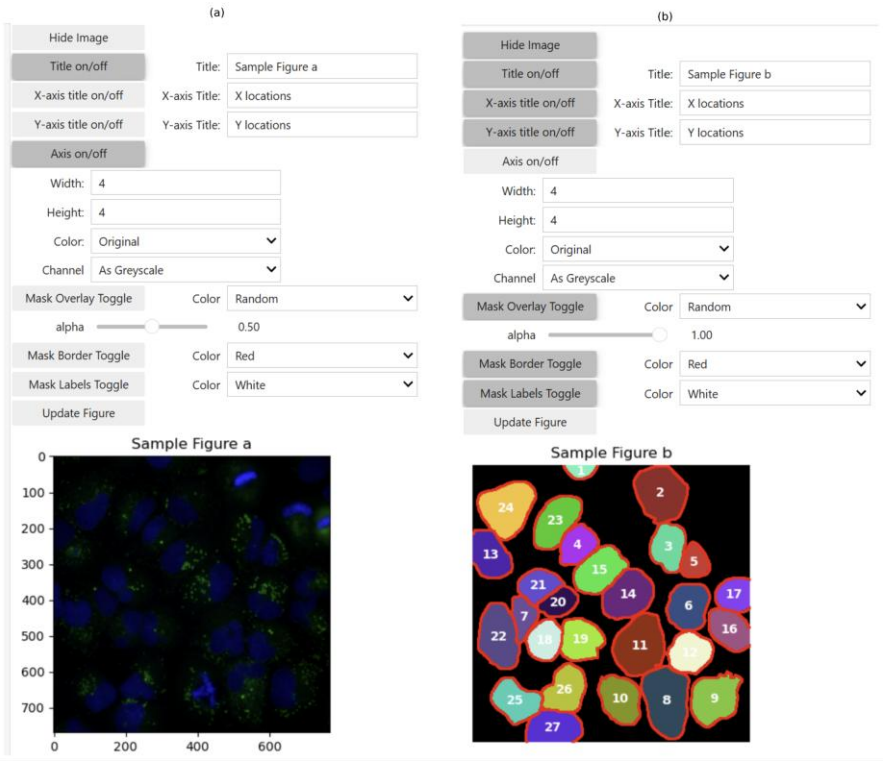


Figure 8: User-input visualization tool in Jupyter Notebook, single image viewing

The web-based application features three pages: visualization of the original image with segmentation annotations, extraction of features, and a fine-tuned BLIP-2 model optimized for captioning cell images. The visualization of the image as a whole (Figure 9) allows customization of graphing features (e.g. title, axis, axis labeling), image display (e.g. color/channel), mask overlay, mask border overlay, and mask labeling that corresponds with the data organization. This tool can easily simplify visualization, which can respond to user input with just the click of a button and display various data through feature extraction such as cell centers, location, size, diameter, and intensity.

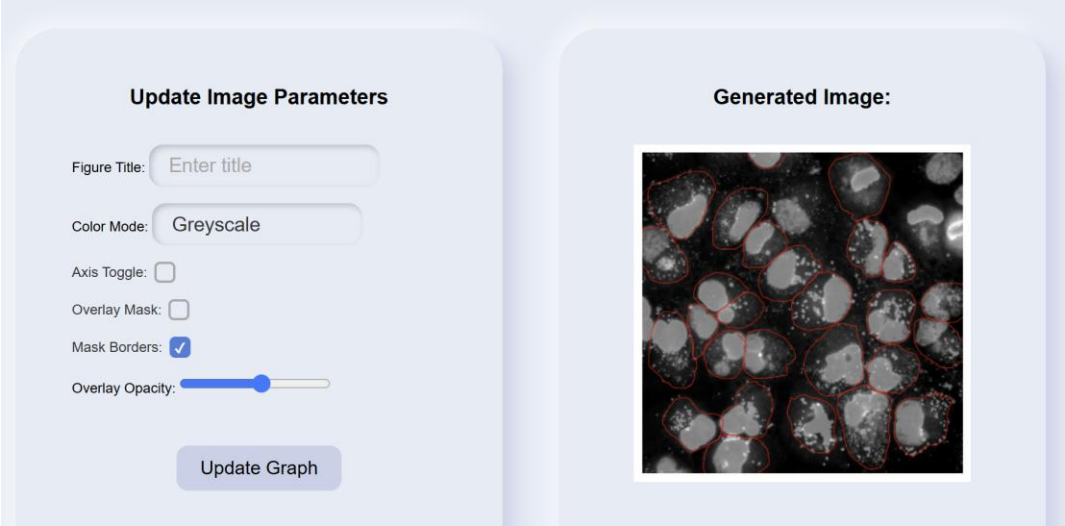


Figure 9: User-input web application visualization tool, single image viewing

BLIP-2 image captioning model

The most prominent idea for the website is to feature the addition of a BLIP-2[33] model fine-tuned specifically for captioning cellular images. BLIP (Bootstrapped Language-Image Pretraining) is a vision-language pretraining (VLP) model specializing in multimodal tasks, leveraging tasks requiring both image and text data. BLIP-2 is built as a generic model that can handle complex tasks with less training than other VLP models such as OSCAR (Object-Semantics Aligned Pre-training), ViLBERT (Vision-and-Language BERT) & LXMERT, and Flamingo (DeepMind). It achieves this type of efficiency by using two unique methods: first, a lightweight Querying Transformer[34] that serves as a bridge between vision models and large language models (LLM); second, it leverages a frozen LLM that significantly reduces the computational expense of end-to-end training. In other words, instead of training an LLM from scratch, it uses an existing LLM whose parameters are fixed – meaning it is no longer being updated– to handle text processing.

BLIP-2 mainly uses Vision Transformers (ViTs) as their vision encoders to pass their data to the query transformer. ViTs comprise the most computationally expensive process of BLIP-2 since they are pretrained on massive datasets such as LAION, containing 5 Billion images paired

with text-based captions. Training on these massive datasets allows BLIP-2 to work on diverse images without additional fine-tuning. Of course, any professional use of BLIP-2 should include fine-tuning training on images and captions of the expected type and quality. In this study, I use ViT-L/16, which extracts features from a single image and generates patch embeddings. These patch embeddings are non-overlapping 16x16 fixed-size patches that can be processed by the transformer. This process can still capture global and most local image features and is significantly more efficient than convolutional neural networks.

While this process is efficient and effective for image captioning, it struggles with other tasks that are important for the proposed pipeline. Generating patch embeddings is crucial for passing information with the transformer, but it reduces the detail that can be extracted. Therefore, tasks such as precise segmentation are often difficult. Additionally, BLIP-2's ViT extracts features and passes its patch embeddings through a single image at a time. This makes it excellent for captioning or answering prompts about individual images but makes it especially difficult to draw meaningful conclusions about patterns between images. Biological data typically consists of large sets of images where the importance falls on identifying differences between experimental conditions, such as determining the effects of a specific treatment.

Training a BLIP-2 model often takes thousands to millions of sample images to properly fine-tune. Due to a limited amount of time in annotating images with truth masks, data augmentation was performed to increase the number of images with existing masks while creating a certain amount of variation. A Python package, albumentations, was used to perform a variety of augmentations by random chance. The workflow contained varying p-values for the following augmentations: horizontal flip, vertical flip, random rotation, brightness adjustments, contrast adjustments, color jitters, elastic transformations, grid distortion, gaussian noise, blur, and local contrast. These provide the dataset with varied images while keeping truth masks intact and are available in Appendix C.

Proposed Image Analysis Pipeline

To overcome this, this study proposes a pipeline that efficiently exploits the specialties of different deep learning algorithms to come to data that will be able to guide researchers to important features in their data. When performing fine-tuned training in the BLIP-2 model, training data based on images segmented with the tried-and-true Cellpose were used and processed through the web GUI to contain accurate mask data. With this, BLIP-2 can easily pinpoint the exact location of ROIs and extract useful information on fluorescent expression, cell spacing, shape, size, and other high-level features. Computational techniques can often quantify features that are difficult to manually analyze such as an overall 10px decrease in diameter of one set or a 5% increase in fluorescent intensity. Additionally, since BLIP-2 does not excel in drawing conclusions about patterns across a large set of images, the user interface will put the data in clear categories, organizing it for optimization by utilizing an LLM to draw comparisons. Essentially, BLIP-2's query transformer acts as a bridge between simple image data and the same language data. This pathway will extend both sides of the bridge from complex image data analysis to complex language analysis. The following captions will be formatted through a python script to be optimally understood by an LLM of a user's choosing.

The comprehensive pipeline (Figure 10) follows image data from acquisition with microscopy through to finalizing pattern analysis. With easily accessible tools such as Cellpose and the web application, the time required for image analysis will be significantly reduced.

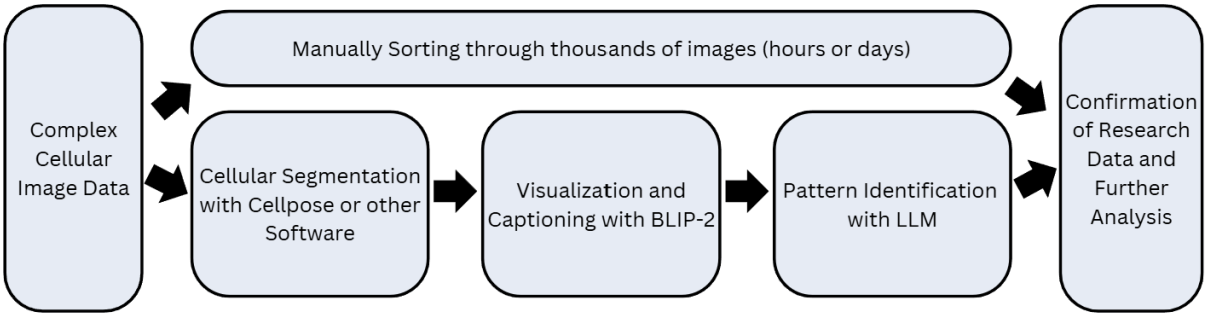


Figure 10: Image Analysis Pipeline Flowchart

Proposed, partially automated pipeline workflow for image analysis of complex cellular data

Future Applications

The BLIP-2 model has yet to be fully linked to the web application. The Web Application currently shows an example output of BLIP-2 and along with additional information. Further studies will properly integrate a seamless pipeline from Cellpose segmentation output to BLIP-2 image captioning of important features. This will include the ability to upload up to three folders of image files – e.g. representing control set, experimental set, and additional – to be annotated with mask borders and captioned with BLIP-2. Additionally, further work hopes to evaluate an optimal LLM for analyzing biological data and integrate its use directly on the web application. This will allow seamless integration of the pipeline to serve as a complete a functional tool for use by researchers.

Chapter 4

DISCUSSION

The current bank of available DL models is incredibly expansive. Hugging Face alone has over 900 thousand models. Most of these models are designed to specialize in a single domain, such as language data or image data. While many of the tasks within this study's proposed pipeline have already been addressed by deep learning techniques, the novelty lies in the way it strings together independent models to utilize their strengths and fine-tune them for the task at hand. By combining the strength of multiple models, each optimized for their specialty, a task as complex as analyzing biological image data can be tackled.

Limitations

Of course, this pipeline has several limitations. For one, it is not fully automated in the way that allows the submission of image data to directly output the information processed by segmentation software, BLIP, and LLM. Instead, it requires manual transference of data from one platform to the next. Currently, the semi-automated process allows for human annotation and verification between the steps, which can act as a trade-off between time and ensuring greater accuracy and adaptability. Future implementations hope to make an LLM directly accessible on the web application that can rely on backend processing to transfer data from one model to the next. With the integration of multiple DL models, more consideration must be placed on the computational cost for researchers with limited access to high-performance CPUs and GPUs. Future studies could explore optimizing the pipeline for efficiency in computational costs by leveraging cloud computing resources or employing lightweight models to improve accessibility.

As discussed, Cellpose performs well in segmentation, but could benefit from additional verification. It is recommended for users to train a Cellpose model on their images and optimize provided parameters to improve specificity when performing segmentation. Differences in annotated segmentation boundaries can significantly impact output results. This is especially true in biological data, accounting for the diverse image types found for cell types, experimental techniques, and microscopy techniques, which can further influence the performance of this model.

The quality of manually annotated truth data directly impacts the quality of trained models on similar data.

Importantly, the results of the proposed pipeline require further quantitative evaluation to ensure reliability and ability to identify trends in cellular image data. Future studies must properly analyze the ability of BLIP-2 to capture relevant features such as cell size, fluorescent expression, odd shapes, or excessive noise. The ability to caption images with applicable information will have a significant downstream impact on the LLM's ability to identify patterns. Further work also hopes to evaluate several LLMs to identify one that is optimal for analyzing biological data and drawing conclusions between multiple images. Either an LLM with a very large pre-training dataset or one specialized in biological jargon would be ideal to handle cellular language data. Difficulties of quantifying the accuracy of BLIP-2 or LLM lie in defining concrete validation metrics for language data and creating large image sets with known cellular patterns. These will be required to properly benchmark the ability of the combined pipeline.

Lastly, AI-generated outputs should be used with caution to avoid biased or false data. Occasionally, biases in AI models can arise from limitations in training data. For example, it is often unclear to a user when a model is suffering from overfitting. Also, AI-generated outputs have been known to potentially include false information – or hallucinations – particularly in the contexts of scientific data and search engines. Therefore, AI should serve as a tool to guide researchers to examine portions of their data more closely rather than replace crucial aspects of their work.

Alternative Models

Given the vast number of available models, BLIP-2 is not the only option for this task. For instance, Visual BERT, an extension of BERT's language model, was strongly considered. Many other models could potentially replace or augment the analysis pipeline, and their accuracy and efficiency warrant further investigation. Additionally, integrating different models could increase generalization to expand this pipeline beyond biological imaging. Future work has the potential to explore the adaptation of this framework for broader applications. This established pipeline has the potential to include a great variety of additional features that would further contribute to its efficiency in the end goal of streamlining scientific research analysis.

Chapter 5

CONCLUSIONS

This study focuses on alleviating a critical bottleneck in biological image analysis by integrating multiple deep learning models into a streamlined workflow. A web application facilitates the integration of visualization of Cellpose segmentation data, BLIP-2 for automated captioning, and an LLM for pattern recognition to establish a complete pathway from image acquisition to a full analysis. This tool will allow biological research scientists to analyze trends in complex biological datasets without requiring coding expertise to interpret segmentation files.

The tools for image and language processing are rapidly evolving with the constant updates in machine learning and deep learning. While the field of artificial intelligence is developing quickly, this study utilizes multiple relevant deep learning models to tackle complex biological data. Approaches such as this one contribute to the growing wealth of knowledge in computational techniques while enhancing tools for development in scientific discovery.

NARRATIVE

When I started my undergraduate education, I was certain that I would major in Biology, but I was unsure of which direction it would take me in this incredibly widespread field. With the privilege of working in several molecular biology labs, I developed a deep interest in genetics in molecular biology and especially the computational aspects of biological research tools. The goal of this signature work project was to familiarize myself with computational techniques to allow me to continue pursuing an education and eventually career in computational biology and bioinformatics. The computer science courses I've taken over the past couple years have allowed me to pursue this path and apply my knowledge in labs and this project.

One of my favorite classes in my college experience was Computational Genomics (COMPSCI 260) at Duke University, taught by Professor Alexander Hartemink. He was able to teach computer science in a biological context in such a way that captured my attention and allowed me to realize how critical these tools are for the scientific research. We explored basic and increasingly more intricate algorithms developed in the 1980s and 1990s and learned how to implement them. For example, we examined BLAST, a well-known algorithm that allows for fast searches and scoring of similar but not necessarily identical sequences. These algorithms, which emerged with the widespread use of computers, caught my attention by showing how the application of computer science techniques requires a shift in thinking. For instance, the Burrows Wheeler Matrix (BWM) and FM-index, used for genomic sequence compression and fragment searches, apply a technique that creates a search index from the first and last characters of each round of the Burrows Wheeler Transform (BWT). While these seemingly random lists of letters are difficult to understand at first glance, the underlying logic for efficient computational searches makes the algorithm highly effective. This class taught me the importance of understanding the foundation of algorithms, many of which are still used today. Therefore, in my paper, I thoroughly explore the beginning of cellular segmentation techniques such as thresholding and the watershed algorithm.

The main inspiration for my project came when I began working at the Bagnat Lab and Di Talia Lab at Duke University during the summer of 2024. I worked with Dr. Priyom Adhyapok to study the sinodal expression pattern of the ERK protein during notochord development of zebrafish

three to eight days post fertilization. Cellpose was being used in the Bagnat and Di Talia lab before I started and its integration to their analytical pipeline intrigued me. We were using Cellpose to computationally measure the fluorescent expression from the confocal microscopy images by comparing the expression within the nucleus to the expression within the cytoplasm. Of course, the first step when quantifying these was to obtain accurate cell segmentation masks for both the cytoplasm and the nuclei. One of my contributions to this project involved manually segmenting masks to train a fine-tuned model for improved accuracy and efficiency. I trained many models on cytoplasm and nuclei segmentation and even performed a comprehensive analysis comparing the accuracy of a trained model over an increasingly large set of truth masks. From this, I realized that training the model took an enormous commitment of time and energy, as it was my focus for around a month of lab work. Additionally, even with a foundation of python coding skills, it was challenging to implement packages and syntax that were unique to Cellpose. Though these tasks were tedious, it was interesting to see how varying training regiments changed the accuracy of models within the same software. This lab work inspired me to explore other segmentation software, which I later incorporated into this project.

The following semester, I took Web-Based Multimedia Design (MEDIART 213) at DKU with Professor Ziv Zeev Cohen. Prior to this, I had experience with website coding languages through self-study and by using website-making tools like WordPress and Elementor while working as a student worker for the Division of Natural Sciences. I had tried to create webpages from scratch for labs and student clubs but struggled to make significant progress. Throughout the class, I enhanced my coding skills in HTML, CSS, and even JavaScript. The class's final project was to create our own website or web application from scratch using HTML, CSS, and JavaScript. My final project consisted of a personal website that reflected my passion for biology and computational sciences and included a page to integrate a web-based visualization tool. I initially committed less time to this portion of the project, but Signature Work and the Capstone sessions gave me the push I needed to complete it. This class also provided me the knowledge I needed to set my project up through a GitHub Repository.

The various biology courses I took throughout my four years, especially the advanced courses such as Molecular Genetic Analysis (BIOL 304) with Professor Linfeng Huang and Advanced Methods in Functional Genomics (BIOL 403) with Professor Jianbo Yue, greatly progressed my knowledge. These classes taught biological techniques and their applications in

research through comprehensive lectures, hands-on lab experiments, and journal club style literature reviews. Learning the details of various microscopy techniques and how each works was crucial to understanding cellular imaging, a significant part of my signature work project. Additionally, reading and engaging with current research papers in molecular biology gave me a view of what imaging techniques are most commonly used and in which context. Moreover, I gained insight into the computational techniques currently being used and where research could benefit from more computational integration: identifying the bottlenecks in scientific research. Fully understanding the field you are working in is essential in creating computational tools to supplement its progression. For instance, biological labs are focused on their research and outcomes and tend to use computational techniques only when they allow their workflows to become more efficient and requires minimal coding skill since many researchers have little expertise in complex computation. Therefore, I settled on the idea of creating an interactive web application with a friendly user interface.

This past year, I've completed two capstone courses working on structuring my project around a topic where I have prior experience but would also be beneficial to my advancement as a student. Throughout my academic journey, I considered several ideas for my signature work. However, the additional time provided by the capstone courses was truly beneficial for further developing this project. It allowed me to develop the interdisciplinary knowledge necessary to align this project with advanced computational biology. I believe that this paper contributes to a field with important applications in the present and holds potential for further great advancements. The completion of this research project and academic paper is an accomplishment, both in the final product and in the knowledge I gained throughout the process. The general goal of my signature work was to familiarize myself with computation techniques to allow me to continue pursuing an education in computational biology and bioinformatics. With the completion of this project, I now know that this next step is one that I am capable of.

REFERENCES

1. Bodaghi, A., Fattahi, N. & Ramazani, A. Biomarkers: Promising and valuable tools towards diagnosis, prognosis and treatment of Covid-19 and other diseases. *Heliyon* 9, e13323 (2023). https://doi.org/10.1016/j.heliyon.2023.e13323

2. Rodriguez-Rios, M., McHugh, B.J., Liang, Z. *et al.* A fluorogenic, peptide-based probe for the detection of Cathepsin D in macrophages. *Commun. Chem.* **6**, 237 (2023). https://doi.org/10.1038/s42004-023-01035-9

3. Tsien, R.Y. The green fluorescent protein. *Annu. Rev. Biochem.* **67**, 509–544 (1998). https://doi.org/10.1146/annurev.biochem.67.1.509

4. Piltti, K.M., Cummings, B.J., Carta, K. *et al.* Live-cell time-lapse imaging and single-cell tracking of *in vitro* cultured neural stem cells—Tools for analyzing dynamics of cell cycle, migration, and lineage selection. *Methods* **133**, 81–90 (2018). https://doi.org/10.1016/j.ymeth.2017.10.003

5. Piccinini, F. *et al.* Software tools for 3D nuclei segmentation and quantitative analysis in multicellular aggregates. *Comput. Struct. Biotechnol. J.* **18**, 1287–1300 (2019). https://doi.org/10.1016/j.csbj.2020.05.022

6. Cakir, S., Gauß, M., Häppeler, K. *et al.* Semantic segmentation for autonomous driving: Model evaluation, dataset generation, perspective comparison, and real-time capability. *arXiv* **2207.12939** (2022). https://arxiv.org/abs/2207.12939

7. Qureshi, I., Yan, J., Abbas, Q. *et al.* Medical image segmentation using deep semantic-based methods: A review of techniques, applications, and emerging trends. *Inf. Fusion* **90**, 316–352 (2023). https://doi.org/10.1016/j.inffus.2022.09.031

8. Ma, J., Xie, R., Ayyadhury, S. *et al.* The multimodality cell segmentation challenge: toward universal solutions. *Nat. Methods* **21**, 1103–1113 (2024). https://doi.org/10.1038/s41592-024-02233-6

9. **Otsu, N.** A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **9**, 62–66 (1979). https://doi.org/10.1109/TSMC.1979.4310076.
10. Roerdink, J. & Meijster, A. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae* **41**, (2000).

11. Kornilov, A.S. & Safonov, I.V. An overview of watershed algorithm implementations in open source libraries. *J. Imaging* **4**, 123 (2018). https://doi.org/10.3390/jimaging4100123

12. Rueden, C. T. *et al.* ImageJ2: ImageJ for the next generation of scientific image data. *BMC Bioinformatics* **18**, 529 (2017). https://doi.org/10.1186/s12859-017-1934-z

13. Schneider, C. A., Rasband, W. S. & Eliceiri, K. W. NIH Image to ImageJ: 25 years of image analysis. *Nat. Methods* **9**, 671 (2012). https://doi.org/10.1038/nmeth.2089

14. Schindelin, J. *et al.* Fiji: An open-source platform for biological-image analysis. *Nat. Methods* **9**, 676–682 (2012). https://doi.org/10.1038/nmeth.2019

15. Gedraite, E. S. & Hadad, M. Investigation on the effect of a Gaussian Blur in image filtering and segmentation. *Proc. ELMAR-2011*, 393–396 (2011). https://doi.org/10.1109/ELMAR.2011.6044249

16. **Grand Challenge.** Challenges. *https://grand-challenge.org/challenges/*

17. Breiman, L. Random forests. *Mach. Learn.* **45**, 5–32 (2001).

18. Berg, S. *et al.* Ilastik: Interactive machine learning for (bio)image analysis. *Nat. Methods* **16**, 1226–1232 (2019). https://doi.org/10.1038/s41592-019-0582-9

19. ZEISS. AI for Advanced Image Analysis: A Practical Guide for Microscopy Analysis with ZEISS Software. (2023). Available at: https://nif.hms.harvard.edu/sites/nif.hms.harvard.edu/files/education-files/Zeiss%20AI%20eBook.pdf.

20. Jones, M. L. & Strange, A. Artificial intelligence for image analysis in microscopy. *Wiley Anal. Sci.* (2023). https://www.analyticalscience.wiley.com

21. OpenAI, Achiam, J., Adler, S. *et al.* GPT-4 technical report. *arXiv* **2303.08774** (2023). https://doi.org/10.48550/arXiv.2303.08774

22. Ronneberger, O., Fischer, P. & Brox, T. U-Net: convolutional networks for biomedical image segmentation. *arXiv* **1505.04597** (2015). https://arxiv.org/pdf/1505.04597

23. Stringer, C., Wang, T., Michaelos, M. & Pachitariu, M. Cellpose: A generalist algorithm for cellular segmentation. *Nat. Methods* **18**, 100–106 (2020). https://doi.org/10.1038/s41592-020-01018-x

24. Schmidt, U., Weigert, M., Broaddus, C. & Myers, G. Cell detection with star-convex polygons. *arXiv preprint* arXiv:1806.03535 (2018).

25. Bankhead, P., Loughrey, M.B., Fernández, J.A. *et al.* QuPath: Open source software for digital pathology image analysis. *Sci. Rep.* **7**, 16878 (2017). https://doi.org/10.1038/s41598-017-17204-5

26. Icy. *https://icy.bioimageanalysis.org/*

27. Carpenter, A. E., Jones, T. R., Lamprecht, M. R. *et al.* CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biol.* **7**, R100 (2006). https://doi.org/10.1186/gb-2006-7-10-r100

28. Kirillov, A. *et al.* Segment Anything. *arXiv* **2304.02643** (2023). https://arxiv.org/abs/2304.02643

29. Bannon, D. *et al.* DeepCell Kiosk: Scaling deep learning–enabled cellular image analysis with Kubernetes. *Nat. Methods* **18**, 43–45 (2020). https://doi.org/10.1038/s41592-020-01023-0

30. Segment Anything. *https://segment-anything.com/*

31. Hugging Face. *https://huggingface.co/*

32. BioImage.io. *https://bioimage.io/*

33. Li, J. *et al.* BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. *arXiv* **2301.12597** (2023). https://arxiv.org/abs/2301.12597

34. Zhang, J., Xiong, F. & Xu, M. G3PT: Unleash the power of Autoregressive Modeling in 3D Generation via Cross-scale Querying Transformer. *arXiv* **2409.06322** (2024). https://arxiv.org/abs/2409.06322

35. Piccinini, F., Balassa, T., Carbonaro, A. *et al.* Software tools for 3D nuclei segmentation and quantitative analysis in multicellular aggregates. *Comput. Struct. Biotechnol. J.* **18**, 1287–1300 (2019). https://doi.org/10.1016/j.csbj.2020.05.022

36. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. *arXiv* 1512.03385 (2015). https://arxiv.org/pdf/1512.03385

37. BioGPT: Generative pre-trained transformer for biomedical text generation and mining. *arXiv* 2210.10341 (2022). https://arxiv.org/abs/2210.10341

38. Ulman V, Maška M, Magnusson KE, Ronneberger O, Haubold C, Harder N, Matula P, Matula P, Svoboda D, Radojevic M, Smal I, Rohr K, Jaldén J, Blau HM, Dzyubachyk O, Lelieveldt B, Xiao P, Li Y, Cho S, et al. An objective comparison of cell-tracking algorithms. *Nat Methods.* 2017;14(12):1141-1152. doi:10.1038/nmeth.4473.

39. Maška, M., Ulman, V., Nečasová, T., Guerrero Peña, F. A., Ren, T. I., Meyerowitz, E. M., Scherr, T., Löffler, K., Mikut, R., Guo, T., Wang, Y., Allebach, J. P., Bao, R., M., N., Rahmon, G., Toubal, I. E., Palaniappan, K., Lux, F., Matula, P., . . . Kozubek, M. (2023). The Cell Tracking Challenge: 10 years of objective benchmarking. *Nature Methods*, *20*(7), 1010-1020. https://doi.org/10.1038/s41592-023-01879-y

40. Caicedo JC, Goodman A, Karhohs KW, Cimini BA, Ackerman J, Haghighi M, Heng C, Becker T, Doan M, McQuin C, Rohban M, Singh S, Carpenter AE. Nucleus segmentation across imaging experiments: The 2018 Data Science Bowl. *Nat Methods.* 2019;16(12):1247-1253. doi:10.1038/s41592-019-0612-7.

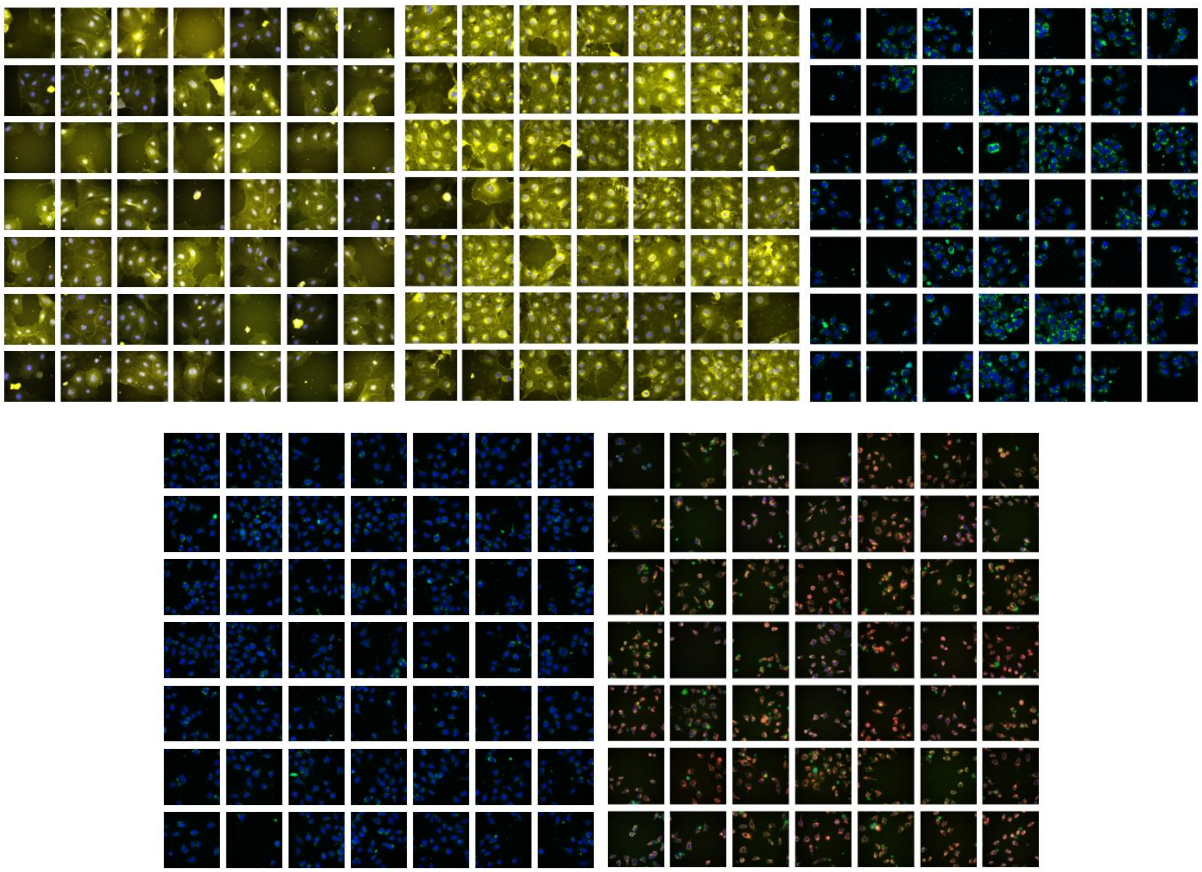
41. Graham, S. et al. CoNIC: Colon nuclei identification and counting challenge 2022. *arXiv* 2111.14485 (2021). https://doi.org/10.48550/arXiv.2111.14485.

42. Ybelkada. (2023). *Fine-tune BLIP2 on an image captioning dataset with PEFT*. Hugging Face. https://huggingface.co/notebooks/Fine_tune_BLIP2_on_an_image_captioning_dataset_PEFT

ADDITIONAL MATERIAL

Appendix A

Overviews for sets of images used in analysis



Five sets of images used for segmentation of truth annotations, testing, and model training

Fluorescent staining corresponds to the lists below.

Group 1: hoechst, calcein AM, PI

Groups 2 and 3: hoechst, BODIPY

Group 4: hoechst, WGA, Phalloidin, conA

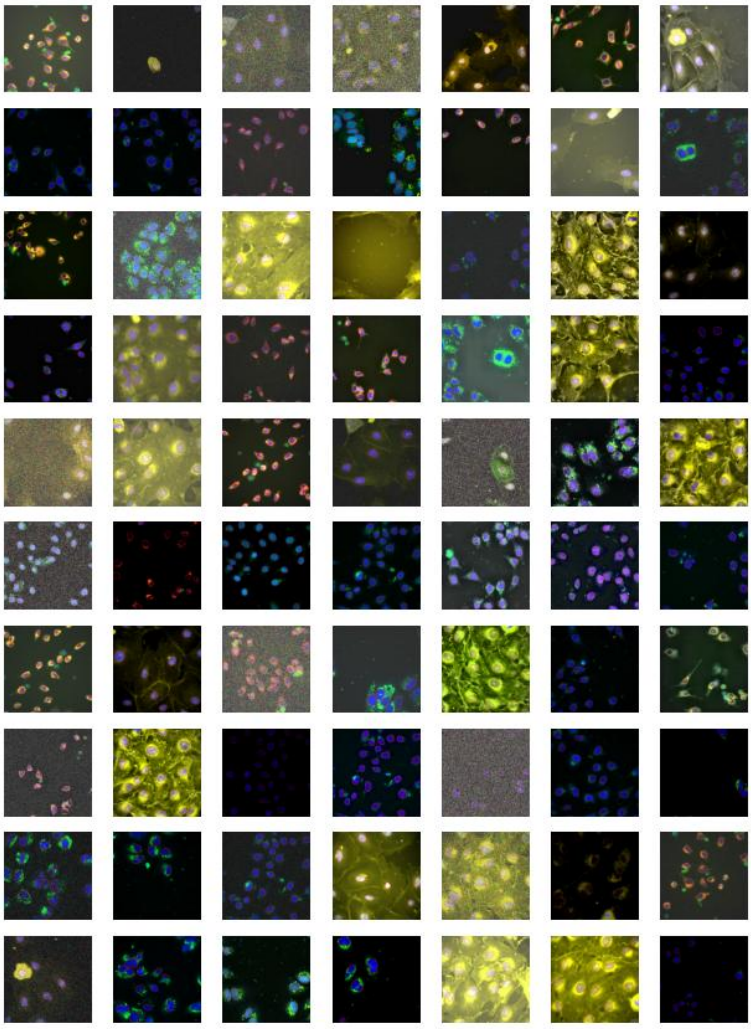
Group 5: hoechst, WGA, Phalloidin, SYTO

Appendix B

Github Repository link: https://github.com/chyiricketts/SW_CellVis.git

Web application link: https://sw-cellvis.onrender.com/

Appendix C



Sample of images augmented with python library Albumentations to obtain diverse image training data for vision-language model BLIP-2