

2021T3 COMP9517 Group Project - WandaVision

Harry Brink - z5117928

Arun Kumar Marndi - z5225111

Samir Mustavi - z5084519

Brett Samuels - z5117709

Elizabeth Willer - z5161594

November 20, 2021

1 Introduction

As of 2021, one of the major biomedical challenge we face as a society is developing a cure for cancer. Cancer as a disease is prolific. Each year approximately 50,000 Australians die to various cancers [1]. The behaviour of cancer is patient specific and highly variable depending on the location within the body. Current accepted treatments include radiotherapy and chemotherapy and are effective for some cancers such as breast cancer, however, ineffective for others.

The development of new cancer treatments that are effective for all types is essential and predicated on our knowledge of the behaviour of the disease throughout the entire body. To further this understanding, analysis of the disease's cellular dynamics at a single cell level is vital. To analyse cellular dynamics microscopy images are taken over a specific time-frame and the inter-cellular behaviour of each cell tracked and monitored throughout the period. Manually this task is cumbersome and time-consuming. Cell segmentation and tracking using computer vision algorithms is a means in which to automate this process and provide accurate and precise information on the specific cellular dynamics.

The primary aim of this report was to develop a program in python which would take a sequence of microscopy images on cells, preprocess and segment the images. Individual cell motion on a micro scale as well as cell size and count on a macro scale was then

to be conducted. A key point of interest was identifying cells undergoing mitosis and continuing track of the resulting daughter cells.

Section 1 provides an in-depth literature review on the current computer techniques used for cell segmentation and tracking. Conclusions will be drawn on the type of computer vision methods to be used in the software. Section 2 outlines the exact methodology used within the program to provide accurate cell segmentation and tracking as per the specifications. The results will then be outlined in Section 3.4. Section 4 will discuss the results obtained and identify shortcomings and areas for future work and improvement. Overall report conclusions will be made in Section 5. The contribution of each group member to the project will then be outlined in Section 6.

2 Literature Review

The use of Computer Vision techniques for tracking and analyzing cells from the microscopic images has been one of the important research topics in the field of Biomedical Imaging, Image processing and Computer Vision. The researchers have devised many state of the art techniques to solve this problem.

Various preprocessing techniques have been used to process microscopic images. Researchers have devised various methods to keep track of cells, detect mitosis and calculate the average area of the cells.

The mitosis of a cell can be detected by observing various visual properties[2]. These properties include key features such as Circularity,Perimeter,Equivalent Diameter,Circularity,Minor to Major axis ratio [3].

However, with the current trends in state of the art technologies such as Neural Network,Deep Learning and Artificial Intelligence, the Neural Network,designed by researchers,is able to train itself from the list of image frames and extract the necessary features to detect mitosis among the cells.[4].

Though artificial and deep learning methods have provided good results, it requires a lot of computational power and expensive hardware needed to train those models and perform tasks.[5]

So, we follow the basic geometric properties of cells to identify which cell is undergoing mitosis and which isn't. We use basic computer vision techniques to detect mitotic division among cells. We fit an equivalent ellipse around the cell in our project using the cv2 function *fitellipse()*[6] and use the properties of ellipse to detect mitosis.This can be achieved by determining the minor to major axis ratio of this equivalent ellipse[3]

3 Methods

Four main steps were taken to analyse the images:

- 1 Preprocessing
- 2 Segmentation
- 3 Tracking
- 4 Detecting Division

Each of these has its own methods that were used. These are explored in the following sections.

3.1 Pre-processing

The image goes through 3 stages of preprocessing. Firstly, closing eliminates any gaps in the cells. Then contrast stretching and binary threshold is used so that the cells have a value of 1, and the background has a value of 0.

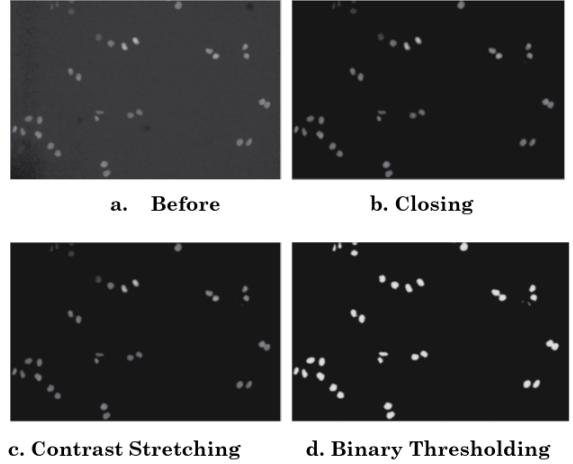


Figure 1: Different stages of preprocesing

3.2 Segmentation

Segmentation was performed by using the cv2 functino *findContours()* on the preprocssed images. This segmented result was provided to the tracking method.

3.3 Tracking

A list of cells is kept, with each cell having an individual object. For every one of these, the closest cell in the new image is found and the similarity of these cells is checked using match shape. If they don't match, the cell has been deleted. Finally, all of the cells in the current image that weren't picked up by the previous step are added (these are new cells).

Originally, *matchTemplate()* was used for checking the similarity of the cells. However, this function is computationally heavy, and depends on scale and rotation. As the cells will likely increase in size and rotate slightly, a scale and rotation invariant method should be used. Therefore, it was replaced with *matchShape()* which is scale and rotation invariant. A comparison of the results provided by these these two functions is given in figure 2. In future, if computational complexity was not a point of interest matchTemplate may give a better result by comparing the features of the cell interiors. However, this would require two additional nested exterior loops to

iteratively rotate and re-scale the template.

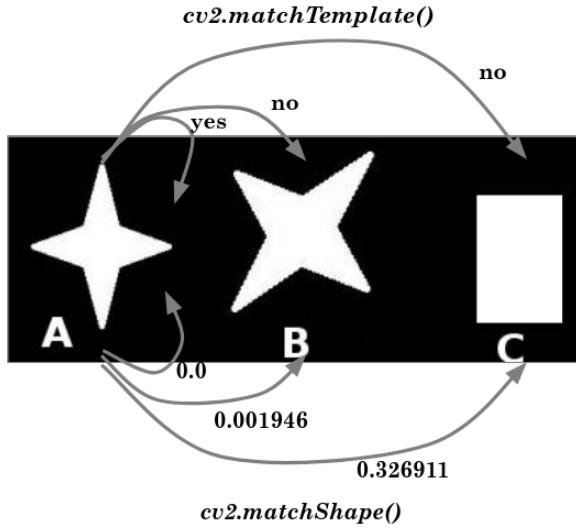


Figure 2: Comparison of `matchTemplate()` vs `matchShape()`. Shapes from [7]

3.4 Detecting Cell Division

There are several different options for detecting mitosis. As mitosis occurs the cells lengthen. Thus, mitosis can be detected by checking how elongated the bounding oval is. This can be done either by checking the eccentricity of the bounding oval, or the ratio of length to width, as seen in figure 3. Furthermore, as mitosis occurs, the center of the cell pinches together. This can be detected by searching for the presence of concave sections. Alternatively, the area of the bounding oval can be compared with the area of the cell, as seen in figure 4. In a dividing cell, the ratio will be substantially lower.

In this project, it was chosen to use the ratio of the major axis to minor axis. For more accuracy, several methods could be combined.

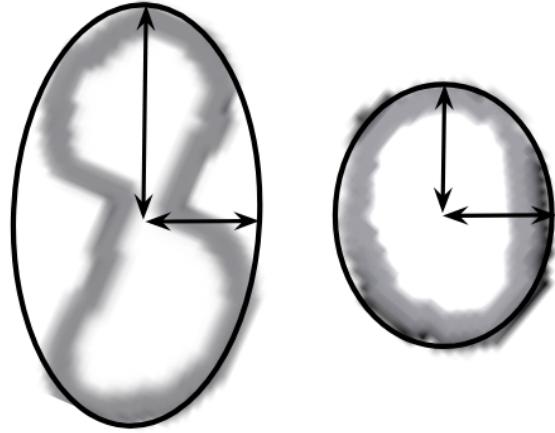


Figure 3: Ratio of major axis to minor axis of a dividing vs non dividing cell

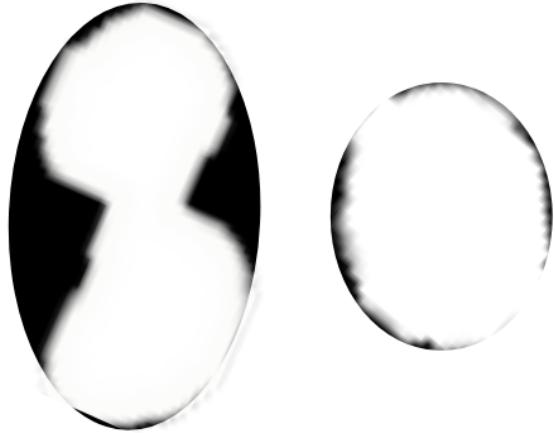


Figure 4: Area of a dividing vs non dividing cell

4 Experimental Results

The following figures show the first frames of all image sets, respectively showing 33, 125, 45 and 90 cells initially in image sets 1, 2, 3 and 4.

To assess the success of the algorithms used for this particular project, the following sets of metrics were calculated for our model:

1. % of cells that were correctly **pre-processed** using background removal techniques on the first frame
2. % of cells that were correctly **contoured** from our cell segmentation function on the first frame

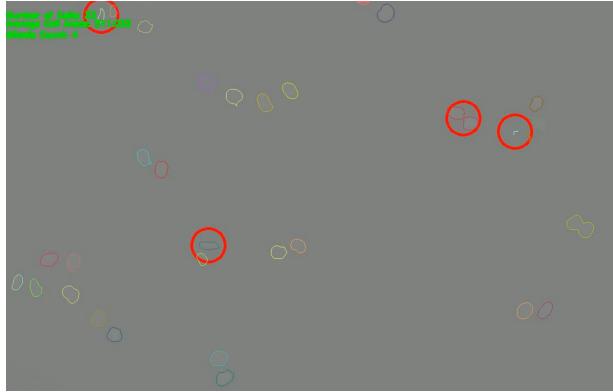


Figure 5: First frame of cell contours in image set 1

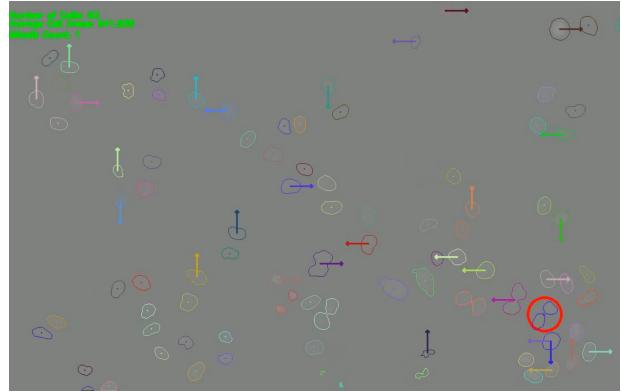


Figure 8: First frame of cell contours in image set 4

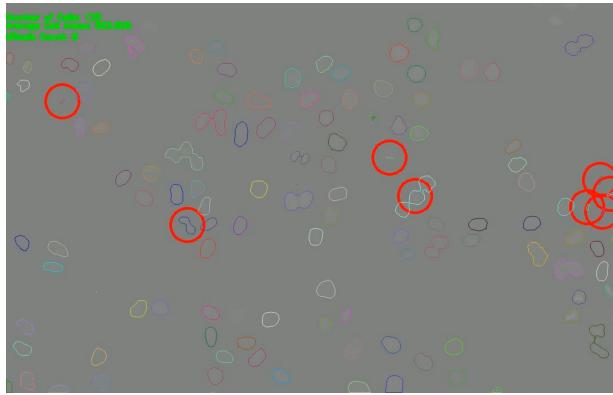


Figure 6: First frame of cell contours in image set 2

accuracy) as shown in the figures below.



Figure 9: Original first frame in image set 1

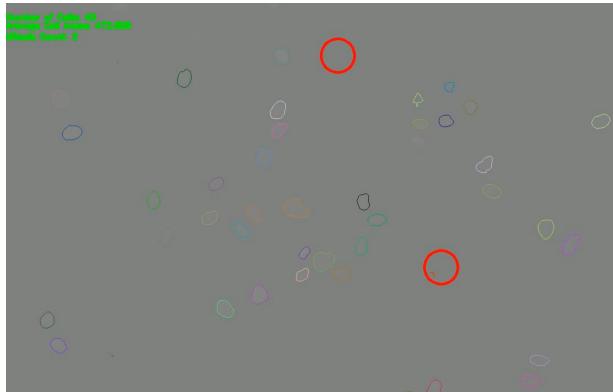


Figure 7: First frame of cell contours in image set 3

3. % of cells that were correctly **identified as having undergone mitosis** over the course of the given image set.

Upon manual inspection, it was revealed that 37 cells could be identified in the first frame of the first image set, of which 35 were correctly pre-processed (94.6%

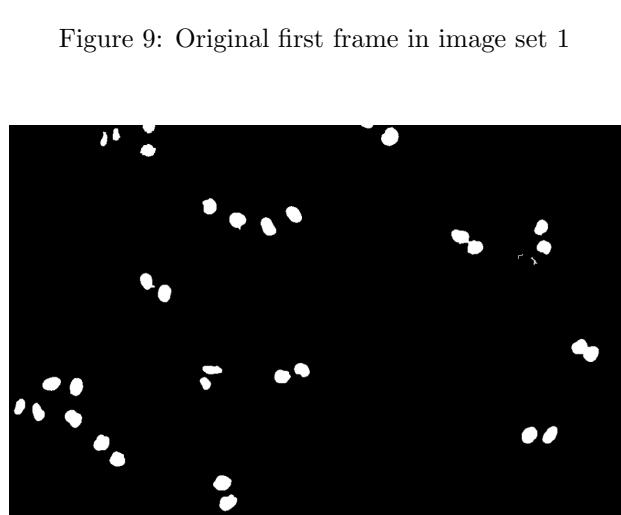


Figure 10: Pre-processing applied to the first frame in image set 1

From this, the segmentation function helped to properly contour 33 out of the 35 pre-processed cells (94.3% accuracy) in this given frame.



Figure 11: Segmentation applied to the first frame in image set 1

Finally, upon manual inspection of different images across the image set, we identified a total of xx/yy correctly identified instances of mitosis. A few examples of frames with mitosis being identified are shown below.

The following metrics were calculated for all the remaining image sets, with the following pre-processed and contoured images displayed below.

The final metrics are displayed in the table below.

Image Set	1	2	3	4
Metric 1	35/37	128/133	45/47	91/92
Metric 2	33/35	125/128	45/45	90/91
Metric 3	98/118	146/198	134/168	130/155
Total	74.1%	69.3%	76.4%	82.0%

Table 1: Final metrics of success for cell identification and tracking

5 Discussion

Image Preprocessing

In regard to image preprocessing, there were no notable issues. The image background was consistently contrast stretched to black and the cells were consistently stretched to white which successfully prepared the image for easy and effective segmentation.

Image Segmentation

The image segmentation component proved to be relatively straightforward due to the effective prepro-

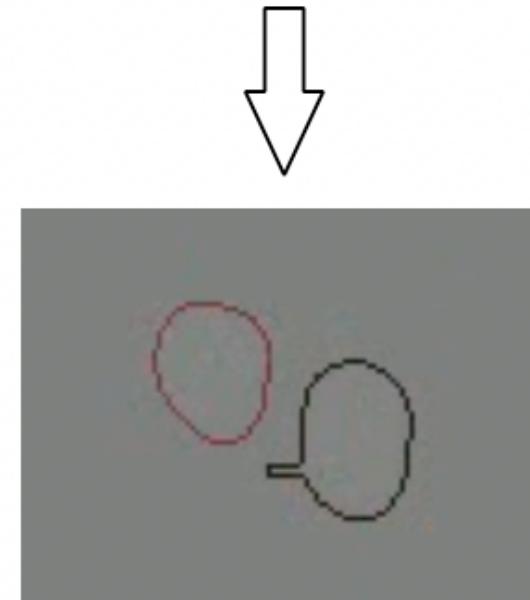
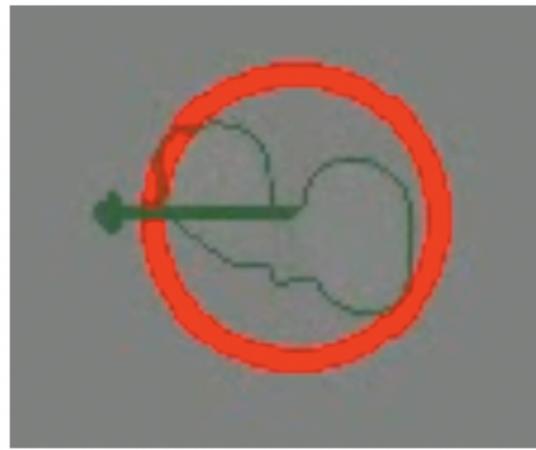


Figure 12: Closeup of a cell identified as undergoing mitosis in one frame, and completely split in the next

cessing. The use of OpenCV methods `findContours`, `moments` and `contourArea` proved to be an effective method for finding cell contours and centroids.

Cell Tracking

In regard to Cell Tracking there were notable issues encountered when attempting to identify previously registered cells in new images. The method of finding the nearest last registered centroid appeared to be mostly effective, however, attempts to improve its accuracy by coupling this method with

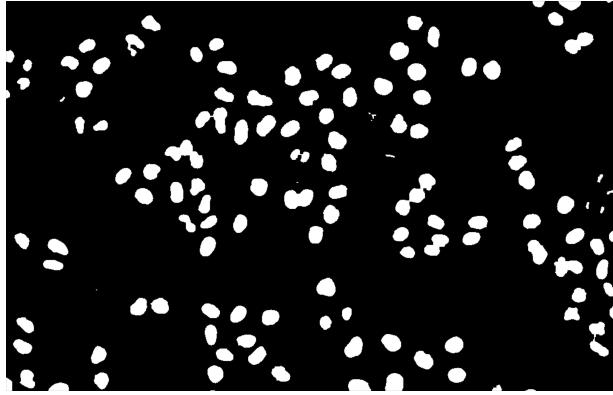


Figure 13: Pre-processing applied to the first frame in image set 2

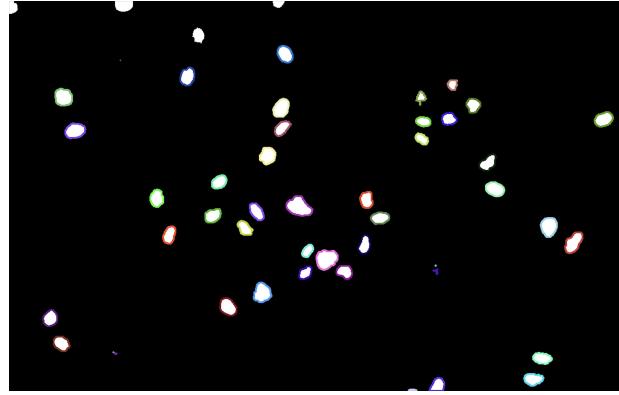


Figure 16: Segmentation applied to the first frame in image set 3

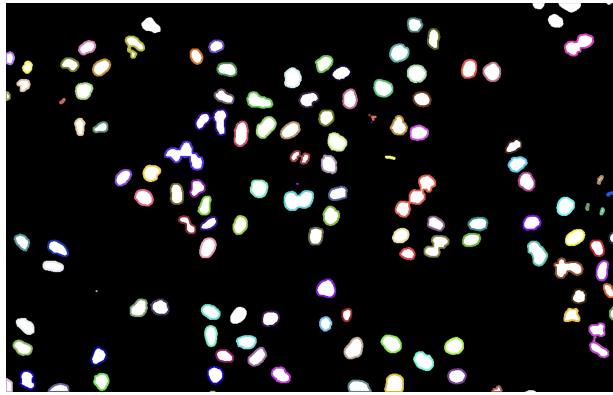


Figure 14: Segmentation applied to the first frame in image set 2



Figure 17: Pre-processing applied to the first frame in image set 4

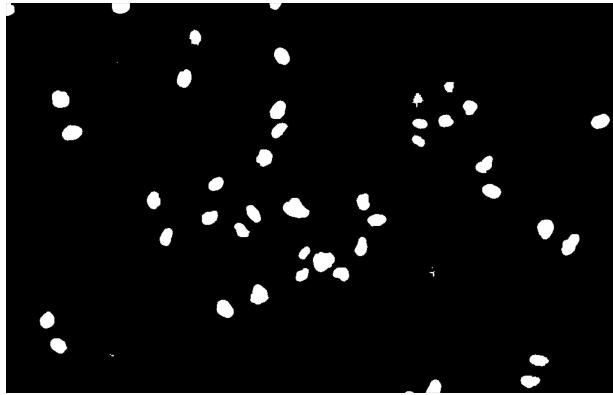


Figure 15: Pre-processing applied to the first frame in image set 3



Figure 18: Segmentation applied to the first frame in image set 4

the OpenCV matchTemplate function generated issues. The matchTemplate function is computationally costly and its accuracy is affected by template scale and rotation. However, this function was later

replaced by OpenCV matchShape which is scale and rotation independent.

In future, if the computational cost issue is solved by using more suitable hardware for the task, the

matchTemplate function is likely to generate a more accurate result by comparing the features of the cell interior before filtering. However, the cell tracking algorithm would need to be altered to rotate and resize the cell before matchTemplate is applied.

Cell Motion Analysis

The majority of the cell motion analysis functions were straightforward and effective due to the solution's object-oriented structure.

There were, however, issues encountered with drawing contours from contour elements taken from the data structure returned from cv2 findcontours. These issues were resolved by storing the contour index of the identified cell in the original findcontours data structure and simply drawing by index using cv2.drawContours. In regard to detecting cell division, there were notable issues related to finding the eccentricity of the cell. A more robust approach would be required in future. Although there were a reasonable number of complications encountered, these issues were relatively superficial and would be overcome with little further experimentation. The underlying structure and design of the final product was adequate overall in completing the required task.

6 Conclusion

The push for advancements in biomedical research is greater now more than ever. The disastrous effects of diseases on the human population have been observed throughout the past two years. Cancer is a key concern for biomedical researchers as it causes approximately 50,000 deaths per year in Australia alone. Understanding of cellular dynamics is a key focal point for biomedical researchers to enhance the fundamental behaviour of all cancer types. The automated analysis of microscopy images using computer vision techniques is essential for furthering our current knowledge. An automated framework allows thousands of frames of microscopy cell images to be analysed within minutes and provide detailed micro data on the individual cells and macro data on the cell

population. This extensive analysis is unable to be computed at the same volume and precision manually by humans. This report aimed to use conventional computer vision techniques to create a program which pre-processes microscopy cell images, undertakes image segmentation and provides micro cellular data and macro cell population data. The identification of cells undergoing mitosis and the tracking of the resulting daughter cells was a key focal point. An in-depth literature review was conducted, and current accepted methodologies were compared and analysed. Methodologies to be used in our program were identified and their specific use case and functionality was outlined and described. The program was developed and provided image sequences were tested with resulting data being output. The output results were analysed shortcomings and errors were discussed. Future works to enhance the program and the automation of cell tracking using computer techniques were outlined. The automation of cellular dynamics analysis on microscopy images was confirmed in this report. An accurate program to analyse sequences was developed.

7 Group Contributions

Harry: I was responsible for writing the code for pre-possessing raw images, counting the number of cells in each frame, calculating the average area of all cells in each frame and generating an mp4 from the finalised annotated images.

Arun: My responsibility includes writing code for mitosis detection, debugging every member's code. Writing literature review and software demo presentation.

Samir: I was responsible for collaboratively writing the code for the function that segments the image and returns the contours and centroids of each cell present and the code for the function that draws the cell trajectory arrows onto the images.

Brett: I was responsible for collaboratively writing the code for the segmenting function with Samir, and

the code for the function that draws the contours onto the images.

Elizabeth: I was responsible for tracking the cells as they moved, and finding the average displacement.
[8]

[8] S. Ahmad and B. Fuller, “Unconstrained iris segmentation using convolutional neural networks,” in *Computer Vision – ACCV 2018 Workshops* (G. Carneiro and S. You, eds.), (Cham), pp. 450–466, Springer International Publishing, 2019.

References

- [1] Cancer Council, “Facts and figures,” <https://www.cancer.org.au/cancer-information/what-is-cancer/facts-and-figures> (Accessed 16/11/2021).
- [2] S. Huh and M. Chen, “Detection of mitosis within a stem cell population of high cell confluence in phase-contrast microscopy images,” in *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pp. 1033–1040, IEEE Computer Society, 2011.
- [3] T. Bandyopadhyay, J. Kubiček, M. Penhaker, J. Timkovič, D. Oczka, and O. Krejcar, *A Semi-Supervised Learning Approach for Automatic Segmentation of Retinal Lesions Using SURF Blob Detector and Locally Adaptive Binarization*, pp. 311–323. 01 2019.
- [4] A. Sohail, A. Khan, N. Wahab, A. Zameer, and S. Khan, “A multi-phase deep cnn based mitosis detection framework for breast cancer histopathological images,” *Scientific Reports*, vol. 11, p. 6215, Mar 2021.
- [5] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, “The computational limits of deep learning,” 2020.
- [6] OpenCV, “Creating bounding rotated boxes and ellipses for contours.” https://docs.opencv.org/3.4/de/d62/tutorial_bounding_rotated_ellipses.html.
- [7] OpenCV, “Contours : More functions.” https://docs.opencv.org/4.x/d5/d45/tutorial_py_contours_more_functions.html.