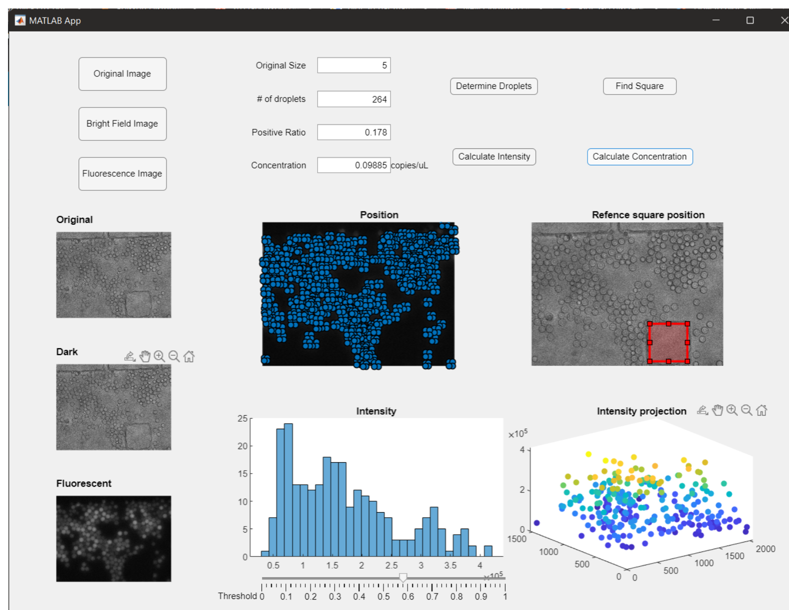
Design of A MATLAB Based Droplet Auto Counting and Analyzing Program

UNIVERSITY OF TWENTE

BACHELOR THESIS ELECTRICAL ENGINEERING



Committee chair
Prof.dr.ir. L.I. SEGERINK

Committee member
Dr.ir. J.T.W. BERENDSEN

Committee member
J.E. VAN DONGEN MSC

External member
Dr.ir. L.J. SPREEUWERS

Author
L.Li

December 1, 2022

Abstract

Up to now, far too little attention has been paid to develop a computer program that detects and analyses fluorescent droplets for early-stage cancer detection. In this project, we present a design process of a MATLAB based auto-counting of the number of droplets, detecting their location, and applying intensity and concentration calculation using the Canny detector and Hough transformation utilized for early-stage cancer detection explicitly. Meanwhile, we designed a GUI (graphical user interface) as the operating system. We evaluated our product both scientifically and ethically. The scientific evaluation indicates that the product shows an excellent performance even compared to the most frequently used commercial product ImageJ, it performs better in terms of detecting low-intensity droplets and the result of intensity calculation is almost identical. To achieve this, we investigated different methods for droplet determination, including algorithm and AI (Artificial intelligence) programmed in Python language as well as the final product based on MATLAB platform but only verified on a small sample range. Ethically, the GUI provides users with a better user experience from the product that they would not be scared of the long programming codes or much prior programming required to get on with the program.

,

CONTENTS

I Introduction 3
I-A Objectives 4

II Theory 6
II-A CRISPR 6
II-B ddPCR 7
II-C The current method 9

III Design 12
III-A Matlab 12
III-B Design steps 13

IV Results 24

V Conclusion and Outlook 29

References 32

Appendix 33
A Code 33

I. Introduction

The aim of this project is to design a MATLAB based droplet auto- detecting and counting program, it preforms analyzing on the collected data. Using droplet detection has recently emerged DNA analysis assays. It allows researchers to detect anomalies in DNA for disease analysis, including cancer detection. [1]

Detecting the changes in the cell DNA (deoxyribonucleic acid) can be an excellent way of monitoring the pathological changes. As the diversity of the trait of creatures that holds biological information is encoded in DNA. [2] It consists of various components, a portion of which is passed from parent organisms to their offspring during the reproductive processes.[2]

Among the various techniques that have been used for DNA detection, droplet digital polymerase chain reaction (ddPCR) is a method that performs digital PCR technique in water-oil emulsion. It allows the polymerase chain reaction to take place in a small volume which the droplets provide.

The high sensitivity and accuracy allow digital droplets be the most preferred screening tool for clustered regularly interspaced short palindromic repeats (CRISPR) engineering.[3] Among the CRISPR system, Cas12a is specifically a cas protein that can be used for its collateral cleavage. It is a technology used for DNA diagnostics, which sense the cancer bio-markers by the CRISPR/Cas system.

The solutes are Cas12a, crRNA (crispr ribonucleic acid) and fluorophore-quencher pairs that are dissolved to create the solution where the Cas12a cuts the fluorophore-quencher pair when the crRNA find complimentary double stranded DNAs that are called the target DNA. Once the CRISPR has cut off the quencher, the fluorescent signal will be measured.[4] Figure 1 is the schematic of quenched fluorescent reporters cut by Cas12 when the target DNA is found. This is correlated to the concentration of target DNA which is non-stationary.

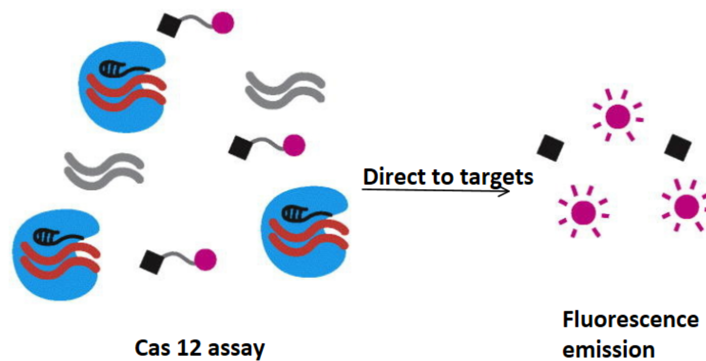


Figure 1: Schematic of the quenched fluorescent reporters cut by Cas12 when the target DNA is found

A. Objectives

However, little research to date has focused on developing a tool that can recognize and automatically count the intensity in a time-variant scale.[5] In order to solve this challenge, the goal of this project has been broken down into several objectives.
Prior to this project, the experimental stages of using CRISPR Cas 12a to cut DNA between fluorophore quencher pair to obtain some fluorescence have been performed. But to improve the analysis of the data (i.e. the fluorescent droplets), a program should be designed in order to analyze the data. To better understand and solve the problem, we have set three primary goals of this project:

1) To investigate the current image processing method in droplet and cell counting,
2) To ascertain an efficient algorithm that allows program to detect and to automatically count the intensity and the number of fluorescence droplets,
3) To develop a user-friendly terminal user interface.

The first one ensures a good understand of current algorithm and investigates their pros and cons. The second goal is a technique approach to solve the problem. The last aim is more related to human and technology relation, as the ultimate aim of product is letting users willing to use it, hence, we have defined such an aim.
There are some more specific requirements should be addressed in order to avoid false negative. Because if the test results are false positve, there will be some future examinations, however, the false false negative will prevent bringing enough cautions to the patients in time.In this project, we aiming to achieve a detection sensitivity over 90%

and false negative ratio lower than 5%. In this case, most droplets should be detected so that it can provide a solid basis to predict whether the object has a positive test result.

The remaining part of the report proceeds as follows: in the II.Theory part, it will contain the essential biology knowledge and a comparison of the existing method will be delivered, while the next section III.Design is concerned with the methodology used for this project, the development and architecture of the design options along with outcomes in the IV.Results section. The V.Conclusion and Outlook section examines if the final output has met the criteria, gives suggestions to the future work and summarizes the entire work, reflect on the introduction.

II. THEORY

This part of report begins by laying out the theoretical dimensions of the research to understand the components of the detection method, the mechanism for detection that is used: The CRISPR. It will then go on to the ddPCR as long as the most important criteria, the concentration. Lastly, we focus on how the recent investigations work out the problem.

A. CRISPR

CRISPR is a powerful gene/DNA-editing system. It allows researchers to screen for disease, add and delete genes in a rapid and precise manner, which makes genome diagnosis faster and more accurate.

The Cas protein was originally found in bacteria, where they help defend against viruses.[6] When bacteriophages attack, they inject their DNA through the bacterial cell membrane. CRISPR is one method of bacteria developed as an immune defence.[7] Every time the cell infects with a virus, it knocks-out a bit of the invading bacteriophage DNA and stores it. In this way, when the bacterial encounters the same virus, it will be able to recognize the invader.[7][8][9]
This bacterial defence system has been modified as a utility genome editing tool. It is not limited to bacteria but has proven its effectiveness in almost every cell, from bacteria to mice and human beings.[2] In 2012, a group published a paper that states the fact that Cas9 can be guided to a specific region in the DNA sequences.[8] In this project, the Cas12 has been used. It has been designed to recognize the cancer bio-marker (target DNA).[9] Once the Cas12 is directed to the target, it will bind to it and cut the bound between fluorophore and quencher pairs. This process can is presented in the figure1.

In this experiments, the CRISPR plays a role as the scissors to keep the fluorophore-quencher pair farther apart. A dark quencher is in conjunction to a fluorophore via a ssDNA part. It absorbs energy and decreases the intensity emitted light from fluorescent molecules.[10] When a pair of fluorophore and quencher are far apart, there is fluorescence presented, if the reporter and quencher are close together, the fluorescence is suppressed.[10] This fluorescence simultaneously reveals the existence of the target and confirms the identity, preventing the false positive of the non-amplicon. It can be easily detected with a simple UV transilluminator. Such a characteristic is the key ability to achieve a good fore/back-ground ration in optical images as what is displayed in the figure 2.

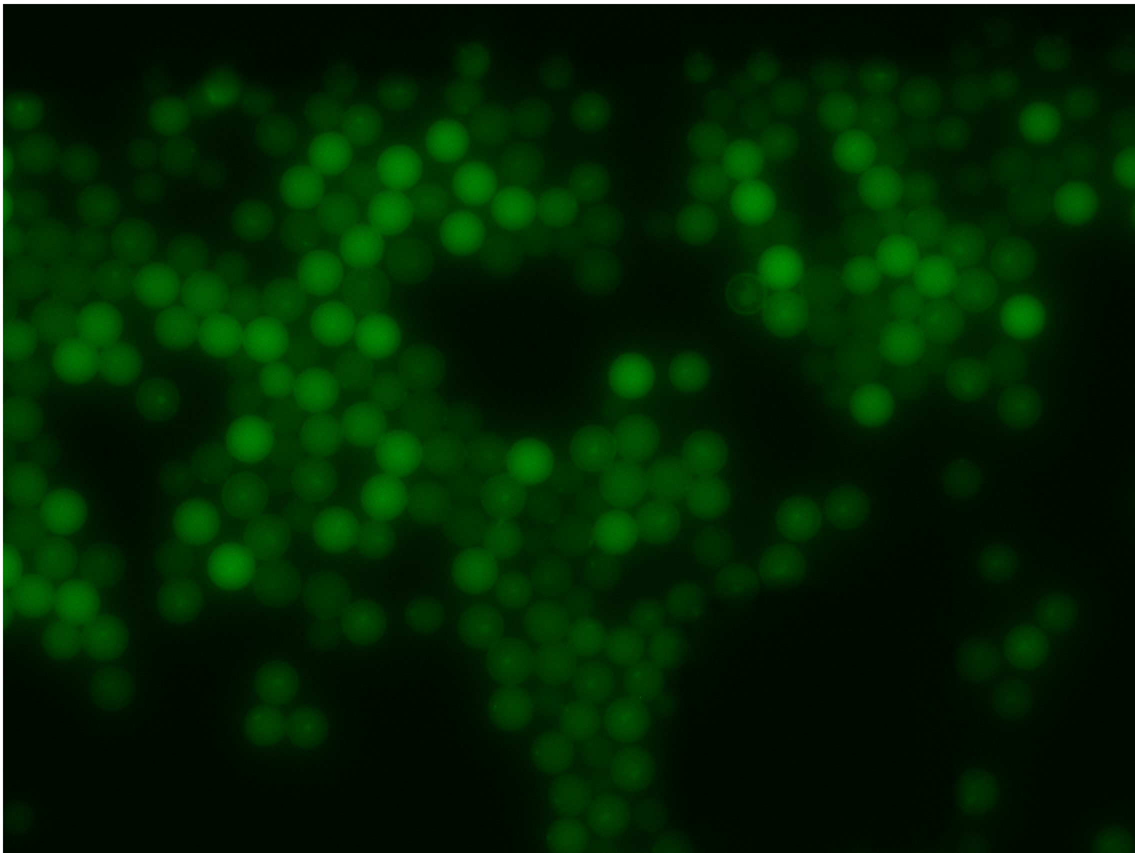


Figure 2: Fluorescent image of Cas12 reaction

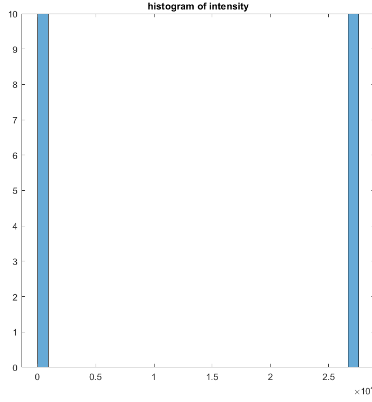
B. ddPCR

The ddPCR digitized the samples into discrete magnitudes, which allows the detection and any analysis possibilities. Compared to the PCR solution, in a wider dynamic range, ddPCR is more precise, which is great for fractional abundance. ddPCR increases signal to noise ratio, provides higher unparalleled precision and removes amplification efficiency that relied on PCR. Digital droplet PCR measures absolute quantities by counting molecular in a quantized water oil droplet partition. More droplets represent more data points and more accurate results at the end of the amplification. Overall, it is more quantified in terms of low concentration tests due to the partitioning.
Since droplets are used, the Poisson distribution describes the way the molecules are distributed. (a representation of Poisson distribution is in the figure **??**) The Poisson distribution is commonly used to express the distribution of rare events in a big samples set, it is best represented to statistical analysis for a countable variable with known constant mean and the arrive of events should be independent, i.e. the waiting time of each event is memoryless. In our case, target molecules are independent, means

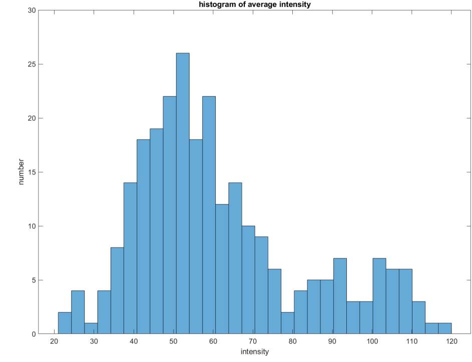
no interaction between each other. The concentration of the fluorescence signal in the ddPCR solution well can be predicted by the Poisson distribution, as shown in the eq(1).

$$f(x|\mu) = \frac{\mu^x}{x!}e^{-\mu};\ x = 0,\ 1,\ 2, ... \tag{1}$$

where μ is the average number of the droplets with an amplicon which have fluorescence signal, x is the number of prediction.



(a) Illustration of ideal intensity plots where positive concentration λ at 26k $copies/\mu l$, and zero intensity implies that those droplets are negative



(b) Illustration of Poisson probability density function for the concentration λ at 51 $copies/\mu l$ with some deviation where the overlapping between the positive sets and the negative set will make is difficult for researchers to derive the positive/negative ratio

Figure 3: The ideal situation for the droplets concentration is on the left side with a constant concentration of 26k $copies/\mu l$, while the real case is demonstrated on the right hand side with different concentration of different droplets sizes

The molarity (molar concentration) defines amount of substance of solute per unit volume of solution, represented by c, as it in formula (V),

$$c = \frac{n}{V} \tag{2}$$

in which, n is the amount of the solute in moles, V is volume of solvent.

Despite of all the convenience of the ddPCR, bearing in mind that the concentration of ddPCR is not always uniform as what shown in the figure 3a, where the negative test results have zero intensity while the positive ones have a high intensity. When the droplets were heat, the size of each droplet would shrunk, which causes the various intensities that will correspond to the different concentration as demonstrated in figure

3b, it working priciple will be explained in details in the III.Design section. Concentration can be future developed as the copies of target divides by the sample size that is demonstrated in eq. (3),

$$c = \frac{copies\ of\ target}{volume\ analyzed} \tag{3}$$

where the volume analyzed can be calculated using the volume formula of spherical shape with radius r.

$$Vol = \frac{4}{3}\pi r^3 \tag{4}$$

Poisson distribution can predict how many copies to expect in each class statistically, since there are randomized partitions of target molecules in each droplet copies. Hence, we used the following equation to determine the concentration,

$$c = -\frac{ln(N_{neg}/N)}{Vol_{avg}} \tag{5}$$

C. The current method

As mentioned in the introduction, the development of the analyzer aspect is delayed. Prior to discussing how to detect the ratio automatically, it is important to emphasize some existing algorithms. the first product gives an inspiration on the algorithm, while the second product has a better detection mechanism.

1) FluoroCellTrack[5]

FluroCellTrack(FCT) is an algorithm designed for automated analysis of high-throughput microfluidic data from different droplet microfluidic. This algorithm uses a pipeline structure that processes a wide-ranging data array. To start with, it reads 16 bits batch-fed images. Secondly, it removes the noise from images as a pre-process, then applies both droplet and contour detections. Last but not least feature is the post-processing that extracts vital information from the data set. Any demanding results can be exported for the future analysis.

The software is developed based on Python 3.6.0, where OpenCV and Python image library are included. Comparing to C++ or Java, Python consists of fewer steps and allows including OpenCV which makes it easier to apply machine learning for real-time computer vision applications in pattern recognition, event detection and artificial intelligence.[11] This algorithm is develop in a way that can process overlay fluorescence and bright-field images. Meanwhile, it automatically

- differentiates droplets,
- differentiates and numbers the live, dead, and invalid in each droplet,
- recalls the co-encapsulation information, and

- quantifies the intracellular fluorescence at a rapid speed.

However, every time the algorithm scans all the data but only recognizes one cell. It limits the analysis to a single cell at a time, which is time-consuming. In order to determine the droplets concentration and derive positive ratio, a large data set will be collect. The data set normally contains 12,00-16,000 droplets, which requires the FCT to scans this the same amount of data image to recognize all the droplets.

2) CellProfiler[12]

CellProfiler (CP) is an open-source software that developed by the Broad Institute Imaging Platform. Together with its companion CellProfiler Analyst (CPA), a supervised machine learning to classify the dataset in two categories, i.e., positive and negative, their intuitive user interface allows users without programming experience to determine characteristics of fluorescence microscopy images.

CP can read more than one hundred image file formats from lossless to lossy, however, the TIFF format is the most preferred ones.[13] As its pixel has been kept stable and suitable for the scientific image data.[14] The drawback of such a format is that comparing to other lossless form of file compression, the detailed and high-resolution picture leads to a large file size. It is hard to share or send. After the figure is converted to grey scale, the CP identifies the foreground and background using the optimal thresholding methods to produce a binary mask of edges. As the mask is covered on the entire image, the extreme values may affect the threshold, which requires user to tune it manually. In reviewing the literature, a case of analyzing droplet digital quantification of viable fluorescent bacteria, the value varied from 0.021 to 0.024. They used more than thirty images to find the most optimized threshold and sensitivity of the determined the method.[12] It can be considered that the sensitivity and thresholding is depended on the strain, droplet incubation time, hardware and labelling techniques. Meanwhile, CP requires external module for multi-colour fluorescent figures.[13]

The segmentation method that CP used does not include the droplets lie outside or on the cropped image diameter. It causes the fact that CP may identify fewer droplets. Other reasons why the counted number declines are that droplets have variated size, low image resolution or intensity. To solve such a problem, it demands users to obtain uniform and good quantality droplet with high image quality, which may not always be the case in real life.

Once the data is exported from CP, it is imported to CPA for analysis purposes. With hundreds of randomized droplets training data, it results in a model that is able to identify the two classifications as mentioned above. The amount of training data is

relatively low comparing to traditional computer vision training set of a library that contains over thousands of data. As the learning curve nor neural net architecture are not demonstrated by the developer, it is hard to evaluate the model skill. However, a report from Estonia has shown that 200 profile pictures are skilful, and the outcomes is able to show the Poisson distribution as expected.[12]

3) Comparison and inspiration

Machine learning algorithm ais the core of the two products, however, the CP has a better learning efficiency, but the FCT has a better error correct system as the steps are performed under the user's instructions. the outcomes of the two products are various based n the input data, as the biggest challenge in the field of computer vision is that there is no one for all solution can be established.

The two products can inspire our future work. They construct a work flow of image pre-processing and stimulate the recognition algorithm's developments by using canny detector and Hough circular transform.

III. Design

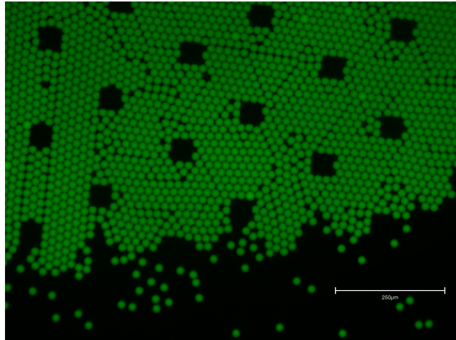
In this section, we firstly explained why MATLAB is chosen as the platform to create the software, then we broke down the product processing workflow into steps.

A. Matlab

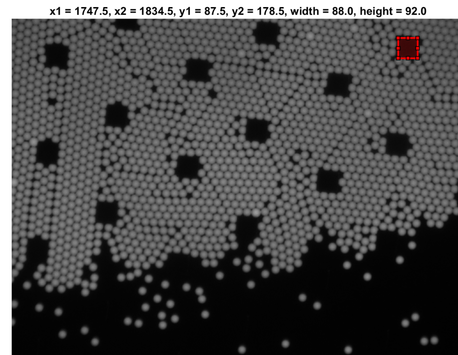
Eventually, MATLAB approach was used for the final product. As this product is going to be used mainly in the biology domain. The program tastes and skills of researchers are a great feature to consider. Even though Python has a large community and has developed libraries like OpenCV, which makes it an open-source, powerful and expanding language. The design idea was first made in Python. MATLAB is lean towards academics, thus, it avoids integration with industry-standard platforms or applications. Meantime, it has extra costs. However, it takes more time for biology people to learn Python, a new programming language. As it has more advantages in machine learning or data science. Those are beyond the regular scopes of biology researchers. The image processing in MATLAB is more elegant and easier to use, learn and expand because of professionally curated packages. OpenCV can be the strongest aspect of Python in terms of image processing. But it is not competitive with the image processing toolbox from MATLAB. Since it does not feel natural for users to access the largest libraries of image-related algorithms, the toolbox is more efficiently implemented in MATLAB. Moreover, the graphics libraries in MATLAB are superior to Python's. There is basically zero cost to integrate all the program into the GUI in MATLAB. The syntax of MATLAB is so unique and inherent that links to the linear algebra and DSP, the fundamental of the image processing algorithm. With such a powerful connection, it can be easy to understand by almost any scientific worker. It comes with high computation costs.

To the aspect of method's qualitative and quantitative in this investigation, the MATLAB approach appears to be a better choice over Python. Because it is more packed and easier to use for biology users, one more thing to accentuate is that even though a number of techniques have been developed, there is no MATLAB based product.

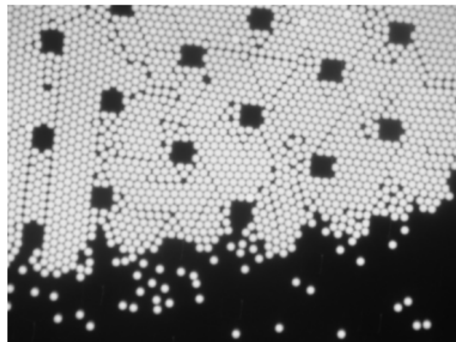
B. Design steps



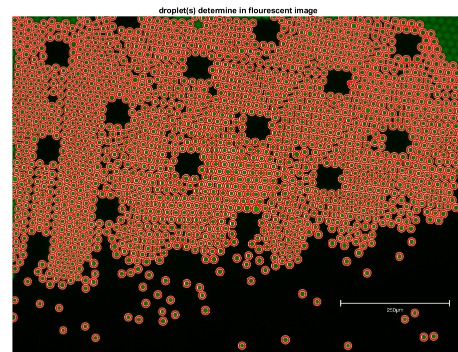
(a) The original fluorescent image



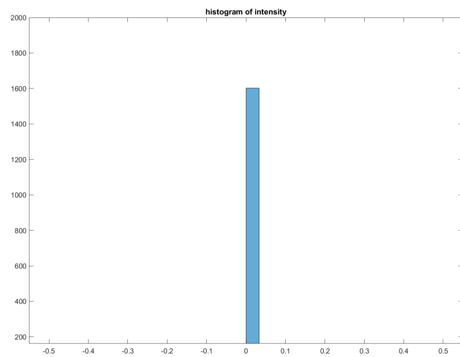
(b) The image has been converted into grey scale with the reference square determined



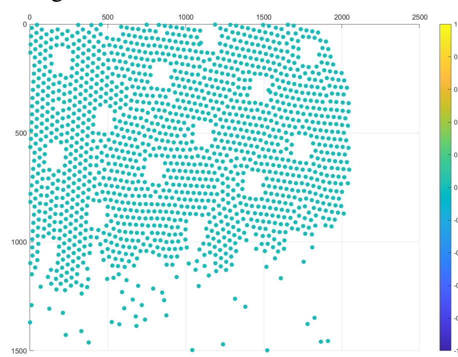
(c) Grey scale image has been pre processed to omit noise and to commit a better detection



(d) The location and size of each droplet demonstrated in the original fluorescent image



(e) histogram of integrated intensity of each droplet represented by the histogram



(f) intensity projected to its matching place

Figure 4: Workflow of one data set

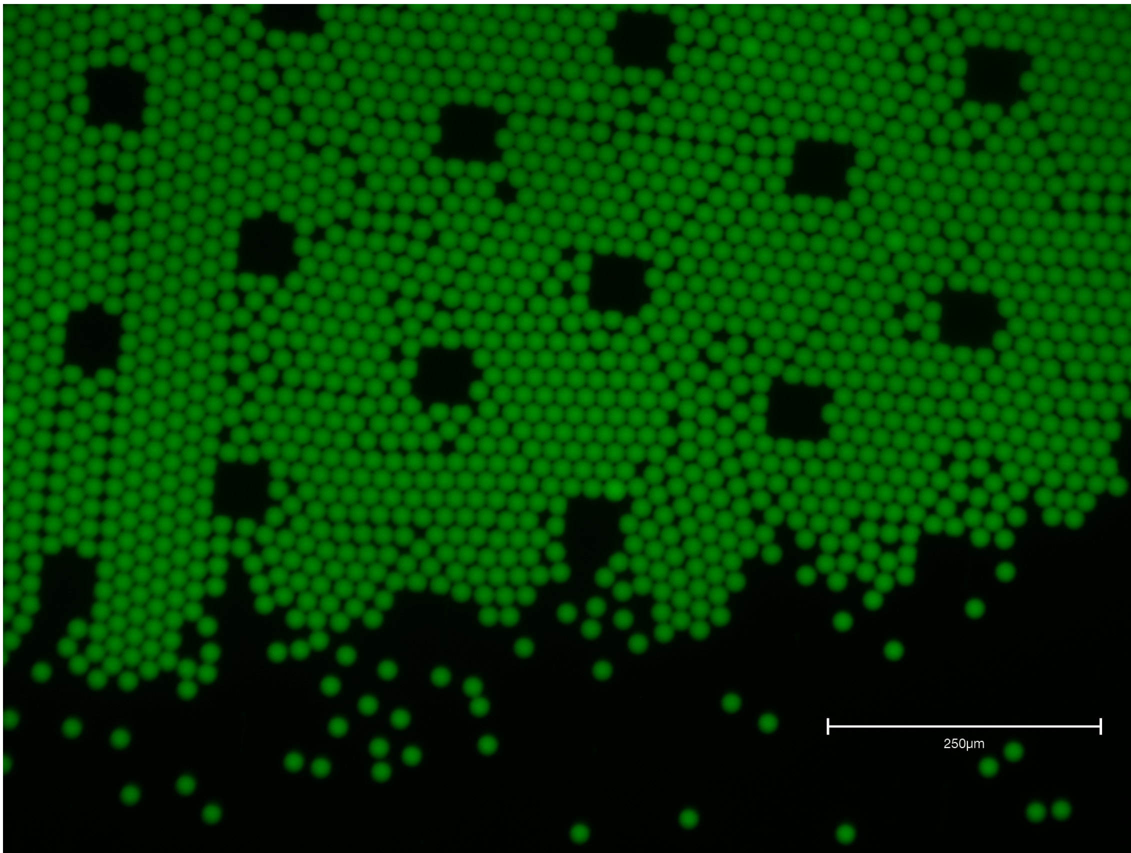


Figure 5: The original fluorescent image of image NO.4(no access to the bright field image)

Figure 5 shows the original fluorescent image, where the green brighten droplets can be found in the top half part, and the background is black. Since the original RGB image contains 3 different channels, The information of each pixel is determined by the combination of the red, green, and blue intensities stored in each colour plane at the pixel's location. We can convert such a colourful image into grey scale image that uses the grey level to represent the intensity. In this way less storage is used to store the same information.
In order to convert the image into grey scale, we used ”im2gray” function in the MATLAB. Such a function allows us to convert truecolour image RGB to the grey scale image by eliminating the hue and saturation information while retaining the luminance. The result image can be found in figure 6, where it can be observed that droplets has a light grey colour while the background keeps the black colour with values 0, since MATLAB has a a range from 0 to 255, from black to white.

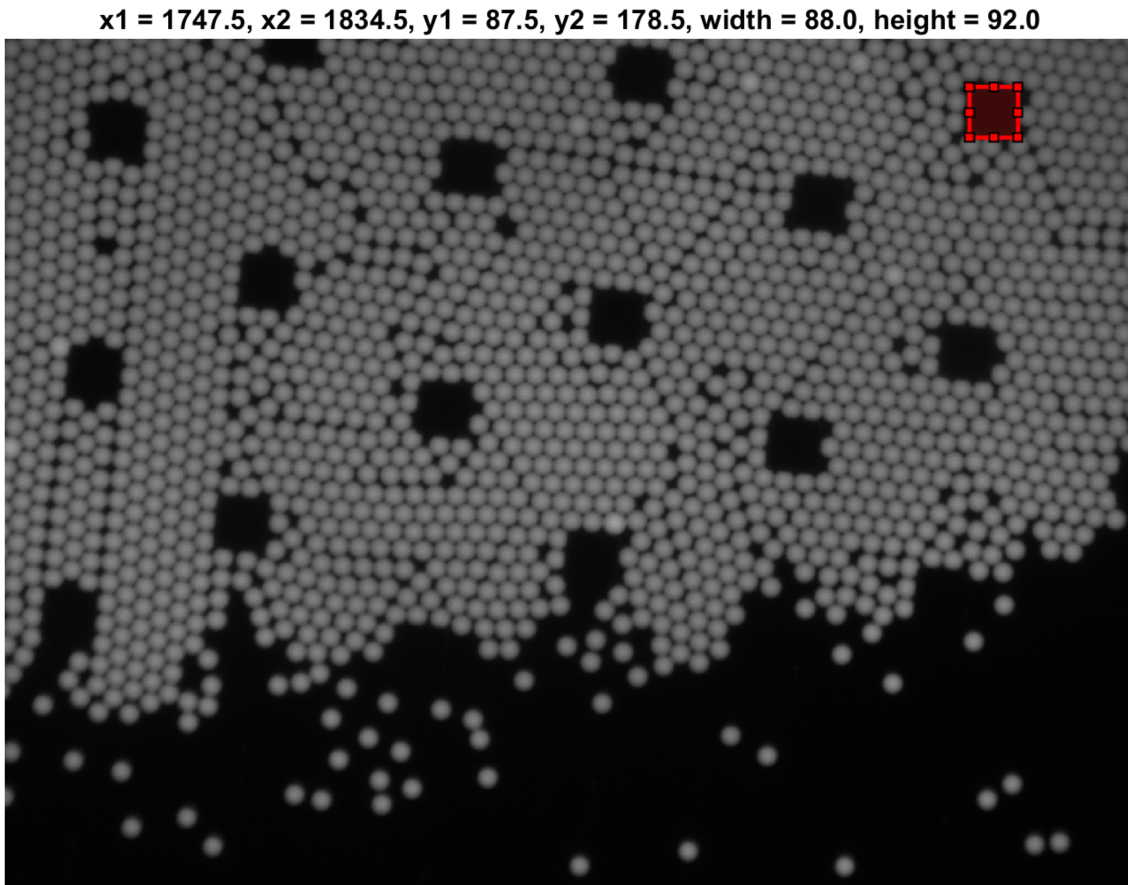


Figure 6: The data image has been converted into grey scale with the reference square determined

It should brought up reader's attention that there is one red square has been selected, which is called the reference square. It has been used to determine the size of droplets, which is used in the equation (3). Since the size of these squares are $50\mu m$, the ratio between the selected sides and the radius of droplets are can be used to calculate the actual volume as following,

$$\frac{side}{50\mu m} = \frac{r_{image}}{r_{actual}} \tag{6}$$

where the side is determined by the "drawrectangle"function as a region of interest (ROI). It allows user to manually select the region. The difference between horizontal and vertical coordinates are assigned to width and height, respectively. As human errors are inevitable, the average value of height and width is the side value. Last but not the least, the double click system is introduced to let users double confirm the choice, which can minimize the misclicks as many as possible.

While the r_{image} is the radius of droplets in the image. However, as the number of

droplets determined from grey scaled images are not as many as we expected, some extra pre-processing steps are essential to performed.

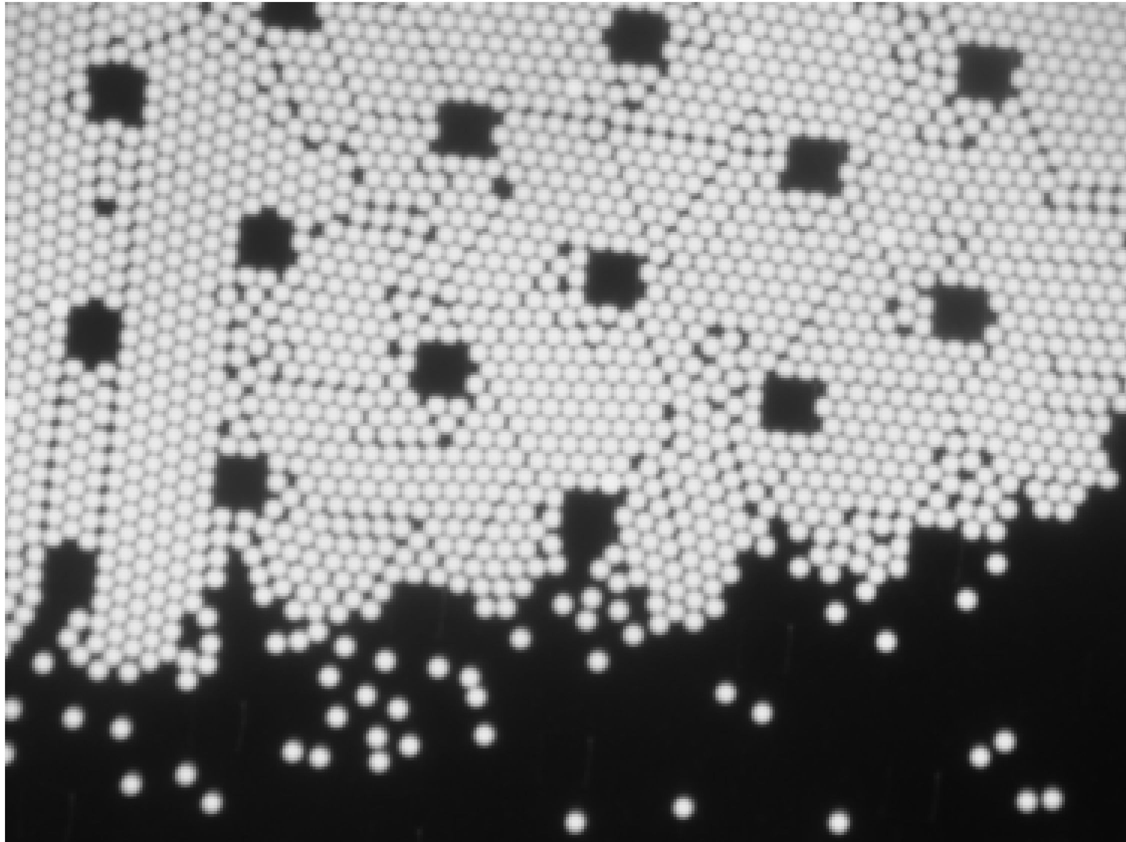


Figure 7: Pre-processing results of the images in grey scale that are used to determine the position and size of droplets

The result of an example of pre-processing is shown in the figure 7. The Gaussian filter has been used for image enhancement and smoothing. Since it helps with decreasing the noise in the images as long as blurring the images, which has less structure with regard to all the unnecessary details from the data. Because the noise mostly being a high frequency component, the Gaussian filter removes components above the cut off frequency. In this respect, the noise and any other high frequency pixels. The image enhancement, more specifically, brightening the image makes it easier to identify the circles' boundaries. Like an amplifier, it has a certain gain value that applies to all the pixels so that the absolute differences between boundaries and the rest increase. We have more edge details as a result. Bearing in mind that the main propose of this project is to detect as much as accurately detect the droplets using edge detection methods, which

can largely benefit from using Gaussian filter.
The Gaussian filter is a non-uniform low pass filter that is commonly applied to images prior to resampling. It ensures that high-frequency information does not suffer from aliasing.[15] because its Fourier transform is a Gaussian distribution centred around zero frequency (mirrored on both positive and negative sides). With adjusting width of the filter, its effectiveness can be easily controlled. Such a filter may not preserve image brightness[16], by reusing the value of the original image, it does not have an influence on the results

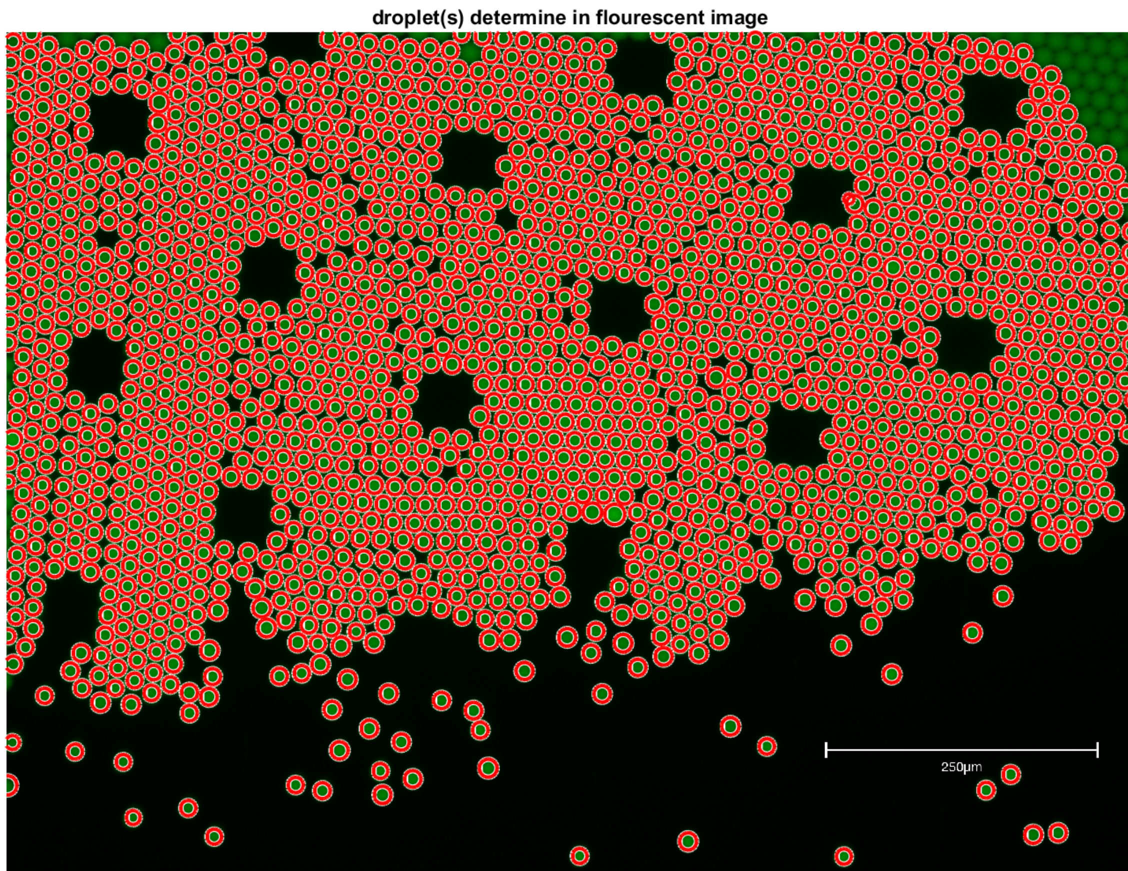


Figure 8: Droplets position with their corresponding radius showed in the fluorescent images of different magnification has been bounded by the red circle, it should be noticed that on the top right corner, a few droplets that the program was not able to detect

The fianl results of determined droplets positions and their radius can be found in the figure 8. A canny detector and Circular Hough Transform (CHT) based algorithm has been developed to localize the positions. The canny detector are used to detect the edge, it extracts useful structural information from different vision objects meanwhile

dramatically reduces computational complexity. It also produces a smoother boundaries by implementing the Non-maxima suppression and thresholding. Once all the edges has been detected, it is important to distinguish which are the boundaries of circle and what aare their radius, from then on, we used the CHT to achieve so. Because the Hough transformer enables us to identify and detect lines, circles, and other geometric shapes in this project the circles.

The kernel is applied to the image with determined centre coordinate mask, i.e., the circular one which has the detected target, to isolate the droplet. It defines in the function that loops through the boundary of the image, for every value in the range (the area) it scans each pixel to apply the filter till the end and stores the data to "blurred" variable. "len()" returns the number of items in the given object. And the "clip()" function is used for the given interval, values outside the interval are clipped to the interval edges, this can pervert the sudden changes in the image.

Gradient magnitude and direction of each pixel are stored in variable "gradient" and "direction", respectively. Zero paddings help to boost up the computation speed. In the famous cameraman picture (Fig.9a), the gradient vector is displayed, where you can see at the edge of the person, the vectors have higher magnitude and the gradient is perpendicular to the edges.
The vertical changes can be computed by taking the difference between the south and north pixels:

$$G_y = I(x, y + 1) - I(x, y - 1) \tag{7}$$

While the horizontal changes can be found use the method by taking the difference between east and west pixels

$$G_x = I(x + 1, y) - I(x - 1, y) \tag{8}$$

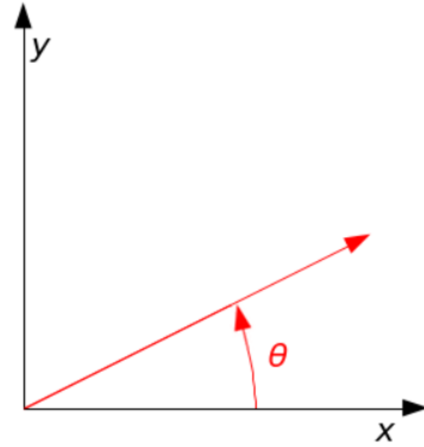
in which G_y and G_x represent the change in image intensity for the central pixel in both the x and y direction.
The gradient magnitude measures how strong the change in image intensity is. It is a real-valued number that quantifies the "strength" of the change in intensity. While the gradient orientation, as the name suggests, it gives an angle to quantify the direction of this change. It shows where this change in intensity is pointing. From Fig.9b, by applying Pythagorean theorem, the magnitude you can found the magnitude as,

$$G_{mag} = \sqrt{G_x^2 + G_y^2} \tag{9}$$

The radio of G_y to G_x can give us the orientation as what displayed in eq.(10),

$$G_{ort} = atan2(\frac{G_y}{G_x}) \tag{10}$$

(a) Cameraman with gradient vectors(open source picture)



(b) Coordinate system

Figure 9: Gradient explanation, the gradient of edge can be found in graph 9a, and the 9b shows the coordinate system that used in script to calculate the magnitude and orientation

the $atan2$ function compute the orientation in radians between π and $-\pi$. The next step is to apply the CHT by the following steps. From the simple linear algebra, a point in image space maps to a line in parameter space, and vice versa. Intuitively, it is possible to create an algorithm for the image detection. To create a parameter space that uses to define the accumulator array. It quantized with appropriate resolutions. All the quantized cells have zero paddings. For a given point, it is possible to plug in the equation of a circle with radius r and centre (a, b), the parametric equation is described below,

$$(x_i - a)^2 + (y_i - b)^2 = r^2 \tag{11}$$

$$=> \begin{cases} x = a + rcos(t) \\ y = b + rsin(t) \end{cases} with\ t \in [0,\ 2\pi) \tag{12}$$

For every point (x_i, y_i) that fall on the circle from the left side of Fig.10 is going to map a circle in parameter space on the right side of Fig.10, which means in the accumulator array, essentially, all the points are voted along the circles for any given point in image space, which is the voting scheme. And the same applies for the second point for the circle, it assigns a circle for each one of these points. As a result, all the circles intersect at one point, which corresponds to the centre of the circle in the image space. Traditionally, the radii R is unknown, it also requires to be assessed from accumulator array as (a, b, r). The dimensionality of Hough space has extended to three.

A point that forms the original circle is equivalent of a cone in the Hough space. For the complete set of points in image space, the maxima in the accumulator array can be found in order to determine the position and the radius of the original shape.

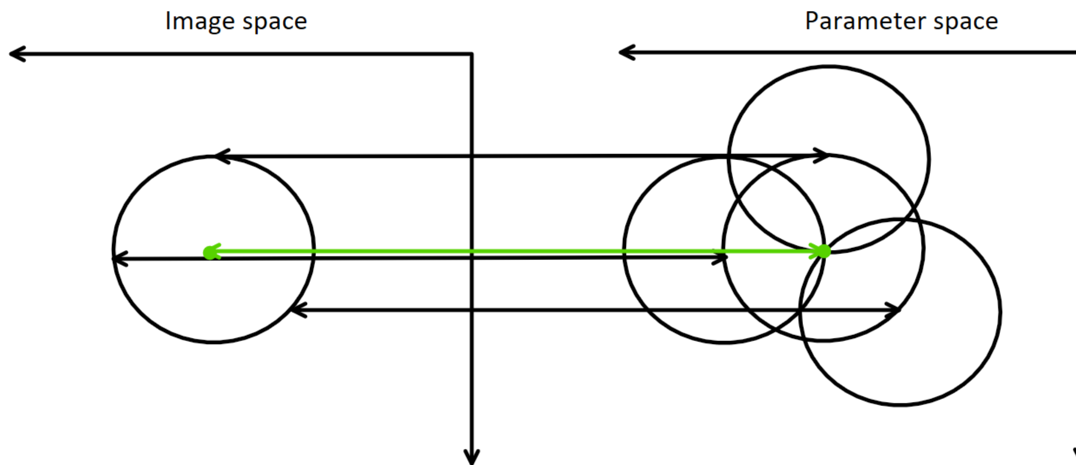


Figure 10: Circle image and its mapping in the parameter space, where the locus of (a,b) points in the parameter space fall on a circle of radius R centred at (x,y). The true centre point will be common to all parameter circles, and can be found with a Hough accumulation array, in the other words, each transformed point in the parameter space is considered as a candidate for being a line and accumulated in the corresponding cell of an accumulator

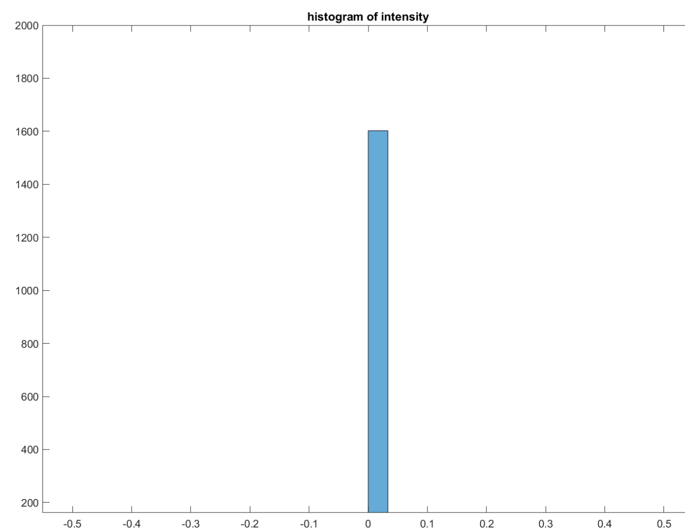


Figure 11: The histogram of integrated pixel intensity of each droplet with the concentration of 0.03 $copies/\mu l$

It should be noticed that the intensity we calculated here is the summation intensity over a single droplet instead of the average. Since the average intensity separates out the intensity based on their size. It can not reflect on the concentration, since the value has been separated out on every pixel within the object (droplet). A higher average intensity does not always indicate higher concentration, it can also be caused by the fact that there are multiple fluorescence are presented. Hence, the integrated intensity was used in the product.

The intensity summation for each droplet has been used to show how bright they are. Because the average of each droplet is influenced by the droplet size, in other word, the concentration, it is not a suitable way to indicate the brightness of droplets. On the other hand, the summation can reflect on the intensity as well as the size, which is used in this project.
"meshgrid" function defines two dimensional coordinates based on the pre-determined vector of each droplet circle into array $xgrid$ and $ygrid$. It evaluates functions of two variables and three-dimensional mesh. In this case, the row of $xgrid$ and the output columns $ygrid$ copy from one to the original images size. Later, a mask for each droplet has been made, it includes the entire area of the circle, and the values of every pixel in this regime has been recorded in "mask" variable. The "sum" function is used to add up all the values and to store in the "int_sum". We also projected the intensity to its corresponding location as what is shown in the figure 12.

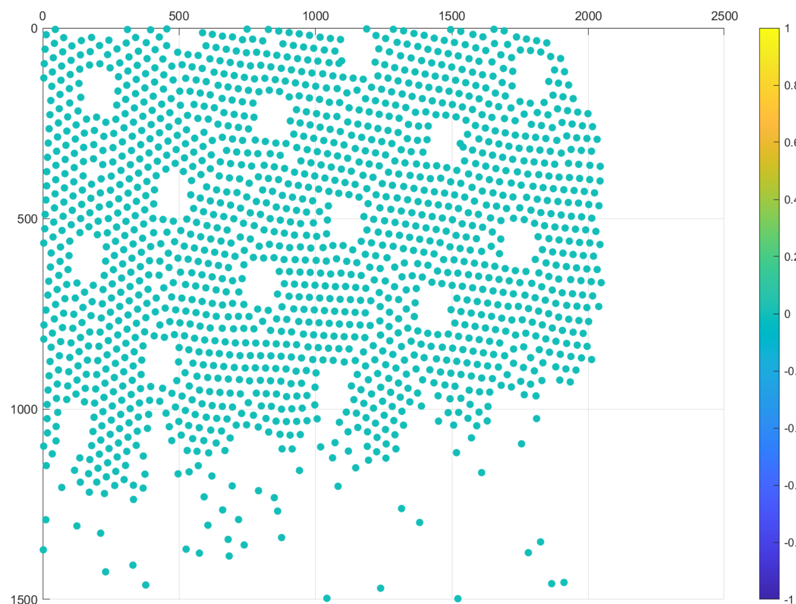


Figure 12: Intensity projection of droplets in their position with colour bar indicates their intensity on the right hand side

The projecting the intensity to its position gives us a clear look on the droplets are distributed. The lower value in colour bar means the actual intensity is higher, moreover, higher and warmer colour means the high intensity. Notably, the scatter does not correspond to the size of droplet but only its location. After calculating the intensity, users can use slider to optimize the threshold value. To use eq.(5), we also need to determine what is the threshold in order to distinguish positive and negative droplets, hence, we introduced a slider in the GUI to allow users to manually find the best optimized the threshold, which can be found in the figure 16.

There are three choices during the design process,

- applying a filter to smooth the image and find the local minima that is the threshold;
- entering a fix number by users;
- using a slider to test for different various of threshold data value.

Once the users find the most optimized intensity threshold, we can compute the total number of negative droplets from the percentage of slider by the equation 13,

$$int_{th} = int_{min} + x \bullet (int_{max} - int_{min}) \tag{13}$$

where int_{th} is the threshold value of intensity, the int_{min} and int_{max} are the lowest and highest bar of pre-recorded intensity. With such a threshold value, it allows us to compare the actual intensity of every droplet, if their values are lower than the threshold, the counter will count numbers up until the loop finishes, which counts as a negative number droplet as it shown in the below algorithm template in $Listing1$,

Listing 1: count number template

```
count = 0
    if droplet's intensity < threshold
        count = count ++
    end
```

Last but not the least, we convert the script into a simple app interactively, using the App Designer from MATLAB Add-ons. We firstly designed the graphical user interfaces and assigned different functions to based on the prior mentioned designing steps. From the idea of model-view-controller, the design pattern that is widely used to help build the frameworks for applications. There are three section in the theory that we need to take into account, the Model, view and controller as it named. In our cases, the model is where the data is manipulated and/or saved, which is the data and the logic of our product that can have a to-one or to-many relationships to other models. In short, it is our core codes. Once there are changes in the data, it notifies the view and controller to do different decisions. While the view is the part of the app that users has direct

interactions with. In this cases, the buttons, slider, different plots and the numerical results. It displays the processed or raw data that is attached to the model. In the same time, it sends appropriate message to the model that allows model to do corresponding updates. While the controller receives commands from users and translates it to pass them. It is the bridge between the view and model. In our product, the MATLAB provides a platform to do such a cooperation. It ensures the smooth and easy to follow users' experience.

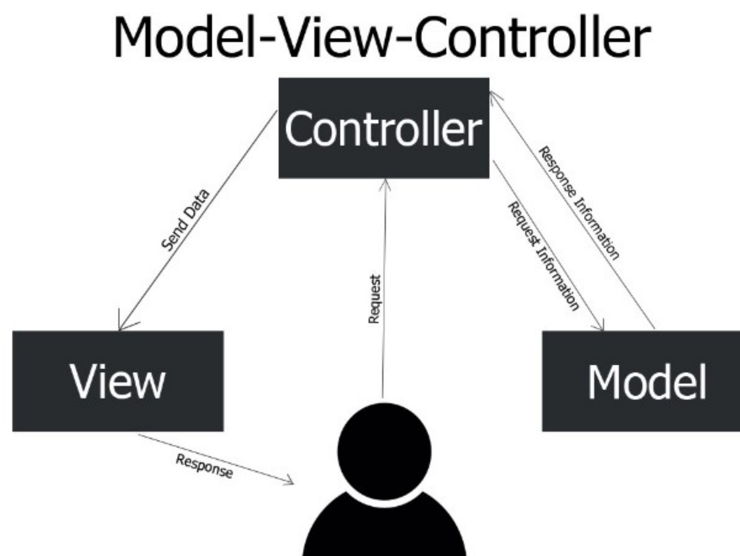
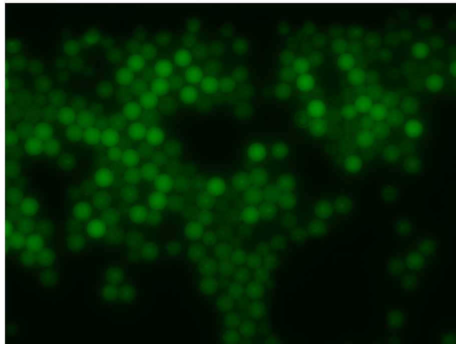
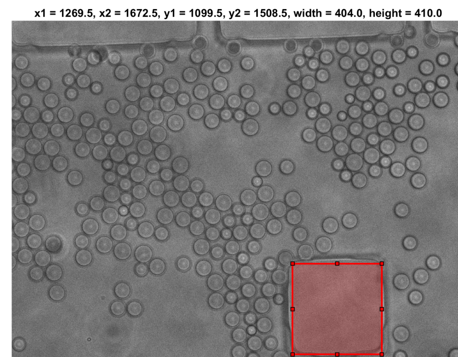


Figure 13: The relationships between the model, view and controller can be found in the graph where it indicates that the view is only section that interact with the users, be cautious that the model represents the data and it is independent from controller and view

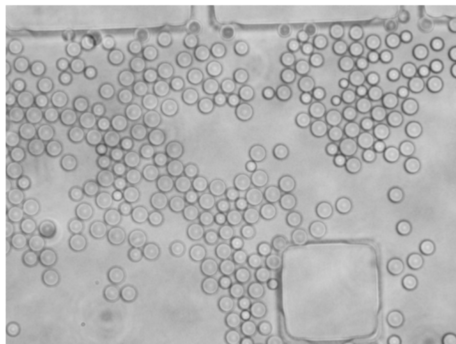
IV. Results



(a) The original fluorescent image of a different data

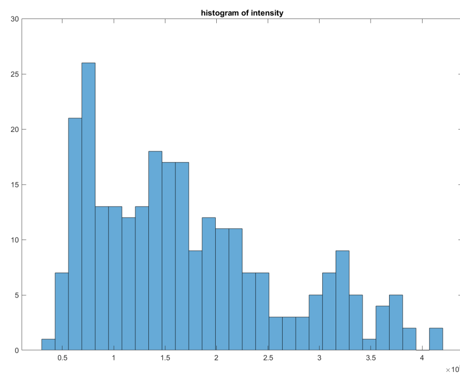


(b) The image has been converted into grey scale with the reference square determined

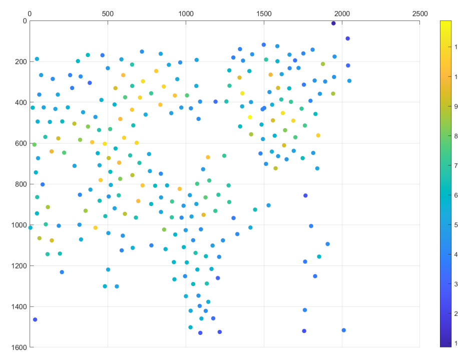


(c) Grey scale image has been pre processed to omit noise and to commit a better detection

(d) The location and size of each droplet demonstrated in the original fluorescent image



(e) histogram of integrated intensity of each droplet represented by the histogram



(f) intensity projected to its matching place

Figure 14: Workflow of different set of image

The grey scale images as well as the reference squares are able to be converted and located well in the given data.
Meanwhile, by observing the results after the Gaussian filter, it can be noticed that the signal to noise ratio (SNR) has increased significantly. The SNR is calculated as following,

$$SNR = \frac{\mu_{signal}}{\sigma_{noise}} \tag{14}$$

in which the μ and σ are the average signal value and the standard deviation to the background, respectively.

Table I: The determined droplets number with left out and inaccurate detection outcomes

Image NO.	Detected NO.	False negative NO.	False negative rate	False positive NO.	Positive predictive value
1	264	10	3.6%	2	99.2%
2	100	0	0%	2	98%
3	103	3	2.8%	19	84.4%
4	1603	36	2.2%	0	100%
5	5665	0	0%	Approx 72	98.7%
6	472	1	0.2%	Approx 80	85.5%

The table I displays the results of the droplets determination. Most droplets are detected, and the product performed well in all 6 samples since there is a limited number of droplets left out. The respondents had practised for an average of 1.2% using formula (15). The false negative rate is calculated as

$$\frac{FN}{FN + TP} \tag{15}$$

where FN means there is no droplets presents but detected by the algorithm. While the TP is the droplets that have been detected faithfully. Conversely, in case of inaccurate detection, the performance in the sample NO. 1, 2, 4 and 5 are quite pleasant. Since their positive predictive values are over 98% derived from formula (16). Comparing to our objective of 90%, the sample NO. 3 and 6 are still within an acceptable range, no more than 6% difference. Positive predictive value is the likelihood that if the droplet has gotten a positive test, where the fluorescence signal is detected. It is calculated as

$$\frac{TP}{TP + FP} \tag{16}$$

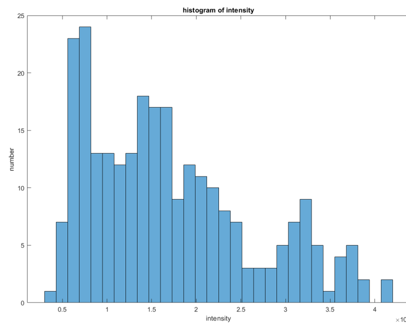
However, it detects the corner of reference square as droplets circles. This happens in sample NO. 6 as well. The reason behind it is most likely due to the Gaussian filter, it blurs the sharp corner and lets the CHT algorithm define the corner as a circle.
Later on, we tested the product on the fake data that generated by ourselves to compare it with real data, and the results are shown in the table below.

Table II: Comparison between the fake data and real test samples

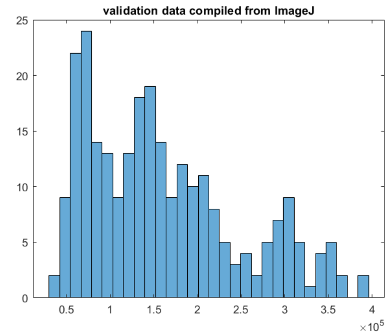
	Average false negative rate	Average positive predictive value
Fake data	0%	100%
True sample	1.47%(±1.45%)	94.3% (±6.7%)

The table II shows the comparisons between the fake data and the real world fluorescent images. The false negative rate and positive predictive value of fake data is incredibly well, since every desired feature has been achieved in those samples, whereas the performance in the true samples has reduced, since there is more interference. But the positive prediction is over the criteria of 90%, and the false negative rate is less than 2%. Because it evaluates how many true droplets have been incorrectly indicated, we would desire it as low as possible. It can therefore be assumed that the product works as expected.

In addition to all the detection and determination results, we can plot the integrated intensity histogram accurate enough with the comparison to the ImageJ, which is one of the most used commercial tool for cell/droplet counting as it shown in the figure 15.



(a) integrated intensity of each droplets



(b) Validation intensity data from ImageJ shows in histogram, the sum of frequency in each is 261

Figure 15: The intensity comparison between our results and the outcomes from ImageJ as validation data, in which the two plots has similar threshold value as well as the similar Poison distribution

The threshold is demonstrated in the following picture 16. Since applying a smooth filter will allow system to perform the operation automatically, however, the drawback is that there is a chance the local minimal value is not the true intensity threshold that will cause the error in the system.

While entering the fix number gives users the full access to find the threshold value

without interpret from machine and it is also more straight forward, but the trade off is that the exact number of local minimal is hard to read from the figure.
The slider gives the maximal freedom to users as well as ensures the accuracy in the determination, as the intensity of each droplet has been calculated in the previous section, and we designed the slider scaled from zero to one to allow users to testify different thresholds. As displayed in the figure 14f, the projection works as what it expected to be.

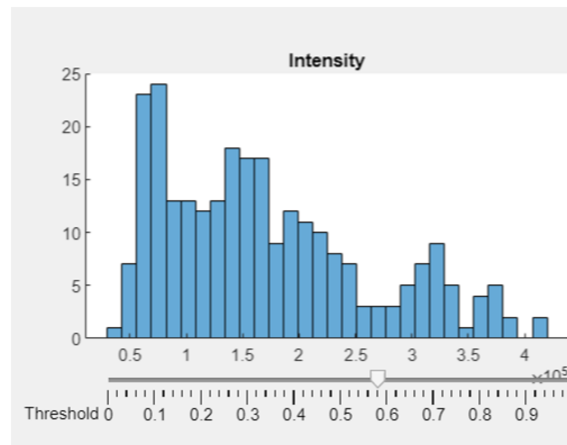


Figure 16: Determination the number of positive and negative droplets from the histogram using a slider, where in the figure the threshold is $260k\ copies/\mu l$

Meanwhile, the graphical user interface (GUI) was finally made as it in the figure 17. The original image or the original size can be used in terms of calculating the volume to determine the concentration. The bright field and fluorescent image can be selected by the user for the processing purposes. By clicking different buttons, the users can apply different features, and the slider is used to determine the positive/negative ratio based on the intensity of the image. The total number, positive ratio and concentration can be found in the text box once the correspond button has been pushed. The position of droplets, intensity and droplet intensity projects on its position are tied in their axes illustrations with image titles. The reference square position allows users to select the square manually, it also compute the resize ratio for the concentration calculation.

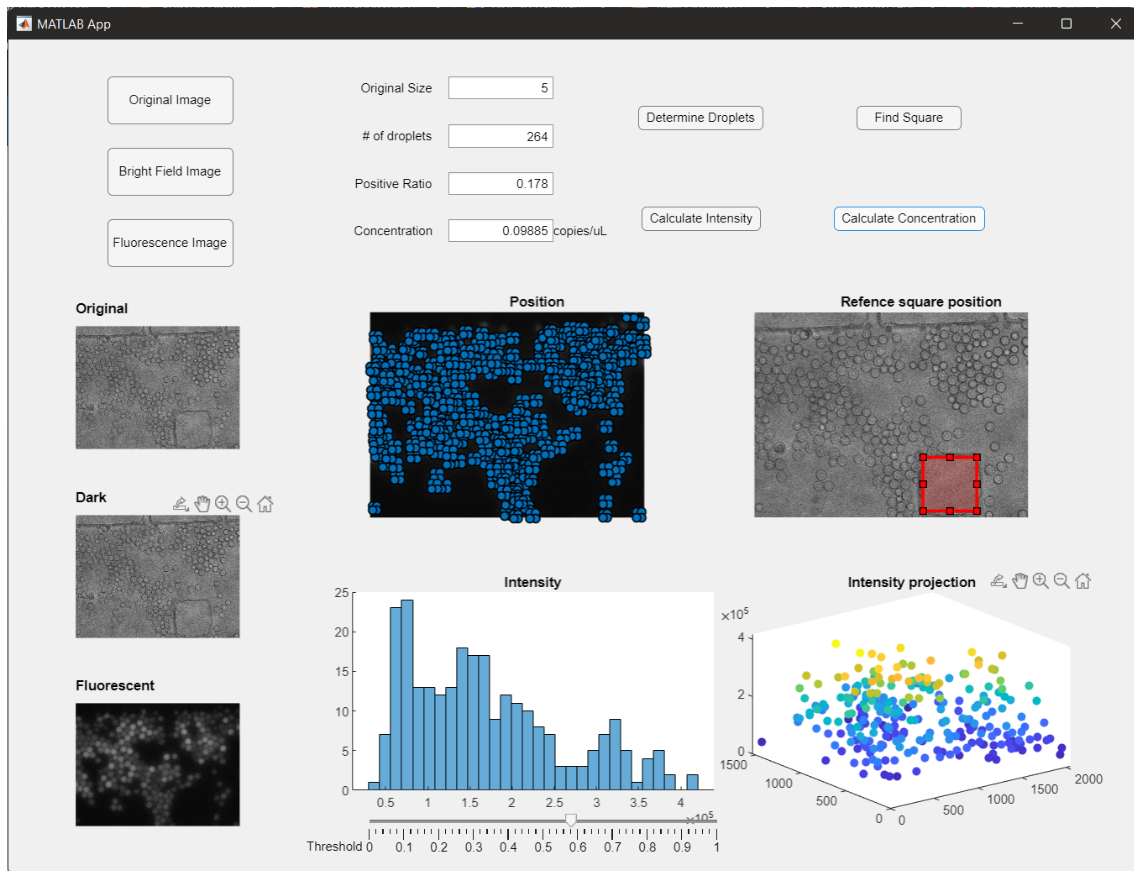


Figure 17: Product GUI

V. Conclusion and Outlook

By observing the fluorescent image and the bright field image carefully, there are some droplets ignored by the ImageJ comparing to the product due to their low intensity or high connectivity. Meanwhile, it can be noticed that there are some droplets are left out in the final product which needs to be fix but with the average accuracy at 94.3% which is over the desired sensitivity and the false negative less than 5%, the product has achieved both basis that were set at the first place. Overall, it can be said that the final product has achieved a better performance than the ImageJ with respect to the number of determined droplets.
To conclude, the main goal of this project is to detect as many droplets as possible and to perform analysis to estimate the probability that the patient has early stage cancer. The product is able to detect the droplets' position as well as their radius. Meantime, as the fake data confirmed, the integrated intensity calculation should be accurate. However, there is no ground truth data presented to verify and evaluate the actual performance. Taken together, these results suggest the project is a success, but there can be more future development.

With a small sample size (9 samples), caution must be applied, as the product may be sufficient in a certain range. Because there is no general solution for every possible situation in terms of image segmentation. The ability of edge detection is limited, it can be considered only as a feature in machine learning and pattern recognition. Because of the presence of noise, it is difficult to detect edge no matter the colour, length or facets.

Comparing the intensity demonstrates in the table or the histogram, the projection is more direct and straight forward to show the position and intensity value.

In the view of human technology relations, as the target groups are most biologist and chemist, who have limited ability of coding. With a GUI, it can be easier for them to accept the product. Meantime, entering the position requires stakeholders to determine the exact position when they took the photo, that is not ideal, since we want it to be more convenient by using machine to automatically find the edges. However, thing does not go as what expected as the actual edges are not found, but multiple interferes were detected. The reason is that the square sides are not straight lines but have certain curvatures. The compromise was finally made by selected the region as what we seem in the product GUI (figure 17).

There are some human mistakes should be take into account. For instance, the square

selection, we let users double click to prevent the human error, and the margin for the side error has been set as 50 pixels. The later term means if the difference of selected rectangle's length and width is within 50 pixels, the system will define it as a square and we will take the average of the two as the side ($side = \frac{width + length}{2}$). As for the double click system, there is a chance that the users become impatient due to various reasons, and they would accidentally make an unwanted choice. However, in the test stage, there are square side that is less than the error margin, these data must be interpreted with caution because the mistake-proofing system can not work against the cases.

To improve the left-out droplets improved by tuning the disk size to obtain a better result or give users access to manually select those low-intensity or high-connectivity droplets. In the case of highly overlapped droplets, it can be fixed by setting the dynamical or global threshold value of the droplet's size. If the size goes beyond the value, there are at least two droplets, the counter should count the region as two. Simultaneously, in the case of low-intensity droplets or left-out ones, it can be thus suggested that allowing users to manually select can also help improve the accuracy as well. Considering the number of those droplets was not high, it should not be a big work from the users' perspective.

The intensity projection can be further improved by adding a data image in the xy-plane, and using a bar instead of a scatter to show the position and value as the altitude map.

Humans are imperfect and our fallibility is universal, it is human nature to misjudge and to make errors. Continuing improvement through human factors analysis is an essential part of the system debugging and testifying reliability. There are some ways to minimize the human impact on our product.

1) adjusting error margin.
2) allowing iteration of choosing.
3) avoiding human involvement by using machine learning or AI for the task.

An adjustable error margin can be tuned based on the size of the pixel area which can reduce the error in the small region. The average of repetitive tasks can lower the error better. Last, completing tasks with AI can eliminate the human factor entirely, since no human is involved in the procedure.

The K-means clustering can be a good solution if there is not enough labelled data present, because it is an unsupervised algorithm. It aims to partition the observed data into a number of clusters (K). The "means" refers to the average of data, which can be

used to find the centroid of clusters, in this case, of each droplet. It allows the model to interact directly with the input vector with no reference to known, labelled data or outcomes. But the risk is that it is hard to find the balance between robust and aggressive algorithms, again, due to its high connectivity. The radius is also required to determine alongside the centroid, but the k-means algorithm can not achieve this directly. Another solution for small data samples is to use few-shot learning. More specifically, meta-learning. It allows us to transfer the source and apply it to the target with no more than 20 labeled samples. It ensures the machine learning benefits from training, and if the system fails to accomplish this efficiently, one can expect the learning mechanism to adapt in case the same class appears again.

Overall, this project was undertaken to design of an auto counting droplets machine for all the possible images and evaluate its performance. The idea was ambitious and complicated. As the project goes on, there were different difficulties that may or may not be conquered. The project goals that have been set at the beginning stage have been achieved at a certain level. We have investigated the current image processing method and were able to design a program that detects and counts the droplets. It is a pity that the machine learning algorithm does not perform as we expected it to. Lastly, we have developed a quite satisfying graphical based user interface. Overall, the project can be considered as a success.
The project requires good self-controlling and time management. During this project, these aspects could be improved, as we spent multiple days on a failed method without any development, but the bright side of this is that it makes the solid ground for the actual product in terms of the pre-processing stage.
To conclude, the project had many learning experiences. At the end of this module, we could say that we learned a mix of new tools that could be used in the future. Signal analysis, machine learning, biology, chemistry, philosophy of technology, and system engineering are some of the tools for a great engineer.

REFERENCES

[1] P. D. Hsu, D. A. Scott, J. A. Weinstein, F. A. Ran, S. Konermann, V. Agarwala, Y. Li, E. J. Fine, X. Wu, O. Shalem, T. J. Cradick, L. A. Marraffini, G. Bao, and F. Zhang, "DNA targeting specificity of RNA-guided Cas9 nucleases," *Nature Biotechnology 2013 31:9*, vol. 31, no. 9, pp. 827–832, 7 2013. [Online]. Available: https://www.nature.com/articles/nbt.2647

[2] R. F. Weaver, *Molecular biology*. McGraw-Hill, 2012.

[3] A. M. Streets and Y. Huang, "Chip in a lab: Microfluidics for next generation life science research," *Biomicrofluidics*, vol. 7, no. 1, 2 2013. [Online]. Available: /pmc/articles/PMC3574129//pmc/articles/PMC3574129/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC3574129/

[4] G. M. Whitesides, "The origins and the future of microfluidics," *Nature 2006 442:7101*, vol. 442, no. 7101, pp. 368–373, 7 2006. [Online]. Available: https://www.nature.com/articles/nature05058

[5] M. Vaithiyanathan, N. Safa, and A. T. Melvin, "FluorocellTrack: An algorithm for automated analysis of high-throughput droplet microfluidic data," *PLoS ONE*, vol. 14, no. 5, 5 2019.

[6] "What is CRISPR? — New Scientist." [Online]. Available: https://www.newscientist.com/definition/what-is-crispr/

[7] R. Barrangou and L. A. Marraffini, "CRISPR-Cas systems: prokaryotes upgrade to adaptive immunity," *Molecular cell*, vol. 54, no. 2, p. 234, 4 2014. [Online]. Available: /pmc/articles/PMC4025954//pmc/articles/PMC4025954/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC4025954/

[8] M. Jinek, K. Chylinski, I. Fonfara, M. Hauer, J. A. Doudna, and E. Charpentier, "A programmable dual-RNA-guided DNA endonuclease in adaptive bacterial immunity," *Science (New York, N.Y.)*, vol. 337, no. 6096, pp. 816–821, 8 2012. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/22745249/

[9] J. E. van Dongen, J. T. Berendsen, J. C. Eijkel, and L. I. Segerink, "A CRISPR/Cas12a-assisted in vitro diagnostic tool for identification and quantification of single CpG methylation sites," *Biosensors and Bioelectronics*, vol. 194, 12 2021.

[10] H. L. Osterman, X. Peng, H. Chen, D. R. Draney, A. Schutz-Geschwender, and D. M. Olive, "The Next Step in Near Infrared Fluorescence: IRDye ® QC-1 Dark Quencher A review of the following studies."

[11] V. Ayyadevara, Y. Reddy, O'Reilly for Higher Education (Firm), and a. O. M. C. Safari, "Modern Computer Vision with PyTorch."

[12] S. Bartkova, M. Vendelin, I. Sanka, P. Pata, and O. Scheler, "Droplet image analysis with user-friendly freeware CellProfiler," *Analytical Methods*, vol. 12, no. 17, pp. 2287–2294, 5 2020.

[13] "Image Processing — CellProfiler 3.0.0 documentation." [Online]. Available: http://cellprofiler-manual.s3.amazonaws.com/CellProfiler-3.0.0/modules/imageprocessing.html?highlight=threshold

[14] "What are TIFF files and how do you open them? — Adobe." [Online]. Available: https://www.adobe.com/creativecloud/file-types/image/raster/tiff-file.html

[15] T. Sakai, M. Narita, T. Komazaki, H. Nishiguchi, and A. Imiya, "Image hierarchy in gaussian scale space," *Advances in Imaging and Electron Physics*, vol. 165, no. C, pp. 175–263, 2011.

[16] E. DAVIES, "Basic Image Filtering Operations," *Machine Vision*, pp. 47–101, 2005.

APPENDIX

A. Code

Listing 2: Product

```
classdef droplet_counting < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                       matlab.ui.Figure
        copiesuLLabel                  matlab.ui.control.Label
        OriginalImageButton_2          matlab.ui.control.Button
        ConcentrationEditField         matlab.ui.control.NumericEditField
        ConcentrationEditFieldLabel    matlab.ui.control.Label
        ThresholdSlider                matlab.ui.control.Slider
        ThresholdLabel                 matlab.ui.control.Label
        OriginalSizeEditField          matlab.ui.control.NumericEditField
        OriginalSizeEditFieldLabel     matlab.ui.control.Label
        DetermineDropletsButton        matlab.ui.control.Button
        FindSquareButton               matlab.ui.control.Button
        CalculateConcentrationButton   matlab.ui.control.Button
        CalculateIntensityButton       matlab.ui.control.Button
        PositiveRatioEditField         matlab.ui.control.NumericEditField
        PositiveRatioEditFieldLabel    matlab.ui.control.Label
        ofdropletsEditField            matlab.ui.control.NumericEditField
        ofdropletsEditFieldLabel       matlab.ui.control.Label
        FluorescenceImageButton        matlab.ui.control.Button
        BrightFieldImageButton         matlab.ui.control.Button
        UIAxes2_5                      matlab.ui.control.UIAxes
        UIAxes2_4                      matlab.ui.control.UIAxes
        UIAxes2_2                      matlab.ui.control.UIAxes
        UIAxes2                        matlab.ui.control.UIAxes
        UIAxes_3                       matlab.ui.control.UIAxes
        UIAxes_2                       matlab.ui.control.UIAxes
        UIAxes                         matlab.ui.control.UIAxes
    end


    properties (Access = private)
        img_org;     % orginal image
        img_dark;    % dark feild image
        img_int;     % bright field iamge
        brightVal;   % parameter for brightness
        dropNum;     % determined droplets number
        center;      % center of droplets
        radii;       % droplets radius
        radii_org;   % original radius
```

```
43          center_org;
44          int_sum;     % intensity of each droplets
45          radius_org; % enter orginal size
46          side;        % refernce side
47          data_org;    % store orginal data position and radius
48          data_dec;    %detected ....
49      end
50
51      methods (Access = private)
52
53          function [ncchans nachans] = chancount(app, inpict)
54          %image channel counting
55          %   IN:
56          %
57          %       any class image array
58          %
59          %   out:
60          %
61          %       # of colour channels
62          %       # of alpha channels
63              numchans = size(inpict,3);
64              if ~ismember(numchans,[1 2 3 4 6])
65                      error('CHANCOUNT: expected image to be I/IA/RGB/RGBA/RGBAAA
                            . What is this %d-channel image supposed to be?',
                            numchans)
66              end
67              hasalpha = 1-mod(numchans,2);
68              ncchans = min(numchans-hasalpha,3);
69              nachans = numchans-ncchans;
70              if nargout < 2
71                ncchans = [ncchans nachans];
72              end
73          end
74
75
76      end
77
78
79      % Callbacks that handle component events
80      methods (Access = private)
81
82          % Button pushed function: OriginalImageButton_2
83          function OrignalImageButtonPushed(app, event)
84              [FileName, FilePath]= uigetfile({'*.tif';'*.jpg';'*.bnp';'*.gif'},
                    'Select a file'); %Don't override path.m
85              if isnumeric(FileName)
86                  return;
```

```
87              end %User did not select a file
88              app.img_org = imread(fullfile(FilePath, FileName)); %Always use
                    full file path, not relative path
89              imshow(app.img_org, 'Parent' , app.UIAxes);
90              lowerBound = 20;
91              upperBound = 100;
92              imgCirc1 = imgaussfilt(imlocalbrighten(app.img_org,0.519),3);   %
                    pre-processing image
93              [app.center_org, app.radii_org] = imfindcircles(imgCirc1,[
                    lowerBound, upperBound]);    %determine the circles
94              app.data_org = [app.center_org app.radii_org];
95          end
96
97          % Button pushed function: BrightFieldImageButton
98          function BrightFieldImageButtonPushed(app, event)
99              [FileName, FilePath]= uigetfile({'*.tif';'*.jpg';'*.bnp';'*.gif'},
                    'Select a file'); %Don't override path.m
100             if isnumeric(FileName)
101                 return;
102             end %User did not select a file
103             app.img_dark = imread(fullfile(FilePath, FileName)); %Always use
                    full file path, not relative path
104             imshow(app.img_dark, 'Parent' , app.UIAxes_2);
105         end
106
107         % Button pushed function: FluorescenceImageButton
108         function FluorescenceImageButtonPushed(app, event)
109             [FileName, FilePath]= uigetfile({'*.tif';'*.jpg';'*.bnp';'*.gif'},
                    'Select a file'); %Don't override path.m
110             if isnumeric(FileName)
111                 return;
112             end %User did not select a file
113             app.img_int = imread(fullfile(FilePath, FileName)); %Always use
                    full file path, not relative path
114             imshow(app.img_int, 'Parent' , app.UIAxes_3);
115         end
116
117         % Button pushed function: DetermineDropletsButton
118         function DetermineDropletsButtonPushed(app, event)
119             temp = app.img_dark;
120 %             brightVal = app.brightness;
121
122             % ceck and convert to gray scale image
123             [ncc nca] = chancount(app,temp);
124             if ncc<=2
125                 imgCirc = temp;
126             else
```

```
127                 imgCirc = rgb2gray(temp);
128             end
129
130             lowerBound = 20;
131             upperBound = 100;
132             imgCirc1 = imgaussfilt(imlocalbrighten(imgCirc,0.519),3);    %pre-
                    processing image
133             % imgCirc1 = imbinarize(imgCirc1);
134             % imgCirc1 = imcomplement(imgCirc1);
135             [app.center, app.radii] = imfindcircles(imgCirc1,[lowerBound,
                    upperBound]);     %determine the circles
136             app.data_dec = [app.center app.radii];
137             imshow(app.img_int, 'Parent', app.UIAxes2), %title('droplet(s)
                    determine in flourescent image')
138
139             %% visualize circle on the original image
140 %               viscircles(centers, radii, 'EnhanceVisibility', false);
141             app.ofdropletsEditField.Value = size(app.radii,1);
142             app.dropNum = size(app.radii, 1);
143
144             % mark circles
145             H = zeros(app.dropNum,1);
146             for n = 1:app.dropNum
147                 h = images.roi.Circle(app.UIAxes2,'Center',app.center(n,:),'
                        Radius',app.radii(n));
148             end
149
150         end
151
152         % Button pushed function: CalculateIntensityButton
153         function CalculateIntensityButtonPushed(app, event)
154             centers = app.center;
155             radiis = app.radii;
156             imgCirc = app.img_int;
157
158             [xgrid, ygrid] = meshgrid(1:size(imgCirc,2), 1:size(imgCirc,1));
                        % grid image
159             app.int_sum = zeros(app.dropNum,1);
160             % mask to located the circles from 1 to n
161             for n = 1:app.dropNum
162                 mask = ((xgrid-centers(n,1)).^2 + (ygrid-centers(n,2)).^2) <=
                        radiis(n).^2;
163                 app.int_sum(n) = sum(imgCirc(mask));
164             end
165             h = histogram(app.UIAxes2_2, app.int_sum, 30); %title('histogram of
                     intensity'), xlabel('intensity'), ylabel('number')
```

```
166             scatter3(app.UIAxes2_5,centers(:,1), centers(:,2), app.int_sum, [],
                     app.int_sum, 'filled');
167         end
168 
169         % Button pushed function: FindSquareButton
170         function FindSquareButtonPushed(app, event)
171             % % sqaure selecting
172             imgCirc = app.img_dark;
173             imshow(imgCirc, 'Parent', app.UIAxes2_4); %title('refence square
                    determination')
174 
175             % drawing
176             while true
177             ROI = drawrectangle(app.UIAxes2_4, 'Color', 'r');
178             x1 = ROI.Position(1);
179             x2 = x1 + ROI.Position(3) - 1;
180             y1 = ROI.Position(2);
181             y2 = y1 + ROI.Position(4) - 1;
182             width = ROI.Position(3);
183             height = ROI.Position(4);
184             temp = abs(width - height);
185             caption = sprintf('x1 = %.1f, x2 = %.1f, y1 = %.1f, y2 = %.1f,
                    width = %.1f, height = %.1f',...
186                     x1, x2, y1, y2, width, height);
187             title(caption, 'FontSize', 15);
188                 if temp <= 50
189                     break
190                 end
191             end
192 
193             app.side = (width + height) / 2;     %calculate side
194 
195             % confirmation
196             wait(ROI);
197             x1 = ROI.Position(1);
198             x2 = x1 + ROI.Position(3) - 1;
199             y1 = ROI.Position(2);
200             y2 = y1 + ROI.Position(4) - 1;
201             sprintf('x1 = %.1f, x2 = %.1f, y1 = %.1f, y2 = %.1f, side = %.1f'
                    ,...
202                 x1, x2, y1, y2, app.side);
203             close all
204         end
205 
206         % Button pushed function: CalculateConcentrationButton
207         function CalculateConcentrationButtonPushed(app, event)
208 %             cenc = size(app.dropNum, 1);
```

```
209             if logical(app.OriginalSizeEditField.Value) == 1    % from the text
                     box
210                 r = app.radius_org;
211                 ratio = app.radii ./ r * 50*10^(-6)/app.side;
212             else                                                % from orginal
                    picture
213                 r = app.radii_org;
214                 if app.center == app.center_org
215                     ratio = app.radii ./ r;
216                 end
217                 r = r * 50/app.side;                    % convert to true size
218             end
219             Vol =sum(4/3 * pi * r.^3)/app.dropNum;              %average volume
220             cenc = -log(1-app.PositiveRatioEditField.Value)/Vol;
221 %             histogram(app.UIAxes2_3, radio, 30);
222             app.ConcentrationEditField.Value = cenc;
223         end
224 
225         % Value changed function: ThresholdSlider
226         function ThresholdSliderValueChanged(app, event)
227             value = app.ThresholdSlider.Value;
228                         changingValue = value;
229 %             intSum = app.int_sum;
230             temp = changingValue*abs(max(app.int_sum)-min(app.int_sum)) + min(
                    app.int_sum);     % the threshold of conceration
231             idx = app.int_sum(:, 1) >= temp;        % comparing to th
232             hits = nnz(idx);                        % count #
233             app.PositiveRatioEditField.Value = hits / app.dropNum;
234         end
235 
236         % Value changed function: OriginalSizeEditField
237         function OriginalSizeEditFieldValueChanged(app, event)
238             value = app.OriginalSizeEditField.Value;
239             app.radius_org = value;
240         end
241 
242         % Value changed function: ConcentrationEditField
243         function ConcentrationEditFieldValueChanged(app, event)
244             value = app.ConcentrationEditField.Value;
245 
246         end
247     end
248 
249     % Component initialization
250     methods (Access = private)
251 
252         % Create UIFigure and components
```

```
253         function createComponents(app)
254 
255             % Create UIFigure and hide until all components are created
256             app.UIFigure = uifigure('Visible', 'off');
257             app.UIFigure.Position = [100 100 1079 796];
258             app.UIFigure.Name = 'MATLAB App';
259 
260             % Create UIAxes
261             app.UIAxes = uiaxes(app.UIFigure);
262             title(app.UIAxes, 'Original')
263             app.UIAxes.TitleHorizontalAlignment = 'left';
264             app.UIAxes.Visible = 'off';
265             app.UIAxes.Position = [32 381 237 175];
266 
267             % Create UIAxes_2
268             app.UIAxes_2 = uiaxes(app.UIFigure);
269             title(app.UIAxes_2, 'Dark')
270             app.UIAxes_2.TitleHorizontalAlignment = 'left';
271             app.UIAxes_2.Visible = 'off';
272             app.UIAxes_2.Position = [32 201 237 175];
273 
274             % Create UIAxes_3
275             app.UIAxes_3 = uiaxes(app.UIFigure);
276             title(app.UIAxes_3, 'Fluorescent')
277             app.UIAxes_3.TitleHorizontalAlignment = 'left';
278             app.UIAxes_3.Visible = 'off';
279             app.UIAxes_3.Position = [32 22 237 175];
280 
281             % Create UIAxes2
282             app.UIAxes2 = uiaxes(app.UIFigure);
283             title(app.UIAxes2, 'Position')
284             app.UIAxes2.Position = [311 312 366 244];
285 
286             % Create UIAxes2_2
287             app.UIAxes2_2 = uiaxes(app.UIFigure);
288             title(app.UIAxes2_2, 'Intensity')
289             app.UIAxes2_2.Position = [311 46 366 244];
290 
291             % Create UIAxes2_4
292             app.UIAxes2_4 = uiaxes(app.UIFigure);
293             title(app.UIAxes2_4, 'Refence square position')
294             app.UIAxes2_4.Position = [676 312 366 244];
295 
296             % Create UIAxes2_5
297             app.UIAxes2_5 = uiaxes(app.UIFigure);
298             title(app.UIAxes2_5, 'Intensity projection')
299             app.UIAxes2_5.Position = [676 46 366 244];
```

```
300
301             % Create BrightFieldImageButton
302             app.BrightFieldImageButton = uibutton(app.UIFigure, 'push');
303             app.BrightFieldImageButton.ButtonPushedFcn = createCallbackFcn(app,
                     @BrightFieldImageButtonPushed, true);
304             app.BrightFieldImageButton.Position = [95 648 120 46];
305             app.BrightFieldImageButton.Text = 'Bright Field Image';
306
307             % Create FluorescenceImageButton
308             app.FluorescenceImageButton = uibutton(app.UIFigure, 'push');
309             app.FluorescenceImageButton.ButtonPushedFcn = createCallbackFcn(app
                    , @FluorescenceImageButtonPushed, true);
310             app.FluorescenceImageButton.Position = [95 580 120 46];
311             app.FluorescenceImageButton.Text = 'Fluorescence Image';
312
313             % Create ofdropletsEditFieldLabel
314             app.ofdropletsEditFieldLabel = uilabel(app.UIFigure);
315             app.ofdropletsEditFieldLabel.HorizontalAlignment = 'right';
316             app.ofdropletsEditFieldLabel.Position = [334 694 71 22];
317             app.ofdropletsEditFieldLabel.Text = '# of droplets';
318
319             % Create ofdropletsEditField
320             app.ofdropletsEditField = uieditfield(app.UIFigure, 'numeric');
321             app.ofdropletsEditField.Position = [420 694 100 22];
322
323             % Create PositiveRatioEditFieldLabel
324             app.PositiveRatioEditFieldLabel = uilabel(app.UIFigure);
325             app.PositiveRatioEditFieldLabel.HorizontalAlignment = 'right';
326             app.PositiveRatioEditFieldLabel.Position = [326 649 79 22];
327             app.PositiveRatioEditFieldLabel.Text = 'Positive Ratio';
328
329             % Create PositiveRatioEditField
330             app.PositiveRatioEditField = uieditfield(app.UIFigure, 'numeric');
331             app.PositiveRatioEditField.Position = [420 649 100 22];
332
333             % Create CalculateIntensityButton
334             app.CalculateIntensityButton = uibutton(app.UIFigure, 'push');
335             app.CalculateIntensityButton.ButtonPushedFcn = createCallbackFcn(
                    app, @CalculateIntensityButtonPushed, true);
336             app.CalculateIntensityButton.Position = [604 615 114 23];
337             app.CalculateIntensityButton.Text = 'Calculate Intensity';
338
339             % Create CalculateConcentrationButton
340             app.CalculateConcentrationButton = uibutton(app.UIFigure, 'push');
341             app.CalculateConcentrationButton.ButtonPushedFcn =
                    createCallbackFcn(app, @CalculateConcentrationButtonPushed,
                    true);
```

```
342             app.CalculateConcentrationButton.Position = [787 615 144 23];
343             app.CalculateConcentrationButton.Text = 'Calculate Concentration';
344 
345             % Create FindSquareButton
346             app.FindSquareButton = uibutton(app.UIFigure, 'push');
347             app.FindSquareButton.ButtonPushedFcn = createCallbackFcn(app,
                    @FindSquareButtonPushed, true);
348             app.FindSquareButton.Position = [809 711 100 23];
349             app.FindSquareButton.Text = 'Find Square';
350 
351             % Create DetermineDropletsButton
352             app.DetermineDropletsButton = uibutton(app.UIFigure, 'push');
353             app.DetermineDropletsButton.ButtonPushedFcn = createCallbackFcn(app
                    , @DetermineDropletsButtonPushed, true);
354             app.DetermineDropletsButton.Position = [601 711 119 23];
355             app.DetermineDropletsButton.Text = 'Determine Droplets';
356 
357             % Create OriginalSizeEditFieldLabel
358             app.OriginalSizeEditFieldLabel = uilabel(app.UIFigure);
359             app.OriginalSizeEditFieldLabel.HorizontalAlignment = 'right';
360             app.OriginalSizeEditFieldLabel.Position = [331 740 74 22];
361             app.OriginalSizeEditFieldLabel.Text = 'Original Size';
362 
363             % Create OriginalSizeEditField
364             app.OriginalSizeEditField = uieditfield(app.UIFigure, 'numeric');
365             app.OriginalSizeEditField.ValueChangedFcn = createCallbackFcn(app,
                    @OriginalSizeEditFieldValueChanged, true);
366             app.OriginalSizeEditField.Position = [420 740 100 22];
367 
368             % Create ThresholdLabel
369             app.ThresholdLabel = uilabel(app.UIFigure);
370             app.ThresholdLabel.VerticalAlignment = 'bottom';
371             app.ThresholdLabel.Position = [285 21 59 22];
372             app.ThresholdLabel.Text = 'Threshold';
373 
374             % Create ThresholdSlider
375             app.ThresholdSlider = uislider(app.UIFigure);
376             app.ThresholdSlider.Limits = [0 1];
377             app.ThresholdSlider.ValueChangedFcn = createCallbackFcn(app,
                    @ThresholdSliderValueChanged, true);
378             app.ThresholdSlider.Position = [345 51 331 3];
379 
380             % Create ConcentrationEditFieldLabel
381             app.ConcentrationEditFieldLabel = uilabel(app.UIFigure);
382             app.ConcentrationEditFieldLabel.HorizontalAlignment = 'right';
383             app.ConcentrationEditFieldLabel.Position = [325 604 80 22];
384             app.ConcentrationEditFieldLabel.Text = 'Concentration';
```

```
385
386             % Create ConcentrationEditField
387             app.ConcentrationEditField = uieditfield(app.UIFigure, 'numeric');
388             app.ConcentrationEditField.ValueChangedFcn = createCallbackFcn(app,
                     @ConcentrationEditFieldValueChanged, true);
389             app.ConcentrationEditField.Position = [420 604 100 22];
390
391             % Create OriginalImageButton_2
392             app.OriginalImageButton_2 = uibutton(app.UIFigure, 'push');
393             app.OriginalImageButton_2.ButtonPushedFcn = createCallbackFcn(app,
                    @OrignalImageButtonPushed, true);
394             app.OriginalImageButton_2.Position = [95 716 120 46];
395             app.OriginalImageButton_2.Text = 'Original Image';
396
397             % Create copiesuLLabel
398             app.copiesuLLabel = uilabel(app.UIFigure);
399             app.copiesuLLabel.Position = [520 604 57 22];
400             app.copiesuLLabel.Text = 'copies/uL';
401
402             % Show the figure after all components are created
403             app.UIFigure.Visible = 'on';
404         end
405     end
406
407     % App creation and deletion
408     methods (Access = public)
409
410         % Construct app
411         function app = droplet_counting
412
413             % Create UIFigure and components
414             createComponents(app)
415
416             % Register the app with App Designer
417             registerApp(app, app.UIFigure)
418
419             if nargout == 0
420                 clear app
421             end
422         end
423
424         % Code that executes before app deletion
425         function delete(app)
426
427             % Delete UIFigure when app is deleted
428             delete(app.UIFigure)
429         end
```

```
430     end
431 end
```