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

In this study, an entirely new web-based application and a fine-tuned vision language model, Bootstrapped Language-Image Pretraining 2 (BLIP-2), were developed for the visualization of segmented cellular data and multimodal translational ability to optimize the efficiency of image analysis in scientific research. 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. Cellular segmentation software was thoroughly analyzed 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. A recommended pipeline is established for segmenting masks with Cellpose for pixel-wise, accurate segmentation and highlighting ROIs with tools provided by the web-based application, followed by automated captioning with BLIP-2 to enable pattern identification with any large language model (LLM) with the recommended use of BioGPT. This comprehensive pipeline follows image analysis from raw to processed language format by utilizing the strong suits of deep learning algorithms. While the patterns identified by the BioGPT or other LLMs using this pipeline should not be regarded as definitive research bindings, this pipeline can provide researchers with a more efficient method of recognizing trends in complex datasets.

ACKNOWLEDGEMENTS

Special thanks to Professor Linfeng Huang and the Wang Cai Biochemistry Lab at DKU for their mentoring and resources throughout this project. Also, thank you to the Summer Research Scholars (SRS) program for providing a wonderful learning opportunity. Lastly, thank you to DKU Undergraduate Studies.

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

As humans, we rely heavily on our sense of sight to understand the world around us. In scientific research, complex data often becomes more accessible and intuitive when accompanied by visual representations. Not only can image representations improve our comprehension, but they can also be a powerful tool that allows us to reason about processes by observing their occurrence or effects. In biological research, when processes are happening at a molecular level, visualization becomes even more 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. A wide range of techniques for biomarkers, including fluorescent dyes, fluorescent proteins, 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. This highly versatile tool can be applied anywhere from fundamental biological processes to specific interactions within cellular dynamics to personalized medicine and more.

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. Additionally, the range of available microscopy options has expanded. For example, the visual characteristics of the same sample prepared and imaged through a confocal microscope and a transmission electron microscope (TEM) will be wildly different. 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 shallow learning models 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.

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. The first hurdle to overcome when analyzing cellular images with computational analysis is allowing the computer to identify where the cell, or other ROI, is in the image. 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 intensity of individual cells. In recent years, cellular segmentation has evolved drastically from human annotation to simple techniques to deep learning. It is currently being used in various fields, including medical imaging analysis and self-driving vehicle vision. I aim to provide sufficient background and explanation of terminology to emphasize the diversity of tools available in the field and how AI integrations in software such as Cellpose contribute to the growth of bioimage analysis.

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
Group 2: hoechst, BODIPY
Group 3 and 4: hoechst, WGA, Phalloidin, conA
Group 5: hoechst, WGA, Phalloidin, SYTO

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 __

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

StarDist

The StarDist Version used was __

Illastik

The Illastik software version used was __

Segment Anything

The SegmentAnything version used was the online resource

Albumentations

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

Add Model specifications and training workflow

Bio-GPT

Add Model specifications and training workflow

Chapter 3

RESULTS

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 and the Watershed algorithm. 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 (Figure 1a). 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 cells in close proximity or 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 (Figure 1b). While the Watershed algorithm can differentiate cells, it is highly reliant on data presenting a consistent gradient toward its borders. Its computational speed and simplicity made it ideal for data like nuclei segmentation in fluorescence microscopy images. As with Otsu's method, this approach is not ideal for images with similar grey means with varying textures or images with a large amount of noise.

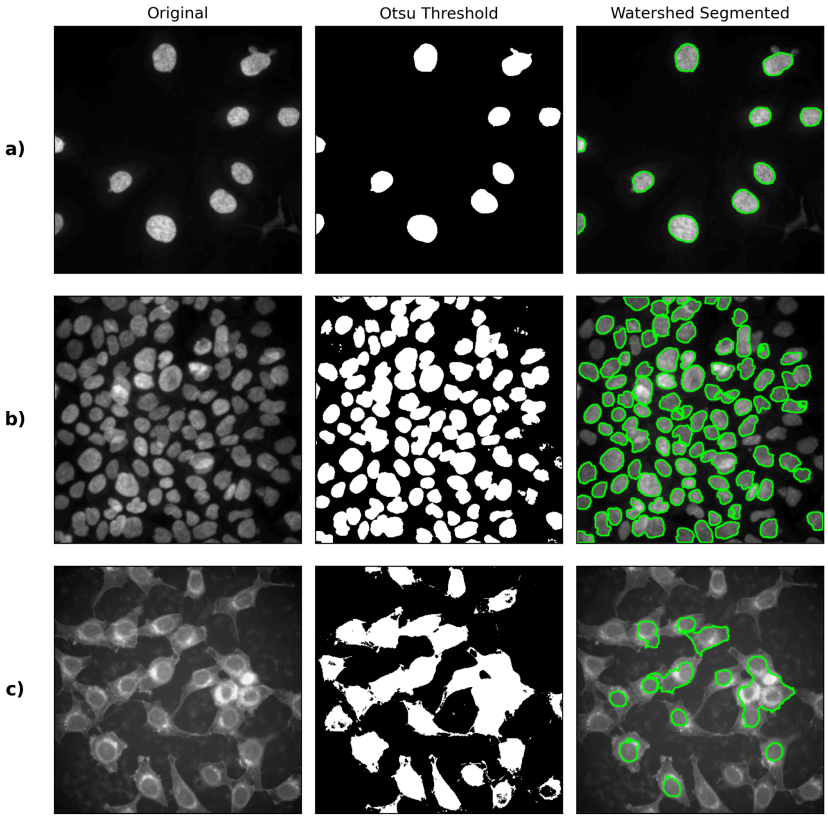

Figure 1: Comparison of Otu's method for thresholding and the watershed algorithm. Demonstrating effectiveness of traditional cell segmentation techniques on (a) hoechst 33342 (ThermoFisher, H3570) stained, dispersed HeLa cells, (b) hoechst 33342 (ThermoFisher, H3570) stained, clustered Hela cells, and (c) hoechst 33342 (ThermoFisher, H3570), WGA, Phalloidin, and conA fluorescent stained HeLa cells. Details of the calculation of Otsu's method and the Watershed Algorithm can be found in Methods

Cellular segmentation presents a plethora of challenges to overcome, which no single software has been able to do. The most immediate issue is creating an algorithm that is able to understand and overcome the high variation in microscopy samples. Some of these issues are illustrated above, including separating cells in close proximity or in contact with each other, low contrast within and between cells, and noisy images (Figure 1c). Diversity in cell shape, size, texture, fluorescent signals, and microscopy technique contributes to the enormous heterogeneity of cell types.

One common type of traditional image editing that can assist in reducing the stress of additional noise in images is called a Gaussian blur. A Gaussian blur is an image-blurring filter typically applied in 2 dimensions to produce a surface where the original points become contours of its immediate concentric circle. While it removes small noise, details within the image are reduced by blurring, as suggested by the name. Still, Gaussian blurs have proved to increase the effectiveness of computer vision programs and are commonly used as a pre-processing measure in many image analysis contexts, especially segmentation. [**CITE GAUSSIAN BLUR PAPER**] Numerous programs offer traditional image editing tools such as thresholding, watershed, Gaussian, and more, and they are available and remain integral to image analysis pipelines.

Still, challenges of overfitting, variation, and time consumption plague the cell segmentation community and require 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 the global scientific community. One of the most innovative efforts in the field is the Grand Challenges in Bioimage Analysis (**CITE SOURCE**). This initiative provides a dataset, typically with a set of ground truth values, released to the community to analyze with their own methods. These types of events not only encourage participation and innovation but also allow for collaboration and transparency to learn from one another. The past couple of years have featured a domination of machine learning and deep learning-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 artificial intelligence refers to the ability of machines to mimic human learning and intelligence. The basis of machine learning is to analyze input data to recognize patterns and optimize the processes to replicate that output data. For instance, a commonly used machine learning technique called Random Forest [**CITE RF PAPER**] 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.

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. While this logical reasoning has the ability to outperform machine learning in many cases, it requires more data points to improve its accuracy. 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 (**CITE GPT SOURCE**). 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 majority of cellular segmentation software today revolves around a Unet framework. The release of Unet in 2015 was monumental for the application of image processing. It featured a U-shaped encoder and decoder made fully with convolutional neural network layers. Unet, simply put, 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 is able to produce much more accurate results with a smaller testing dataset and deal with more complex shapes than other methods. For instance, Cellpose is only trained on around 70,000 segmented cells and only requires around an additional 250 labeled images to induce effective fine-tuning (**CITE CELLPOSE**). Unet is even incorporated in the design of many algorithms and software, including superresolution, Gaussian noise, and other cellular segmentation software such as StarDist, Cellpose, the popular Dall-E art generator, and more.

More labs are adopting deep learning into their image processing pipelines and encountering challenges balancing its effectiveness with the time spent annotating training data and customizing parameters. 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. 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, hopefully, accurately segments your targeted specimen. Oftentimes, labs specialize in a specific model and imaging technique. A common problem of overfitting in deep learning models occurs when a model performs well on the training data but poorly on unseen data. This can be a direct result of manually labeled ground truth values saturated with highly 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 (**CITE SOURCE**) 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.

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 (segment anything), 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.

The motivation of my study was to create an extension of one or multiple of the existing cellular segmentation software to aid in post-segmentation analysis. This would reduce the time and coding expertise required for researchers to unpack and reorganize complicated outputs in order to filter and evaluate their targeted cellular features. A literature review was first conducted on several of the available software and evaluated based on qualitative benchmarking to facilitate narrowing down my search to perform a deeper quantitative analysis. In Table 2, the most popular software are summarized based on a literature review and basic analysis. Categorizing

the software by their algorithm type (traditional, ML, and DL), whether their package includes pre-trained models, self-trained models, and features.

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-troughput, 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 UNet (CNN-based)	Yes	Yes	Yes	Pretrained models for nuclear/cell segmentation
StarDist	2018	DL using UNet (Star-convex polygons)	Yes	Yes **	Yes **	Optimized for star-convex objects
Cellpose	2020	DL using UNet (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

Table 1: Qualitative assessment of 8 available segmentation platforms

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

ImageJ has been a staple in scientific image analysis for decades since its development as NIH Image in 1987 (**CITE 25 YEARS PAPER).** While it specializes in traditional forms of image analysis, it remains one of the most widely used tools by keeping a core set of design principles. It offers access to the most basic of tools such as pixel measurements, color scaling, and image cropping, which are still integral to the needs of any image analysis pipelines while incorporating a huge variety of scientific operations for pre-processing of images, including Gaussian blur, Hessian matrix, Mean, Variance, Min, Max, and all sorts of noise reduction and diffusion filtering. The flexibility of the program to open and edit all types of image files greatly increases its accessibility. ImageJ, now also known as Fiji, has been able to expand its core design by offering optional plug-ins. These plug-ins encompass a large range including 3D visualization, Otsu's and other sorts of thresholding, the watershed algorithm, and other types of simple cellular segmentation. Deep learning-based cellular segmentation is available through plug-ins collaborating with several of the software that will be discussed, including Illastik, StarDist, and Cellpose.

Illastik is an example of machine learning cellular segmentation, or 'shallow learning.' Instead of relying on neural networks, it uses the previously described Random Forest technique 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. When employing this training data, only labeled pixels are considered. At the same time, the rest are ignored in contrast to the annotations required for deep learning truth values data in which every pixel must belong to a given class (Figure 2). Random forest, as well as other shallow learning techniques, will randomize and build pattern pathways off of the limited training data to reverse engineer the optimal computational pathway to arrive at the corresponding segmentation.

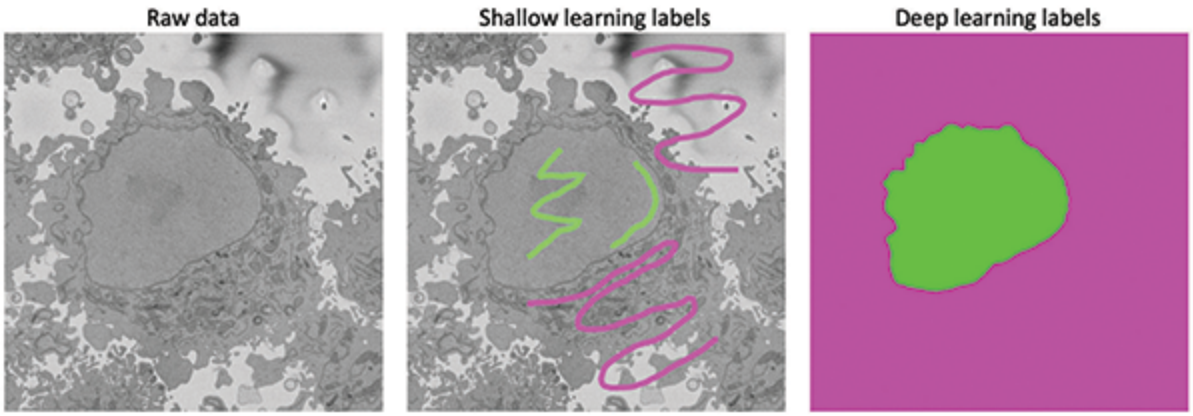

Figure 2: Illustrating the difference between shallow learning and deep learning labels
A is the original image in greyscale, b is the image prepared for training the Ilastik program with this version and c is a deep learning label made in cellpose **REPLACE THIS FIGURE**
RIGHT NOW IT IS NOT MADE BY ME AND TAKEN FROM ANOTHER PAPER– WILL FIX

Deep learning algorithms require a far more strict set of data annotations for truth values. This process can be extremely tedious and time-consuming as carefully tracing one's mouse over thousands of cell or nuclei borders just to train a model can seem like an inefficient use of time. Ultimately, the results of utilizing deep learning have proven to be significantly more accurate and minimize the need for manual corrections after processing. Available interfaces for deep learning-based cellular segmentation which tackle different problems include Cellpose, StarDist, Deepcell, and Aivia.

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 with a Unet architecture, it is able to 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 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

helps 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.

Unlike Cellpose and StarDist, DeepCell 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.

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 provides a platform for several of the aforementioned programs and specializes in histology samples. Other platforms, including Icy and CellProfiler, 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.

Lastly, Segment Anything (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 perceptrons (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.

All the tools mentioned above are open-source, fostering opportunities for others to learn by developing wrappers and plugins that have the potential to further advance the tools themselves and the knowledge in the field. Amongst the software analyzed, only Aivia is closed-source and is only available as a commercial product. From 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. A review of its ease of use was purely based on user accounts.

When selecting software to move into quantitative analysis, the diversity of algorithms was considered, as well as their differences in the availability of pre- and self-trained models and specialized features to compare their accuracy on a set of fluorescent HeLa cell images. The four selected software, Illastik, SegmentAnything, StarDist, and Cellpose, represent ML techniques, zero-shot generalization, and two DL CNNs optimized for star-convex polygons and generalization. Table 3 summarizes the data for the quantitative comparison of four software: Ilastik, SegmentAnything, StarDist, and Cellpose.

	Ilastik	**Segment Anything**	**StarDist**	**Cellpose**
Ease of Use	Moderate	Easy	Difficult	Difficult
Learning Curve				
Algorithm Type	ML (Random Forest)	ViT with Promptable DL	DL using UNet (Star-convex polygons)	DL using UNet (flow-based CNN)
Customizability	High	High	High	High
F1 score with pre-trained model	N/A	**IN PROGRESS**	**IN PROGRESS**	0.67826 Should remove border cells
Training difficulty				Easy, just upload files
F1 score with trained model on similar cells	**IN PROGRESS**	N/A	**IN PROGRESS**	0.79955
F1 score with trained model on different cells	**IN PROGRESS**	N/A	**IN PROGRESS**	0.2534
Weighted average F1 score	**IN PROGRESS**	**IN PROGRESS**	**IN PROGRESS**	0.6555

Table 3: Testing using F1 scoring on pretrained models and trained models

WRITE MORE ABOUT THE SPECIFICS OF WHICH PRETRAINED MODELS AND TYPE OF CELL IMAGING

Comparative qualitative analysis was carried out using three sets of 49 images with combinations of fluorescence staining (see Methods for more detail). Pre-trained models were used for the quantification of StarDist and Cellpose. Specifically, Cellpose's nuc model and StarDist's ___ model. Two sets of similar images were annotated with ground truths to train Illastik, StarDist, and Cellpose. Ilastik does not provide pre-trained nuclei models; therefore, the segmented cells were only analyzed after shallow learning training annotation for Ilastik. Conversely, SegmentAnything is a base model for segmentation and does not provide the function for self-training models. The three testing sets were also annotated with ground truths for comparison with the model's segmentation results. The segmentation accuracy was quantified with the F1 Score or the Dice Coefficient (see Methods for more details).

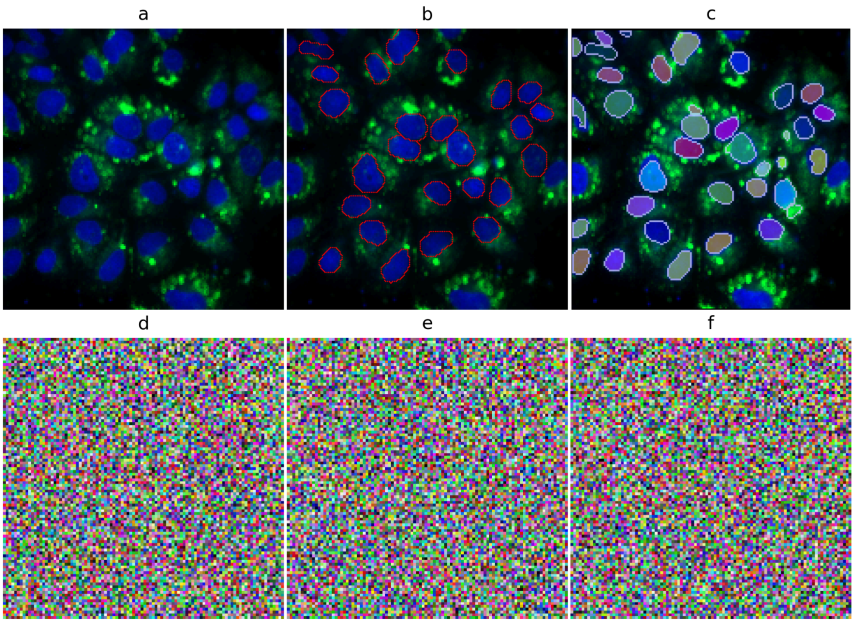

Figure 3: Comparison of Segmentation results of (a) raw image with (b) truth masks from (c) Cellpose, (d) StarDist, (e), Illastik, and (f) SegmentAnything

STILL IN THE PROCESS OF FINISHING MODEL TRAINING AND ANALYSIS. ADD DISCUSSION OF DATA RESULTS IN THIS SECTION. But I expect that Cellpose and Stardist will perform best on this data? Unsure. StarDist may also do very well. I believe SegmentAnything actually struggles a lot with cellular data despite its popularity in the media.

Masks that were along the borders were removed for inconsistency in being able to identify cells that were partially obstructed.

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, can affect these models. The cells used in this study were HeLa cells stained with various fluorescent dyes (see Methods for more detail). My ground truth annotations are available in **Appendix A**. Additionally, customized parameters in each of the models can account for variations such as the diameter parameter in Cellpose and SegmentAnything's selection of points and bounding boxes.

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, like StarDist, while it performs well in segmentation, it does not offer any feature extraction analysis after segmentation, and it is difficult to view the data outside of the software without downloading and setting up additional software. Within Cellpose, a user can only view the masks (without numbered labels) and basic mask statistics, such as the number of masks. Analyzing data through Python is not particularly difficult but 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. It can be accessed at https://sw-cellvis.onrender.com, and the code can be found through my GitHub repository: https://github.com/chyiricketts/SW_CellVis. The code has also been adapted for use

in Jupyter Notebook: **JUPYTER NOTEBOOK LINK** (Figure 4). The web-based application features three pages: visualization of the original image as a whole, a fine-tuned BLIP model optimized for captioning cell images, and built-in access to BioGPT, an LLM trained on biological data. The visualization of the image as a whole (Figure 5) allows customization of graphing features (title, axis, axis labeling), image display (size, 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.

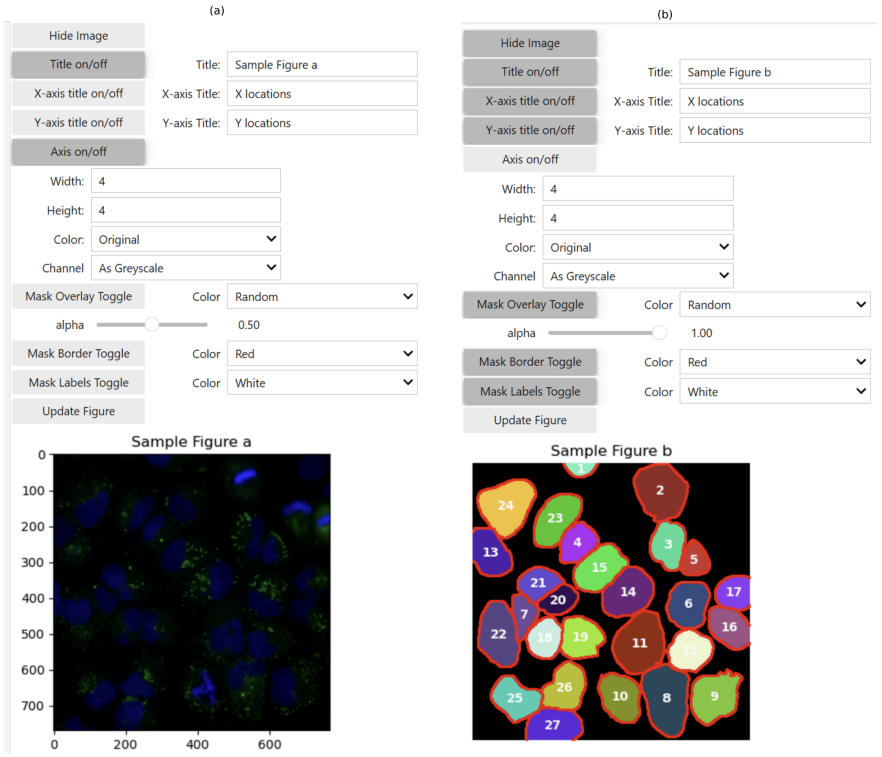

Figure 4: User-input visualization tool in Jupyter Notebook, single image viewing

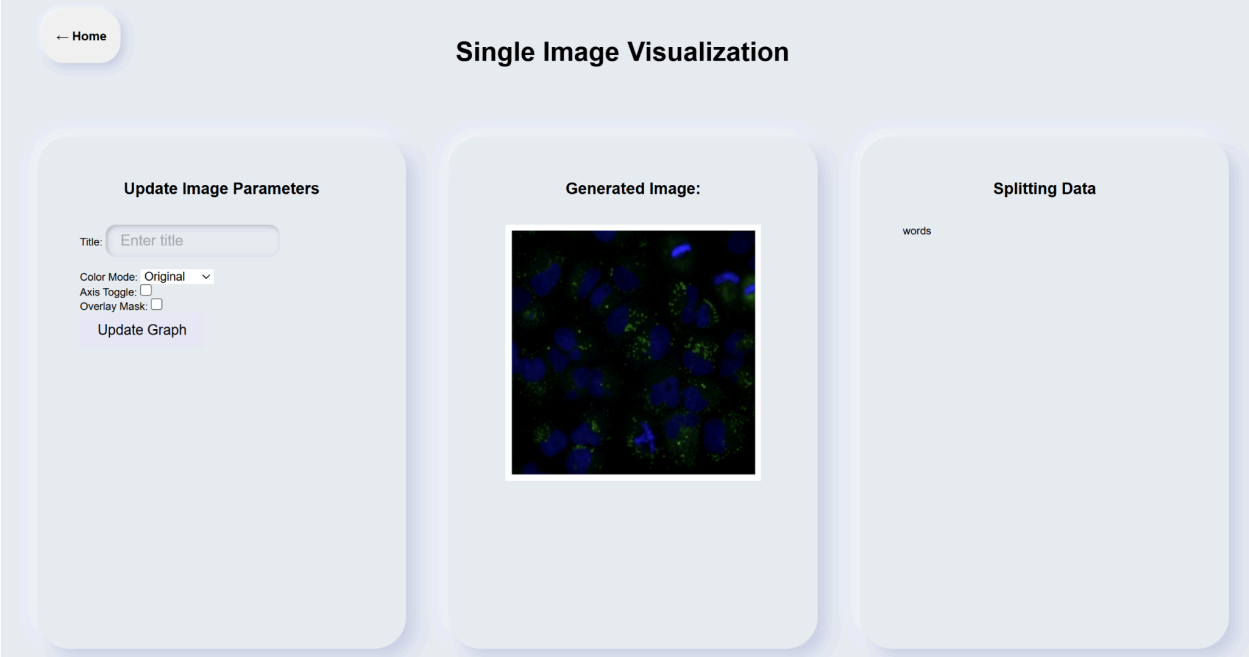

Figure 5: User-input web application visualization tool, single image viewing

The most prominent feature of the website is the addition of a BLIP-2 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 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 to work fairly well on diverse images without additional fine-tuning. Of course, any professional use of BLIP 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'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.

To overcome this, this study proposes a pipeline that efficiently exploits the strong-suits 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 model, training data based on images segmented with the tried and true Cellpose were used and processed through the web GUI to contain the accurate mask data. With this, BLIP is able to identify the exact location of DOIs easily and extract useful information on fluorescent expression, cell spacing, shape, size, and other high-level features. Additionally, since BLIP 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 a large language model to draw comparisons. Essentially, BLIP'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.

Training a BLIP 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 used 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.

The BLIP model is (**WILL BE**) linked and easily accessible on the second page of the website. To remove the bulk for the rest of the pages, the BLIP model is implemented on a separate web application. Up to 3 groups of files can be uploaded such as testing data, a control group, and one more. The files can be easily uploaded and processed through the BLIP model. The following captions will be formatted through a python script to be optimally understood by an LLM of a user's choosing.

THIS IS NOT IMPLEMENTED YET AND THIS DESCRIBES THE HOPE DESCRIPTION OF BIOGPT AND ITS IMPLEMENTATION WILL ALSO BE INCLUDED.

Chapter 4

DISCUSSION

Deep learning has become an overwhelming presence in our daily lives in just a few years, with applications ranging from self-driving vehicles to autonomous systems, even in cases where its use may not always be appropriate. Nevertheless, it has proven to be a helpful addition for automating tasks that were once extremely tedious and time-consuming. By acknowledging this addition as an invaluable tool, its capabilities can be leveraged to target certain tasks, allowing technology to expand its reach even into areas such as advanced medical imaging and diagnostics.

This study presents a comprehensive image analysis pipeline, integrating image segmentation, pattern recognition by an LLM, and an interactive web application with built-in BLIP and BioGPT capabilities. While many of the tasks within this pipeline have already been directed by deep learning techniques, the novelty of this pipeline lies in the way it strings together independent models and fine-tunes them for the task at hand. This pipeline 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 BioGPT. Instead, it allows for human annotation between the steps, ensuring greater accuracy and adaptability. Currently, many deep learning algorithms are designed to specialize in a single domain, such as language data or image data. Therefore, combining multiple models, each optimized for their specific task, provides the opportunity to tackle a task as complex as analyzing biological image data using their combined strengths. Of course, of the thousands of models available, there are other algorithms that could perform similar tasks. One that I highly considered was VisualBERT, an extension of BERT's language model (Bidirectional Encoder Representations from Transformers) that also specializes in performing image captioning. Many other models could replace or augment the analysis pipeline, and their accuracy and efficiency could be analyzed in further studies.

As seen in Table 2, although Cellpose performs well in segmentation, it is not perfect, and therefore, this data can benefit from additional verification. Additionally, it is recommended for users to train a Cellpose model on their images to improve specificity when performing segmentation. Quantitative analysis was not conducted on the outputs of BLIP or BioGPT since defining concrete accuracy parameters for language data is more difficult. However, it can be assumed that their outputs are not flawless. This pipeline grants the user the ability to step in and perform additional verification when necessary.

While a quantitative analysis was carried out, it is important to note that the results should not be taken as concrete facts. Artificial Intelligence learns from the data provided and generates conclusions, but differences in data annotation 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. Additionally, AI-generated outputs have been known to potentially include false information (hallucinations), particularly in scientific contexts. Therefore, artificial intelligence should serve as a tool to guide researchers to examine portions of their data more closely rather than replace crucial aspects of their work.

It's important to understand the rapid development of the machine learning and deep learning community. The world of image and language processing is constantly being updated with new models or new combinations of techniques that perform more efficiently and accurately or require less training data. This paper efficiently integrates relatively new technology (within the past few years) and strives to be a useful tool within image processing. In performing this study, I find that I am most impressed with the community and support surrounding this rapidly developing field. So many algorithms, models, and software are open-sourced, allowing individuals and companies to communicate openly, building off of each previous development. Fiji, and many others, is open source, not only regarding its source code but also in its relationship with other platforms, openly accepting suggestions on improving its features. Many companies and software even provide tools to allow people like myself to step into its world without a large background in using it. To encourage participation, many competitions exist, such as the Cell Tracking Challenge (CTC) (**CITE 25 and 26**), the Data Science Bowl (DSB) Challenge (**CITE 27**), and the Colon Nuclei Identification and Counting (CoNIC)

Challenge(**CITE 28**). In conjunction with this standard practice, my web application code, along with additional code and datasets, are available through links in Appendix A.

Chapter 5

CONCLUSIONS

Conclusion:

The initial motivation for the study resided in alleviating the time-consuming bottleneck of image analysis in biological research by developing an optimized pipeline for increased efficiency in pattern recognition. A comprehensive accuracy benchmark of machine learning and deep learning-based cellular segmentation software was conducted to understand the efficiency of pre-trained and fine-tuned models. Following that, a web application was established to simplify the visualization of segmentation files and a deep learning pipeline was included to establish a seamless flow from segmentation to automated image captioning and pattern recognition using an LLM. By linking Cellpose, BLIP, and BioGPT in a structured workflow, the goal is to allow researchers to analyze biological images more efficiently while leaving room for human verification.

Currently, this workflow is sufficient for pattern analysis but has great potential for increased specificity and future improvements. Additional fine-tuning of BLIP with greater diversity in the image and caption type would be highly beneficial to generating research-relevant data. Improvements to the web application could include the direct integration of Cellpose into visualization, with a larger selection of options for the visualization of individual images and masks. Additionally, an automated pipeline including only backend processing to funnel the data from segmentation through to BLIP and BioGPT could be beneficial while keeping the ability for human annotation between steps optional. Ideally, a future study would incorporate additional validation metrics for the language-based outputs to quantitatively analyze the validity of outputted data. This web-based application 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.

REFLECTION

Reflection:
WRITE 1000-2000 WORDS OF REFLECTION

Both scholarly and creative/design papers need to include a section of 1000 – 2000 words for the SW narrative that articulates how the 3 thematic courses and the outcomes of the capstone courses contribute to the students' SW project. Students should reflect on the SW experience and how it prepared them for future goals.

3 thematic courses

- **Web Design (Multimedia 2..) for coding skills and the idea of a web application**
- **Computational Genomics at Duke (Comp260) for history of algorithms**
- **Bio 304 with Linfeng Huang for microscopy and using software to analyze it**

SRS

- **Working with prof huang**

Bagnat/Di Talia Lab

- **Working with cellpose and actually understanding the importance of it in a real context.**

REFERENCES

Nature Format: Authors. Title of paper. *Journal Name* volume, page numbers (year).

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
    - Biomarker paper

2. 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
    a. Quantitative analysis of existing cellular segmentation programs

3. 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
    a. Cellpose

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

5. 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

a. CellProfiler

6. 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
    a. Ilastik

7. 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
    a. Fiji

8. 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
    a. ImageJ2 update

9. 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
    a. ImageJ 25 years paper

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

11. 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.
    a. ZEISS book

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

13. Hou, B., Qin, L. & Huang, L. Liver cancer cells as the model for developing liver-targeted RNAi therapeutics. *Biochem. Biophys. Res. Commun.* **644**, 85–94 (2023). https://doi.org/10.1016/j.bbrc.2023.01.007
    a. Linfeng huang paper

14. Huang, L. & Lieberman, J. Production of highly potent recombinant siRNAs in *Escherichia coli*. *Nat. Protoc.* **8**, 2325–2336 (2013). https://doi.org/10.1038/nprot.2013.149
    a. Methods paper

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
    a. Gaussian blur

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

17. 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
    a. BLIP-2

18. 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
    a. Querying Transformer

19. 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

20. 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

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

22. 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
    a. DeepCell

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

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

25. 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.

26. 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

27. 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.

28. 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.

Appendix A

ADDITIONAL MATERIAL

Appendix:

Add figures and code?

Some weird template for appendix on the sw template pdf

Github link:

Web application link:

Jupyter Notebook:

Annotated truth files:

Additional Codes