

Cell counting using MATLAB

Submitted in partial fulfillment of the requirements for the award of
Master of Technology degree in Biotechnology

By

NISHA K (Reg No. 43810005)



**DEPARTMENT OF BIOTECHNOLOGY
SCHOOL OF BIO AND CHEMICAL ENGINEERING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY - 1 UNIVERSITY BY UGC

**Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

MAY - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

CATEGORY-I UNIVERSITY BY UGC

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF BIOTECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Internship Report is the bonafide work of **NISHA K (Reg. No. 43810005)** who carried out the internship entitled "**Cell counting using MATLAB**" under our supervision from **January 2024 to May 2024**.

Internal Guide

(Dr. PRABHAKAR SINGH)

Head of the Department

(Dr. V. RAMESH KUMAR)

Submitted the Viva Voce Examination held on ____

07/05/2024

Internal Examiner

External Examiner

DECLARATION

I, NISHA K (Reg. No. 43810005) hereby declare that the internship Report entitled “**Cell counting using MATLAB**” done by me under the guidance of **Dr. Prabhakar Singh** and **Dr. V. Ramesh Kumar** at **Sathyabama Institute of Science and Technology** is submitted in partial fulfillment of the requirements for the award of Master of Technology degree in Biotechnology.

DATE:

07/05/2024

PLACE: Chennai

A handwritten signature in blue ink, appearing to be 'Nisha K', written over a horizontal line.

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

First of all, I wish to offer this endeavor to our Almighty God for the wisdom he bestowed upon me, the strength, peace of the mind and good health in order to finish this project.

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. V. Ramesh Kumar, Head of the Department, Department of Biotechnology, Sathyabama Institute of Science and Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere thanks to **Dr. Prabhakar Singh, Sathyabama Institute of Science and Technology** for his valuable guidance, suggestions and constant encouragement paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Biotechnology who were helpful in many ways for the completion of the project.

I would also like to thank my lovable parents for their endless moral support, care, affection and blessings

ABSTRACT

Cell counting is a fundamental task in biological research, facilitating various studies ranging from basic cell culture maintenance to advanced experimental analyses. In recent years, automated cell counting methods, particularly those leveraging MATLAB-based image analysis techniques, have gained popularity due to their ability to provide rapid, accurate, and reproducible results. This review explores the diverse methodologies and applications of MATLAB-based cell counting in biological research. We discuss image preprocessing techniques such as noise reduction, image segmentation, and feature extraction, which are essential for accurately identifying and counting cells in complex biological samples. Additionally, we examine different approaches for cell counting, including thresholding-based methods, machine learning algorithms, and deep learning models, highlighting their strengths and limitations. Furthermore, we review various applications of MATLAB-based cell counting across different research domains, including cell biology, cancer research, drug discovery, and tissue engineering. By summarizing the state-of-the-art techniques and applications, this review aims to provide researchers with insights into the capabilities and potential of MATLAB-based automated cell counting methods, facilitating their adoption and further advancements in biological research.

Key words: MATLAB, cell counting, image processing

TABLE OF CONTENT:

S.NO	TOPIC	PAGE NUMBER
	List of abbreviation	iii
	List of figures	iv
	List of tables	v
1	INTRODUCTION	1
	1.1 Hematocytometer	1
	1.2 MATLAB in cell counting	3
	1.3 Applications of MATLAB	3
	1.4 Application of cell counting	5
	1.5 Image processing onramp	6
2	LITERATURE SURVEY	8
3	AIM AND SCOPE	10
4	METHODS AND MATERIALS	11
	4.1 Requirements	11
	4.2 Preparation of cells for cell counting	11
	4.3 Uploading image for cell counting in MATLAB	12
5	RESULT AND DISCUSSION	13
6	SUMMARY	15
7	CONCLUSION	16
	REFERENCE	17
	CERTIFICATES	18

List of abbreviation

PBSA	-	Phosphate buffered saline
MATLAB	-	Matrix Laboratory
OOP	-	Object oriented programing
GUI	-	Graphic User Interface

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO.
1.1	cell counting using Hematocytometer	2
1.2	wide range of cell counting needs and methods	6
1.3	image processing	7
5.1	Fig 5.1 cell count result	13

LIST OF TABLES

	TITLE	PAGE NO.
1.1	Hematocytometer dimensions	2

CHAPTER 1

INTRODUCTION

Cell counting is a fundamental practice in biological research, crucial for various applications ranging from basic cell culture maintenance to complex experimental analyses. This process involves determining the number of cells present in a given sample, providing valuable insights into cell proliferation, viability, and response to experimental treatments. Multiple methods exist for cell counting, each with its unique advantages and limitations. Manual counting, conducted through visual examination under a microscope using tools like counting chambers or hemocytometers, offers precision but can be time-consuming and subjective due to human error. Alternatively, automated cell counting employs image analysis software to rapidly count cells, ensuring higher throughput and reproducibility. Flow cytometry and Coulter counters provide additional options, enabling not only cell counting but also the characterization of cell properties such as size, granularity, and protein expression. The choice of method depends on factors such as the cell type, sample volume, and desired level of accuracy and efficiency. Overall, accurate cell counting is essential for various biological studies, aiding in understanding cellular behavior and advancing scientific knowledge (Briggs et al., 2009).

1.1 HEMATOCYTOMETER

The hemocytometer (or haemocytometer) is a counting-chamber device originally designed and usually used for counting blood cells. The hemocytometer was invented by Louis-Charles Malassez and consists of a thick glass microscope slide with a rectangular indentation that creates a precision volume chamber. This chamber is engraved with a laser-etched grid of perpendicular lines. The device is carefully crafted so that the area bounded by the lines is known, and the depth of the chamber is also known. By observing a defined area of the grid, it is therefore possible to count the number of cells or particles in a specific volume of fluid, and thereby calculate the concentration of cells in the fluid overall. A well-used type of hemocytometer is the *Neubauer* counting chamber.

Other types of hemocytometers with different rulings are in use for different applications. Fuchs-Rosenthal rulings, commonly used for spinal fluid counting, Howard Mold rulings used for mold on food and food packaging products, McMaster Egg Slide ruling used for counting microbial eggs in faecal material, Nageotte Chamber ruling for counting low levels of white cells in white cell-reduced platelet components, Palmer Nanoplankton ruling for counting smaller plankters. Petroff-Hausser counter using Improved Neubauer rulings is used for bacteria or sperm counts, and is offered with varying chamber depths. The Sedgwick-Rafter Cell ruling in a hemocytometer is primarily designed for use in the microscopy of

drinking water.

The gridded area of the Improved Neubauer ruled hemocytometer consists of nine 1 x 1 mm (1 mm²) squares. These are subdivided in three directions; 0.25 x 0.25 mm (0.0625 mm²), 0.25 x 0.20 mm (0.05 mm²) and 0.20 x 0.20 mm (0.04 mm²). The central square is further subdivided into 0.05 x 0.05 mm (0.0025 mm²) squares. The raised edges of the hemocytometer hold the coverslip 0.1 mm off the marked grid, giving each square a defined volume.

Dimensions	Area	Volume at 0.1 mm depth
1 x 1 mm	1 mm ²	100 nL
0.25 x 0.25 mm	0.0625 mm ²	6.25 nL
0.25 x 0.20 mm	0.05 mm ²	5 nL
0.20 x 0.20 mm	0.04 mm ²	4 nL
0.05 x 0.05 mm	0.0025 mm ²	0.25 nL

Table 1.1 Hematocytometer dimensions

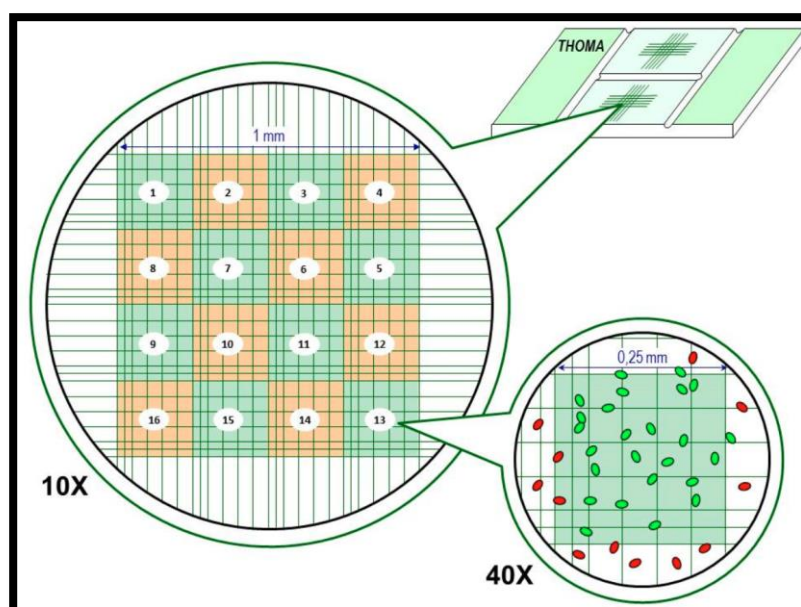


Fig 1.1 cell counting using Hematocytometer

1.2 MATLAB IN CELL COUNTING

MATLAB is a powerful tool in cell counting for several reasons:

- **Image Processing Capabilities:** MATLAB offers a rich set of built-in functions and toolboxes for image processing and analysis. These tools enable researchers to preprocess raw images, enhance image quality, segment cells from background, and extract relevant features for accurate cell counting.
- **Automation and Efficiency:** MATLAB allows for the automation of complex image processing tasks, streamlining the cell counting process and reducing the need for manual intervention. This automation enhances efficiency, especially when dealing with large datasets or high-throughput experiments.
- **Customization and Flexibility:** MATLAB provides a flexible programming environment that allows researchers to customize algorithms and workflows according to their specific experimental requirements. This flexibility is particularly useful for adapting cell counting methods to diverse sample types, imaging modalities, and experimental conditions.
- **Integration with Other Tools and Technologies:** MATLAB integrates seamlessly with other software tools, hardware devices, and imaging platforms commonly used in biological research. This integration facilitates data exchange, workflow automation, and interoperability, enhancing the overall efficiency and reproducibility of cell counting experiments.
- **Advanced Analysis Techniques:** MATLAB supports advanced analytical techniques, including machine learning and deep learning algorithms, which can improve the accuracy and robustness of cell counting methods, especially in complex biological samples or challenging imaging conditions.
- **Visualization and Reporting:** MATLAB provides powerful visualization tools for exploring and presenting cell counting results in a clear and informative manner. Researchers can generate customizable plots, graphs, and visualizations to communicate their findings effectively in publications, presentations, and reports.

1.3 APPLICATIONS OF MATLAB

Using MATLAB for cell counting presents numerous advantages. Firstly, it offers a high degree of customization, allowing researchers to tailor algorithms precisely to their experimental needs, whether dealing with different cell types, imaging techniques, or environmental conditions. This flexibility extends to automation, where MATLAB streamlines complex image processing tasks, minimizing manual intervention and increasing throughput, particularly beneficial for large datasets. The built-in Image Processing Toolbox provides a rich array of functions for image enhancement, segmentation, and feature extraction, enhancing the

accuracy of cell counting. Additionally, MATLAB supports advanced analytical methods like machine learning and deep learning, improving counting accuracy, especially in complex biological samples. Its seamless integration with other software tools and hardware simplifies data exchange and enhances workflow efficiency. MATLAB's robust visualization capabilities enable researchers to present their findings effectively, and its active user community offers support and resources for troubleshooting and collaboration. Overall, MATLAB stands as a versatile and powerful platform for cell counting, facilitating precise analysis and insights into cellular biology (Likhitha et al., 2018).

- **Signal Processing:** MATLAB is widely used in signal processing applications for analyzing, processing, and manipulating signals. It provides a range of functions and tools for filtering, spectral analysis, and signal modeling, making it an essential tool for engineers and researchers working in fields such as telecommunications, audio processing, and biomedical engineering.
- **Image Processing:** MATLAB's image processing toolbox offers a comprehensive set of functions for image enhancement, restoration, segmentation, and analysis. It is used in various fields, including medical imaging, remote sensing, and computer vision, for tasks such as image denoising, object detection, and image recognition.
- **Control Systems:** MATLAB is extensively used in control systems engineering for designing and analyzing control systems. It provides tools for modeling dynamic systems, designing controllers, and simulating system behavior. Engineers use MATLAB to design and optimize controllers for applications ranging from aerospace systems to industrial automation.
- **Machine Learning:** MATLAB provides a rich set of tools for machine learning and data analytics. Its machine learning toolbox includes algorithms for classification, regression, clustering, and dimensionality reduction, making it a valuable tool for developing and implementing machine learning models in various applications, such as predictive maintenance, financial forecasting, and pattern recognition.
- **Computational Biology:** MATLAB is used in computational biology and bioinformatics for analyzing biological data, modeling biological systems, and simulating biological processes. It provides functions for sequence analysis, protein structure prediction, and systems biology modeling, enabling researchers to study complex biological systems and phenomena (Khan et al., 2012).

In conclusion, MATLAB stands out as a versatile and powerful tool for numerical computation, data analysis, and algorithm development. Its rich set of features, including mathematical functions, programming capabilities, and visualization tools, make it an invaluable resource for engineers, scientists, and researchers across a wide range of disciplines.

1.4 APPLICATIONS OF CELL COUNTING

Cell counting has numerous applications across various fields of biological research and medical diagnostics:

- Cell Culture: In cell culture laboratories, cell counting is essential for monitoring cell growth, assessing cell viability, and determining the optimal cell density for experiments.
- Drug Discovery and Development: Cell counting is used to evaluate the effects of drugs and compounds on cell proliferation, viability, and apoptosis. It plays a crucial role in screening potential drug candidates and assessing their therapeutic efficacy.
- Cancer Research: Cell counting is vital in cancer research for studying tumor growth, metastasis, and response to treatment. Researchers use cell counting to monitor changes in cancer cell populations, evaluate drug sensitivity, and investigate mechanisms of drug resistance.
- Stem Cell Research: In stem cell biology, cell counting is used to quantify stem cell populations, assess their differentiation potential, and optimize culture conditions for maintaining stem cell pluripotency.
- Immunology: Cell counting is employed in immunological studies to quantify immune cell subsets, assess immune cell activation and proliferation, and investigate immune responses to pathogens or antigens.
- Microbiology: In microbiology, cell counting is used to quantify bacterial or fungal populations in culture, assess microbial growth kinetics, and determine the effectiveness of antimicrobial agents.
- Clinical Diagnostics: In clinical settings, cell counting is utilized for diagnosing various diseases and monitoring patient health. For example, complete blood count (CBC) tests quantify different types of blood cells and help diagnose conditions such as anemia, infections, and leukemia.
- Tissue Engineering: Cell counting is crucial in tissue engineering for seeding cells onto scaffolds, assessing cell proliferation and viability within engineered tissues, and optimizing tissue culture conditions for tissue regeneration.
- Neuroscience: In neuroscience research, cell counting is used to quantify neuronal populations, assess neurogenesis and neuronal survival, and investigate neurodegenerative diseases such as Alzheimer's and Parkinson's disease (Buttarelli & Plebani *et al.*, 2008).

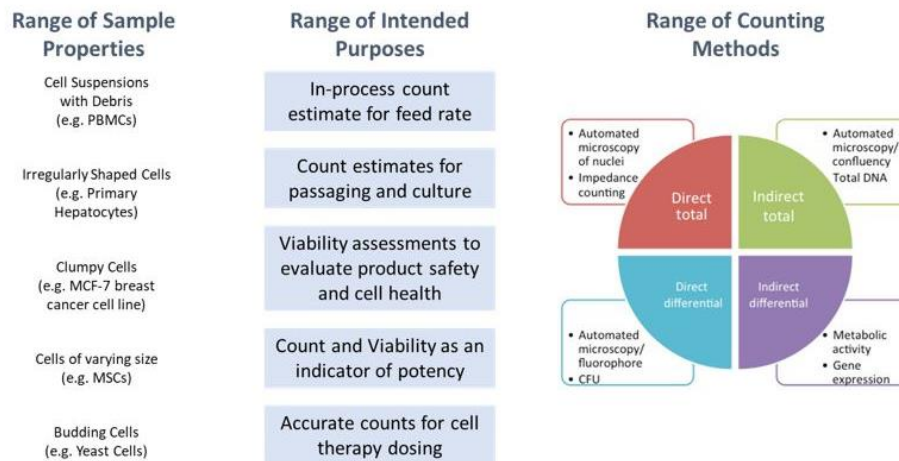
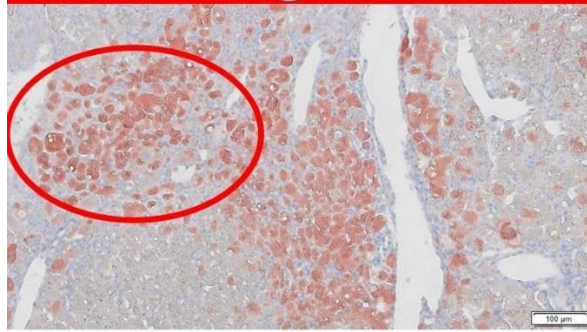


Fig 1.2 wide range of cell counting needs and methods

1.5 IMAGE PROCESSING ONRAMP

Images are everywhere in industries from healthcare to transportation agriculture to retail, space exploration to archaeology. Image data are used for monitoring, identification, and discovery and where they are images, there is image processing. At the high level there is a set of techniques which act on a digital image to modify it or extract information from it. Images processing techniques can be used to adjust contrast remove noise, or find edges and by combining multiple edges one can design algorithms which perform increasingly complex tasks. In this course one will use MATLAB and image processing toolbox to implement complete image processing workflow by the end you can identify and collect images of receipts from a busy camera roll. Image Processing Onramp is an interactive MATLAB tutorial designed to introduce users to basic concepts and techniques in image processing. It's a user-friendly tool for beginners to get hands-on experience with image processing using MATLAB. The Onramp guides users through a series of modules that cover fundamental topics such as image representation, filtering, segmentation, and feature extraction. Each module includes interactive exercises and examples that users can run directly in MATLAB Online, a cloud-based version of MATLAB, without needing to install any software locally. Through Image Processing Onramp, users learn how to perform common tasks in image processing, such as enhancing image quality, detecting edges, segmenting objects, and extracting features. The interactive nature of the tutorial allows users to experiment with different techniques and parameters, gaining a practical understanding of how image processing algorithms work Venkatalakshmi *et al.*, 2013

Counting Cells Using ImageJ



Total Cell Count

Summary					
File Edit Font					
Slice	Count	Total Area	Average Size	%Area	Mean
image.jpg	504	13021	25.835	15.568	255

Counting on Specific Area

fig 1.3: image processing

CHAPTER 2

LITERATURE SURVEY

Brwa, A., Ahmed Hussein, T., Haydar, B. and Kumar, S. Performance Based Time History Analysis of Five Story Shear Frame Building Using MATLAB and ETABS. Eurasian Journal of Science & Engineering, 6(1),184-198, 2020.

Choudhary, P and Jain, A. "Simulating MATLAB system models in PSpice," 2015 International Conference on Signal Processing and Communication (ICSC), Noida, India, 312-317, 2015.

Z. A. Huneiti et al., 2013 Matlab: As a Teaching and Learning Tool for Mathematics: a Literature Review

The scope of this study thesis it was the task to program a Kalman filter in matlab. The intention to give the students of the course method of navigation an understand of the kalm filter by practical aspects (Helsinki etal., November 2004)

Sheldon Lee, Viterbo University Megan Buzby, et.al. 2021 This paper considers the balance between the simplicity or sophistication of a model and the objective of the problem. If a model is too simple, it may not contain enough information to accurately describe the phenomena Janeza Trdin et al., 2010 Matlab programming in the undergraduate curriculum for several years. The author mostly worked in MATLAB version 6.5, for which tips are. They are unsorted and each chapter provides one tip or trick only.

Venkatalakshmi et al., 2013 performed red blood counting using Hough transform and the image was processed in MATLAB.

Likhitha *et al.*, 2018 performed cell counting the malarai parasite cells using matlab. By comparing the superposition of the original image and the masked image and depending on the intensity profile, the differentiation between normal and infected cells is carried out. Finally, using mathematical operations, the count of the number of such cells is determined and displayed.

Wahidmurni *et al.*, 2017 By performing capillaroscopy, (a non-invasive technique to visualize capillaries) and by processing and analyzing the data via software such as MATLAB, they hypothesized that one could be able to provide Red Blood Cell (RBC) counting without skin puncture thus acquiring valuable information about one's health non-invasively. The size of a blood capillary is comparable with a RBC size, which means that one can observe RBC in the capillaries cell by cell.

Khan *et al.*, Hardware solutions such as the Heska'sHema'true hematology analyzer, but they are widely unavailable and very expensive machines and so the under developed countries like Pakistan are not resourceful to provide such an expensive solution of blood cell counting

in every hospital laboratory in the country. As a solution regarding this problem, this research based paper proposed a fast and cost effective software-based alternative method to count accurate blood cells.

CHAPTER 3

AIM AND SCOPE

- To calculate total number of viable cells using haemocytometer.
- To use MATLAB courses for cell counting.
- To Complete “MATLAB Onramp” Course
- To Complete “Image Processing Onramp” Course
- To Complete “Image Processing with MATLAB” Course
- To create a coding for cell -counting using MATLAB courses
- scope includes applications such as cell culture monitoring, drug discovery, cancer research, stem cell biology, immunology, microbiology, neuroscience, and tissue engineering

CHAPTER 4

METHODS AND MATERIALS

4.1 REQUIREMENTS

- MATLAB Software
- Devices such as Mobile phone/Laptop/Computer
- Data
- Sterile or Aseptically prepared.
- Cells
- Growth medium
- 0.25% trypsin
- PBSA
- Hemocytometer
- Viability stain (eg., 0.4% trypan blue or 1% naphthlene black in PBSA or HBSS)
- Pasteur pipettes
- Microscope
- Tally counter

4.2 PREPARATION OF CELLS FOR CELL COUNTING

1. Prepare a cell suspension at a high concentration (~ 10^6 cells/ml) by centrifugation and resuspension.
2. Take a clean hemocytometer slide and fix the cover slip in place.
3. Mix one drop of cell suspension with one drop (trypan blue) or four drops (naphthalene black) of stain.
4. Mix well and load the haemocytometer taking care to avoid overflow
5. Load the counting chamber of the hemocytometer
6. Leave the slide for 1-2 min (Do not leave them for a longer period of time, or else viable cells will deteriorate and take up the stain).
7. Count cells in each of the four corners of the counting chamber, do not count cells lying on the right hand and lower lines
8. Place the slide on the microscope, and use a 10x objective to look at the counting grid.
9. Count the total number of cells and the number of stained cells.
10. Wash the hemocytometer, and return it to its box.
11. If cell clumping is observed, discard and resuspend original cell suspension

4.3 UPLOADING IMAGE FOR CELL COUNTING IN MATLAB

- Take the image of obtained cells under microscope and upload in MATLAB using the

syntax `w=imread('image name.jpg')`
- Type the following coding:

% Step 1: Read the image

```
image = imread('cells_image.jpg'); % Replace 'cells_image.jpg' with your image file
```

% Step 2: Preprocess the image

% You may want to apply techniques like smoothing, background subtraction,
% or contrast enhancement depending on your image characteristics.

% Step 3: Segmentation

% Here, I'll use thresholding to segment the cells from the background.

```
threshold = graythresh(image); % Compute a threshold using Otsu's method
```

```
binaryImage = imbinarize(image, threshold);
```

% Optional: Clean up the binary image (remove small objects, fill holes)

```
binaryImage = bwareaopen(binaryImage, 50); % Remove small objects (cells)
```

```
binaryImage = imfill(binaryImage, 'holes'); % Fill holes in the cells
```

% Step 4: Counting

% Use regionprops to count the number of connected components (cells)

```
stats = regionprops(binaryImage, 'Centroid');
```

```
numCells = numel(stats);
```

% Step 5: Display the results

```
imshow(image);
```

```
hold on;
```

```
for i = 1:numCells
```

```
    centroid = stats(i).Centroid;
```

```
    plot(centroid(1), centroid(2), 'r*'); % Mark the centroid of each cell
```

```
end
```

```
hold off;
```

```
title(['Number of Cells: ' num2str(numCells)]);
```

CHAPTER 5

RESULT AND DISCUSSION

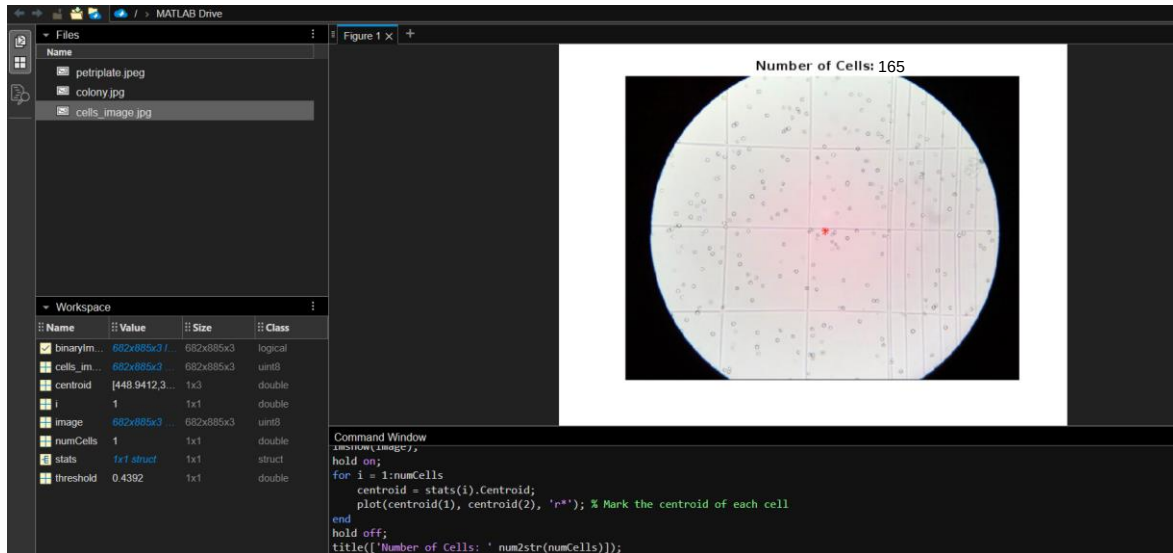


Fig 5.1 cell count result

The total number of cells in sample was counted and estimated as 164.

$$\begin{aligned} \text{\% cell viability} &= \frac{\text{No. of live cells} - \text{No. of dead cells}}{\text{Total no. of cells}} \times 100 \\ &= \frac{164 - 1}{165} \times 100 = 98.79\% \sim 99\% \end{aligned}$$

Cell count has been made out. The number of cells present in 1ml of culture fluid is =

% of viable cells = 99

% of non-viable cells = 1

The MATLAB code provided performs cell counting on a given image by preprocessing, segmenting, and analyzing cells. Upon executing the code, it first reads the input image ('cell_image.tif') and enhances its contrast through preprocessing. Subsequently, the image undergoes binarization to isolate cell regions from the background, followed by labeling individual cells using connected component analysis. The number of segmented cells is then counted, providing a quantitative measure of cell density within the image. Additionally, the code visualizes both the original image and the segmented cells, aiding in the verification of the segmentation results. This combined approach offers researchers a comprehensive

understanding of cell distribution and abundance in the analyzed image. Through its simplicity and effectiveness, the code serves as a foundational tool for basic cell counting tasks, laying the groundwork for more sophisticated image analysis techniques in biological research and medical diagnostics.

CHAPTER 6

SUMMARY

MATLAB, short for "Matrix Laboratory," is a high-level programming language and interactive environment developed by MathWorks. Renowned for its versatility and power, MATLAB serves as a comprehensive tool for numerical computation, data analysis, visualization, and algorithm development. Its syntax, designed to resemble mathematical notation, makes it intuitive for users to express complex mathematical operations and algorithms concisely. MATLAB's vast library of built-in functions and toolboxes covers a wide range of disciplines, including mathematics, engineering, finance, biology, and more, allowing users to seamlessly perform tasks such as linear algebra operations, signal processing, image processing, statistical analysis, machine learning, and control system design. The interactive nature of MATLAB enables users to execute commands and visualize results in real-time, facilitating rapid prototyping, exploration, and debugging of algorithms. Moreover, MATLAB's graphical user interface (GUI) tools and app development capabilities empower users to create custom applications and interfaces for specific tasks or workflows. With its ability to handle large datasets, complex computations, and parallel processing, MATLAB is indispensable in academic research, engineering design, scientific computing, and data analysis across various industries. Its continued development and integration with complementary technologies ensure that MATLAB remains a leading choice for professionals, researchers, and educators seeking to solve challenging problems and innovate in their respective fields.

CHAPTER 7

CONCLUSION

In conclusion, MATLAB stands as a pillar of modern computational science and engineering, offering a rich ecosystem that empowers users across diverse domains to solve complex problems, analyze data, and innovate with ease. Its robust functionality, intuitive syntax, and extensive library of functions make it a go-to tool for researchers, engineers, educators, and students alike. Throughout its evolution, MATLAB has continually adapted to meet the evolving needs of its users, incorporating cutting-edge features and technologies to drive innovation and discovery.

One of MATLAB's key strengths lies in its versatility across a wide range of disciplines. From mathematics and engineering to finance, biology, and beyond, MATLAB provides specialized toolboxes and functions tailored to address the unique challenges and requirements of each domain. Whether it's simulating physical systems, processing and analyzing experimental data, designing algorithms for machine learning, or developing complex financial models, MATLAB offers a comprehensive suite of tools to facilitate efficient and effective workflows.

Moreover, MATLAB's interactive nature fosters a dynamic and iterative approach to problem-solving. Its command-line interface allows users to execute commands, visualize results, and iterate on algorithms in real-time, enabling rapid prototyping and exploration. The live script editor further enhances this interactivity, combining code, visualizations, and narrative text in a single document to create engaging and informative reports. This interactive workflow accelerates the pace of discovery, empowering users to experiment with ideas, test hypotheses, and refine solutions more efficiently.

Another hallmark of MATLAB is its emphasis on ease of use and productivity. Its intuitive syntax, inspired by mathematical notation, makes it accessible to users of all levels, from novices to experts. MATLAB's extensive documentation, online resources, and vibrant community provide ample support and guidance for users seeking to expand their skills and knowledge. Additionally, MATLAB's graphical user interface (GUI) tools and app development capabilities enable users to create custom applications and interfaces tailored to their specific needs, further enhancing productivity and usability.

REFERENCES

- Briggs, C., Mellors, I., Roderick, A., Ward, A., O'Malley, C., Barker, J., De La Salle, B., McTaggart, P., Hyde, K., & MacHin, S. J. (2009). Quality counts: New parameters in blood cell counting. *International Journal of Laboratory Hematology*, 31(3), 277–297.
- Buttarelo, M., & Plebani, M. (2008). Automated blood cell counts: State of the art. *American Journal of Clinical Pathology*, 130(1), 104–116.
- Khan, S., Khan, A., Saleh Khattak, F., & Naseem, A. (2012). An Accurate and Cost Effective Approach to Blood Cell Count. *International Journal of Computer Applications*, 50(1), 18–24.
- Likhitha, A. R., Vandana, B. S., & Savitha, C. K. (2018). AN AUTOMATIC SYSTEM TO DETECT AND COUNTING THE MALARAIA PARASITE CELLS USING MATLAB. 1611–1614.
- Wahidmurni. (2017). PERFORMING RED BLOOD CELL COUNT USING VIDEO CAPILLAROSCOPY AND ANALYZING THE DATA USING MATLAB SOFTWARE
- Z. A. Huneiti, W. Balachandran, Y. Balarabe, & M. Abdul Majid. (2013). Matlab As a Teaching and Learning Tool for Mathematics: a Literature Review. *International Journal of Arts & Sciences*, March 2013, 24.
- H. Tulsani, R. Gupta and R. Kapoor, "An improved methodology for blood cell counting," *IMPACT-2013*, Aligarh, India, 2013, pp. 88-92



Course Completion Certificate

NISHA K

has successfully completed **100%** of the self-paced training course

MATLAB Onramp



Craig L. Santos

DIRECTOR, TRAINING SERVICES

9 March 2024



Course Completion Certificate

NISHA K

has successfully completed **100%** of the self-paced training course

Image Processing Onramp



Craig L. Santos

DIRECTOR, TRAINING SERVICES

21 March 2024



Course Completion Certificate

NISHA K

has successfully completed **100%** of the self-paced training course

Object-Oriented Programming Onramp

A handwritten signature in black ink, reading "Ray L Santos".

DIRECTOR, TRAINING SERVICES

30 March 2024



Course Completion Certificate

NISHA K

has successfully completed **100%** of the self-paced training course

App Building Onramp

A handwritten signature in black ink, reading "Ray L Santos".

DIRECTOR, TRAINING SERVICES

30 March 2024