A Computer-Aided Framework for Cell Phenotype Identification, Analysis & Classification

Subramanian Pradeep

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

A. Lynn Abbott, Chair
Eva Schmelz
Jia-Bin Huang

July 18, 2017
Blacksburg, Virginia

Keywords: Computer Vision, Machine Learning, Cell Elasticity, Cell Phenotype

A Computer-Aided Framework for Cell Phenotype Identification, Analysis & Classification

Subramanian Pradeep

ABSTRACT

Cancer is arguably one of the most dangerous diseases and the major causes of death in the modern day. It becomes increasingly harder to treat and cure the disease as it makes progress. Detecting cancer at an early stage can help in preventing it from affecting an organism. However, it is very hard to detect at an early stage. The best possible way to tackle this disease is to first study it at a cellular level. This study aims at identifying various phenotypic traits of these cells in the Dielectrophoresis (DEP) based microfluidic device experimental setup and subsequently classifying the cells from the rest. A general framework for automatic labeling, identifying and classifying the malignant from the dead cells is developed in this work. The framework shows a top-down approach starting from static background subtraction, tracking, automatic labeling, feature extraction and finally classification. The data used in this work are videos of live and dead human prostate cancer (PC-3) cells flowing through the microfluidic device. Previous studies have shown that there are significant differences in morphological attributes between cancerous and non-cancerous cells. We focus mainly on shape, texture and geometry as the prominent attribute in our work and subsequently use them for classification. In this work we obtain good tracking results through optical flow as compared to previous work. For classification, linear classifiers such as logistic regression and linear Support Vector Machine (SVM) showed decent results. The machine learning algorithms use Histogram of Oriented Gradient (HOG) features plus the elliptical features as a combined feature vector. The elliptic features branch out this study to another direction that is useful in calculation of physical properties such as the cell elasticity through video processing and we propose a model for the same for the given setup. Currently, the elasticity of a single cell is calculated using expensive and time consuming procedures such as the atomic force microscopy (AFM). Using our framework, we can potentially obtain elasticity for a batch of cells in much less time. Also, our cell classification algorithm procedure is suitable for real time applications and can be a proposed futuristic concept for selective killing of cells.

A Computer-Aided Framework for Cell Phenotype Identification, Analysis & Classification

Subramanian Pradeep

GENERAL AUDIENCE ABSTRACT

Cancer is one of the most dangerous disease and a major cause of death in the modern day. It becomes increasingly harder to treat and cure the disease as it makes progress. Detecting cancer at an early stage can help in preventing it from affecting an organism. However, it is very hard to detect at an early stage. The best possible way to tackle this disease is to first study it at the cellular scale. Our study aims at identifying various characteristics of these cells from videos recorded in a dielectrophoresis (DEP) based cell sorting setup. A general framework for identifying and classifying the malignant from the dead cells is developed in this work. We use computer vision algorithms for detection, tracking and analyzing characteristics of the cells. We use these characteristics to classify the cells into live and dead with an accuracy of 95% using standard classification algorithms used in machine learning such as support vector machine and logistic regression. The study of such properties of the cells also enables us to propose a model to estimate Young's modulus of elasticity of the cells. Currently, time consuming techniques such as the atomic force microscopy (AFM) are being used to determine the elasticity of a single cell at a time. Using our work, we can potentially obtain elasticity for a batch of cells in much less time. Our cell classification algorithm procedure is suitable for real time applications and can be a proposed futuristic concept for selective killing of cells.

Acknowledgements

This project was started as a part of the work at PhenoCHIP LLC, a startup in Blacksburg, Virginia, for cell analysis and phenotyping. The main motivation for me towards working on this project is a humble thought that this idea could be very useful for saving mankind from one of the most dreadful diseases. I am very grateful for all the valuable time and support by my advisor Dr. A. Lynn Abbott to conceive and realize the ideas involved in the completion of this thesis project. A lot of credit goes to him for the ideas involved in this project. I sincerely thank Dr. Eva Schmelz for the motivation and ideas involving cell elasticity. She was the reason I branched the project into multiple areas – each of which is an important part of the success of this project. I also thank Dr. Jia-Bin Huang for the constant and prompt support for letting me know the limits of using a certain technique. Special thanks to Mr. Elliot McAllister, General Manager of PhenoCHIP, for making us familiar with the trends, providing us with extensive resources and keeping me aware of countless possibilities and questions. Special thanks to Dr. Rafael V. Davalos and his student Temple Douglas for providing the data and the critical information required for this project. I sincerely thank all the support and advice from all of you. This project is a success because of your valuable time and support.

Contents

List of Figures

List of Tables

Chapter 1

Preface

1.1 Introduction

Analyzing phenotypic traits of different types of cells is a trending research area. Phenotype is the observable external characteristics of an organism that is a result of interaction of the genotype with the surroundings. For example, previous studies show that there are significant morphological differences in a cancerous and a non-cancerous cell. In our study we focus on analyzing phenotypic traits of such cells based on shape, size and texture in images. We also study about elasticity of cells and propose a model to calculate the same using the experimental setup shown in chapter 3. Such properties could be very useful in determining which type of cell we are dealing with. A study [20] showed that there are observable differences in cell membrane shape when viewed under a microscope. Other properties such as the nucleus size and shape also differed between two classes of cells such as shown in [20, 21, 29]. Analyzing the phenotype of a cell has the same importance as studying a genotype. However, it is much more convenient and less expensive to study.

A lot of current research is aimed at analyzing properties of microscopic cells [11, 38]. Experiments involving the dielectrophoresis and microfluidic device have also been studying various types of cells based on phenotype [11, 10]. There are excellent works that show cell group separation on certain fluid flow rates and that certain cells respond to specific frequencies of electric fields [6]. Such experiments have been frequently used for cell counting, sorting and research. Using phenomena such as dielectrophoresis, we can potentially sort and classify cells based on phenotypes – even for cells under the same genotype. This research is especially useful for studying cancer cells and classifying them from benign or even from cells at different stages of cancer. One of the main difficulties faced in cancer treatment is that of identifying the tumor initiating cells (TIC) and then prescribing the appropriate treatment. Such cells have a high degree of overlap in properties among different types of benign and malignant cells. By using a phenotype based profiling and differentiating platform, we can identify the TICs faster than the current techniques. Our work shown in this thesis could potentially be such a platform that can revolutionize the current methods of disease treatment.

In other cell identification methodologies, biomarkers are frequently used to detect and characterize cells. Biomarkers are usually fluorescent substances that highlight the cells and are used for monitoring protein activity in the cellular level that are linked to progression of a disease. Certain cells have surface antigens that do not attract biomarkers. Morphological or a phenotype analysis can be a way around for identification or sorting of such cells. Experiments involving dielectrophoresis based microfluidic devices can be effectively utilized to study phenotypes of such cells. In our work we show how this setup can be utilized in a non-traditional way and suggest some futuristic utilities and possibilities.

1.2 Motivation & Contribution

The main motivation for pursuing this work is that a computer aided classification framework, when combined with the DEP experimental setup from Dr. Davalos' lab (Bioelectromechanical systems lab), can open up new horizons and give a new direction to disease treatment. This setup can not only be used for physical cell separation but potentially also for real time video based cell phenotype identification, classification and calculating mechanical properties such as cell elasticity. Much of the previous research either involved processing images of different types of cells for data collection and classification [12], or physical separation of cell phenotypes through micro-fluidics, flow cytometry [11], dielectrophoresis [4], optical tweezers [44], etc. Our experimental setup, as explained in chapter 3, is a combination of image based phenotype analysis and physical cell separation. The cells travelling through the microfluidic device can be physically separated through dielectrophoresis and the video-recordings of the cell behavior in the device encourages us to develop a software framework for identification and classification of cell populations in real-time. In this work we primarily focus on developing a strong foundation for a complete software framework for cell tracking, segmentation, feature extraction/data collection and classification. This framework can be cascaded with the existing setup to give a drastic boost to its utility.

The main contribution of this thesis is a complete framework for segmentation, tracking, feature extraction and classification of the cells. This further branches out to showing a good utility of the DEP setup, which is to calculate properties such as the elastic modulus of the cells. This application is a potential improvement to the previous techniques such as the Atomic Force Microscopy (AFM) for determining the elastic modulus of cells. AFM based technique is expensive and time consuming. We also propose futuristic applications of our cascaded setup which can potentially give a new direction to the existing treatment methods of diseases such as cancer.

1.3 Thesis Organization

This thesis is organized into nine chapters. Each chapter, apart from the first chapter, starts with relevant introductory literature survey. We start by discussing background of cell morphology, physics and dynamics useful for this experimental setup and our research. The most important concepts relevant to our application are fluid dynamics, contact mechanics and cell morphology. In chapter 3 we have a brief discussion of the experimental setup and the design of the micro-fluidic device. Chapter 4 is the crux of this thesis. It explains the framework in a top-down approach discussing every concept used and how they have been applied to our data. It starts by giving a generic overview of the framework as a block diagram and moves to describing each block in separate sections. This includes segmentation, tracking, automatic class labeling, feature extraction and classification. In chapter 5 we discuss the important features that further explain the morphological aspects of the cells and also branches out the thesis to calculation of Young's Modulus of Elasticity which is described in chapter 6. Chapter 7 is fully based on qualitative and quantitative evaluation of the framework and the algorithms within. We finally conclude with a summary of our work and with useful suggestions for future work.

Chapter 2

Background information

2.1 Cell Morphology and Phenotype

A lot of study has been conducted on the external morphology of different types of cells. Works such as [1, 2, 3] show that malignant and benign cells have differences in elastic properties under same circumstances. For example, certain cancer cells are soft and spongy and show irregular shapes [3]. Their nuclei have irregular shapes and are larger than nucleus of a benign cell [29]. The benign cells are compact and tend to retain shape. They have well defined cell membranes and are semi-permeable. Such differences have been a major motivation for cell separation through flow cytometry [11], dielectrophoresis and cell manipulation and separation techniques such as optical tweezers [44].

Dielectrophoresis has been shown to be one effective method for separating cells of different populations [6, 33]. In these experiments the cells are passed through a microfluidic device. The cells gain momentum from the fluid flow in the device. Electrodes in the device manipulate the cells through electric field at different frequencies. As an example, malignant cells tend to stick to posts when an electric field of certain frequency is applied. However, the dead cells do not respond to such frequencies. There is some degree of overlap between two sub classes of cells, for example, circulating tumor cells CTCs [28] and benign cells. Such overlaps make it difficult to separate such populations.

Fluid dynamics makes up an essential part in our experimental setup to calculate physical properties such as elasticity. Extensive details about tumor cell morphology are shown in [20]. They show that cancerous cells have intensely colored or, on contrary, a pale cytoplasm along with irregular sizes and shapes and differences in tissue surface texture. They also show prominent differences in organelles of the cell such as in mitochondria, golgi apparatus, etc. Another work [38] analyzes phenotypic differences from microscopy images using transport based morphometry. They show how feature descriptors in image processing are useful for differentiating the cells. They show an automated approach that combines some of the statistical pattern recognition techniques along with the mathematics of optimal mass transport.

Information about differences in cell morphology is utilized in our work for classifying two types of cells. We primarily keep ourselves limited to differentiate cells based on shape and size characteristics. We use feature descriptors such as histogram of oriented gradients (HOG) and ellipse fitting to classify cells purely based on geometrical shape and surface texture. Our data uses relatively low quality data in terms of resolution, frame rate and magnification which is easier to obtain and computationally faster to process through software. The next few sections have a formal introduction to the concepts and mathematics that were useful for our research.

2.2 Fluid Dynamics

In a microfluidic device, the cells (Figure 1) gain momentum due to the fluid flow and hit the cylindrical posts. The cell(s) under the influence of the fluid flow experience drag due to dynamic viscosity of the fluid in the direction of fluid flow.

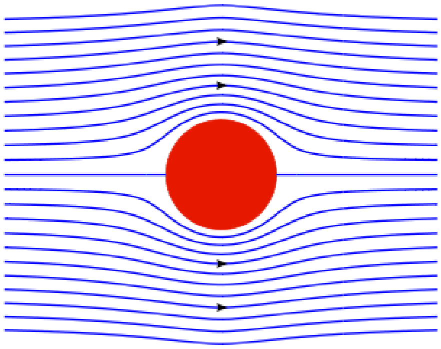


Figure 1. Illustration of a cell (in red) moving in a creeping fluid flow (from left to right). Source: Wikipedia (fair use).

Considering creeping flow, the expression for resultant force on a spherical particle (a cell) will be given by the Naiver-Stokes' equation:

$$m_c \frac{dv_c}{dt} = 6\pi\gamma_F R(v_F - v_C) + F_s \qquad (1)$$

where γ_F is the dynamic viscosity of the fluid, R is the radius of the spherical particle, v_c is the velocity of the cell/spherical particle, v_F is the velocity of fluid flow, m_c is mass of the cell (assuming uniform mass density) and F_s includes other external forces (such as gravity, etc.) on the cell, assumed to be constant and zero later in the thesis (chapter 6). It is assumed that the cells move nearly in a straight line between the cylindrical posts. The magnitude of resultant velocity and scalar speed will thus have the same value between consecutive frames.

Upon solving the above differential equation (1) we can obtain an equation for velocity of a spherical particle under the influence of fluid flow. This time dependent equation is given by:

$$v_c = (v_{ci} - v_F - \frac{F_s}{6\pi\gamma_F R})e^{-(6\pi\gamma_F R/m_c)t} + v_F + \frac{F_s}{6\pi\gamma_F R} \tag{2}$$

The force expression $6\pi\gamma_F R(v_F - v_C)$ in (1) is the Stokes' drag (Stokes' force or the viscous drag) [47]. Further, we consider the cell as an oblate spheroid because of lack of information of depth axis of the cells. Oblate spheroid is an ellipsoid in which two of the three principal axes are equal as shown in Figure 2. Moreover, we assume that the major axis of the cell is aligned with the direction of fluid flow between the cylindrical posts. As shown in [45] the force expression will then be given by:

$$m_c \frac{dv_c}{dt} = 6\pi\gamma_F aK(v_F - v_C) + F_s \tag{3}$$

where

$$K = \left(\frac{8}{3}\right)\frac{(\beta^2 - 1)}{\frac{\beta(3\beta^2 - 2)}{\sqrt{\beta^2 - 1}}\tan^{-1}(\sqrt{\beta^2 - 1}) - \beta} \tag{4}$$

where β is the ratio of semi-major axis a of the cell to the semi-minor axis b.

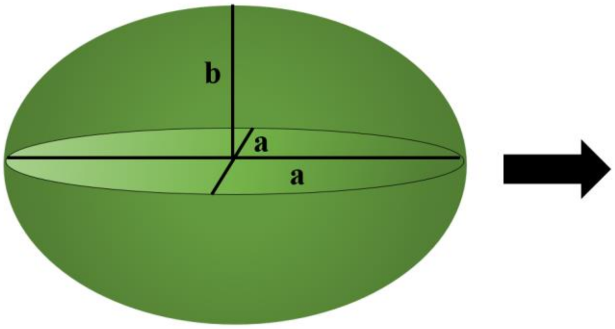


Figure 2. A spheroid moving in direction left to right. The major axis is aligned to the direction of fluid flow which is same as the horizontal axis.

It is assumed that v_{ci} is the initial velocity of the cell. This equation implies that the cell velocity, after a large time frame, will equal the non-exponential term of the above expression given for velocity. Gravity and other external forces are not considered in our work. There is no information about the depth at which the cells move in the microfluidic device. Effects of gravity, although negligible, can be included to explain the cell dynamics along the depth axis. The above equations are sufficient to describe a suitable and simple model of the cell dynamics in the experimental setup, and the relevant variables are derived through video processing.

2.3 Applied Mechanics

The elasticity of cells is considered to be a good characteristic property for differentiating their phenotype. For example, as shown in [3], some cancerous cells are softer than their benign or non-malignant counterparts. We utilize the Hertz model for analyzing the contact mechanics that occur in our system between the pillars and the cells [1]. In the given micro-fluidic device experiments [4], the cells and the posts form a sphere – cylinder contact model and have an elliptic contact surface area with eccentricity e. For an indentation x on the sphere due to hitting a pillar [5], the model equation is given as follows:

$$x = \frac{2QF}{\alpha} \int_0^{\pi/2} \frac{d\theta}{\sqrt{1 - e^2 \sin^2 \theta}} \tag{5}$$

where F is the force causing the indentation and Q and α are given by the following expressions for a Poisson's ratio of 0.5:

$$Q = \frac{9}{16\pi E} \tag{6}$$

$$\alpha^3 = 2QFa \int_0^{\pi/2} \frac{\sin^2 \theta \, d\theta}{\sqrt{1 - e^2 \sin^2 \theta}} \tag{7}$$

where a is the semi-major axis of the cell and is considered here as equivalent diameter of a sphere. E is the Young's Modulus of elasticity of the cell coming in contact with the cylindrical pillar. As shown above, the expression for the indentation is dependent upon elliptic integrals of first and second class that do not have a closed form solution. They are dependent upon the value of e and usually approximated as power series in terms of Legendre polynomials.

Such models are very rare and hardly used in practice [5]. Thus, we estimate the contact mechanics as a sphere-sphere collision model as shown in [1, 2, 3] which could provide us with close to ideal results. This model is practical and widely used in atomic force microscopy (AFM) experiments.

The setup used to study the cells can be used for very interesting applications such as to calculate elasticity of the cells. Currently AFM is put to use for calculating physical properties of cells as shown in works like [1, 2, 3]. It takes a long time and care to calculate the elastic modulus of cells using this technique. Typically, the AFM techniques use conical, parabolic and spherical indenters to poke the cell and obtain measurements of elasticity based on the indentation. Our setup has better prospects of doing the same task for multiple cells in a short period of time. The elasticity of the cells is estimated here using the Hertz contact model for two spheres of unequal sizes for a Poisson's ratio of 0.5.

The Hertz model for elasticity for two colliding spheres is given by:

$$F = \frac{16}{9} E \sqrt{\frac{RR_p}{R + R_p}} x^{3/2} \tag{8}$$

where E is the Young's modulus of elasticity of a cell, R is the radius of a cell just before collision, R_p is the radius of the pillar/posts, x is the indentation caused on the cell due to the normal force F upon hitting the rigid pillars.

For a cell hitting a rigid flat surface, the expression in (5) remains the same apart from the equivalent radius which is simply replaced by the cell's radius as $R_p \to \infty$. The force acting on the cell is mainly the Stokes' force obtained from the Naiver-Stokes' equation for creeping flow. In later chapters we will see how we can calculate the above values using video processing.

For a more detailed analysis, we also investigated the sphere-cylinder Hertz contact model for the cell and pillar linking it to an expression given in [5]. The expression for indentation caused by the force arrives at a non-converging elliptic integral. However, we can conclude that the indentation on the cell, as given in the previous section, has an exponential relation to the Young's modulus of elasticity for an ellipsoid-cylinder contact model. The equations in the previous section show that elasticity of the particle or body can be described with its dimensions just before the collision occurs and indentation upon collision. All forces in this experiment are considered to be along the horizontal axis (x-axis). Forces along the vertical axis (y-axis) are negligible throughout the collision instances because majority of the forces caused due to the fluid flow is along the horizontal axis in the assumption of a creeping flow. In chapter 6, we discuss more about the Hertz model and how it can be applied to the system to calculate elasticity.

2.4 Dielectrophoresis

Dielectrophoresis is a phenomenon (shown in Figure 3) in which force is exerted on a dielectric particle under the influence of a non-uniform electric field. It is highly dependent on the dimensions, polarizability of the dielectric particle and frequency of the electric field and does not require the particle to be charged unlike electophoresis. Dielectrophoresis has shown to be an effective method for cell separation or isolation as shown in [4, 6, 36]. This phenomenon is very useful for selective isolation of different populations of cells. This is because different populations of cells respond to different frequencies of electric field. Although some degree of overlap is seen in some of the sub-populations of cells which makes it harder to separate. Our work utilizes dielectrophoresis to analyze and record characteristics of cells in the setup.

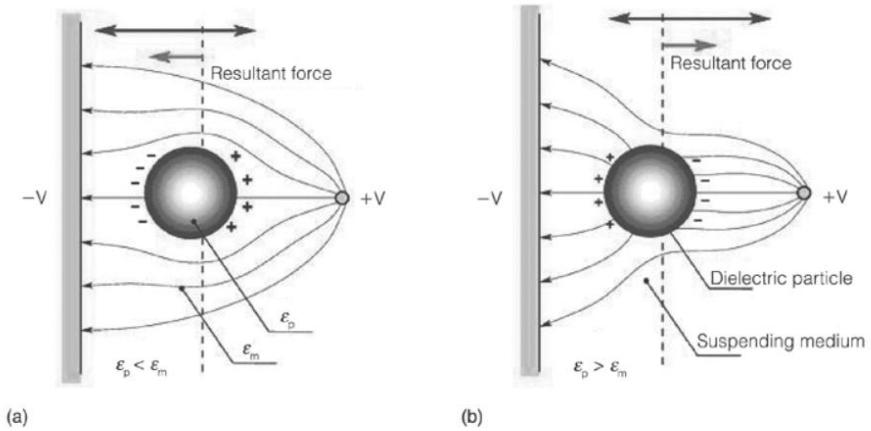


Figure 3. Image showing positive and negative DEP force in a) and b). Courtesy [46] (fair use).

A simple model to define the dielectrophoretic force is to assume the particle under influence to be a homogenous sphere which is surrounded by a conducting dielectric medium. As shown in [4], for a sphere of radius R in a medium with complex permittivity ϵ_m the time averaged dielectrophoretic force is given by:

$$< F_{DEP} > = 2\pi R^3 \epsilon_m \, Re[f_{CM}] \nabla |E_{rms}|^2 \tag{9}$$

where $\nabla |E_{rms}|$ represents the local field gradient, $Re[]$ represents the real part and f_{CM} is the frequency dependent Clausius-Mossotti factor given by:

$$f_{CM} = \frac{\epsilon_p - \epsilon_m}{\epsilon_p + 2\epsilon_m} \tag{10}$$

where ϵ_m and ϵ_p are the complex permittivity of the medium and the particle, respectively. The complex permittivity is defined as follows:

$$\epsilon = \varepsilon_r - j\frac{\sigma}{\omega} \tag{11}$$

where σ is the conductivity of the material, ε_r is the permittivity and ω is the frequency.
A similar expression for dielectrophoretic force can be given for particle with shapes of ellipsoids of revolution (or spheroid). An expression for a field aligned oblate spheroid with two equal principal axes gives us a more general version of the time dependent force:

$$F_{DEP} = \frac{\pi a^2 b}{3}\epsilon_m\, Re\left[\frac{\epsilon_p - \epsilon_m}{\epsilon_m}\right]\nabla|E|^2 \tag{12}$$

where a is the semi-major axis of the spheroid and b is the semi-minor axis.

Dielectrophoresis is mainly used for analyzing electrical properties of the cells. These are useful measures of cell phenotypes, which can be used for classification and separation. For example, measuring particle velocities in the electric field gradient can be used to characterize sub-populations of cells that otherwise show common properties of belonging to the same family. The Clausius-Mossotti factor determines the effective polarizability of the particle. It is dependent upon the frequency of the applied electric field and ranges between $+1$ to -0.5 and holds true for spherical particles. The direction of force is dependent upon the sign of the CM factor. The DEP experiments usually involve analyzing cell responses to changing frequencies and voltage applied. The cross-over frequencies of positive and negative DEP forces is of keen interest. This could be a useful measure of type of cell under consideration especially within the same type of population (e.g., stages of cancer cells). In later chapters we will see how dielectrophoresis is utilized for feature extraction and automatic class labeling of cells in the microfluidic device.

Chapter 3

System setup

3.1 Introduction

Dielectrophoresis [35] has been an extensively used as a technique for cell separation and isolation. For utilization of this phenomenon an experimental setup is designed by many researchers, which is usually a pillar based micro-fluidic device, and cells are passed through it with the help of a buffer fluid at a certain flow rate. One such work is shown in [7] which shows the fabrication process for creating microchannels in the microfluidic device. The device is used for controlling xylinum cells through dielectrophoretic forces in the microchannels. Another work [4] shows an extensive discussion of their design of microfluidic device. Their setup is primarily used for selective isolation of live and dead cells through contactless dielectrophoresis. They present two microfluidic devices capable of separating live leukemia cells from dead cells utilizing their electrical properties.

In [33] the researchers design a microfluidic chip of the same order as the size of the cells. They show that these cell-scale structures improve viability, trapping efficiency and throughput while reducing pearl chaining [36]. In their work it is shown that only one or two cells are trapped around the pillars (or posts) and the rest are carried away by fluid flow. There are improvements in viability of off-chip cell collection and analysis. Another great advantage of using smaller pillars is for its application in our work through our classification framework and physical quantity evaluation. If only one cell hits a particular pillar at a time, it becomes easy to target a specific cell and analyze its properties through video processing. The pillar array based microfluidic device can thus be extended to analyzing multiple individual cells hitting multiple respective pillars. Earlier works [36] showed how the contactless dielectrophoresis (cDEP) was designed by electrodes not being in contact with the sample fluid. Electrodes are capacitively coupled to a fluidic channel through dielectric barriers. Application of high frequency electric fields to these electrodes induces electric field in the channel. Further, a recent work [37] shows an alternative cDEP design to facilitate cell isolation for identification by Raman spectroscopy. This study presents an alternative way to fabricate a cDEP device allowing for higher operating voltages, improved replication, and the opportunity for analysis using Raman spectroscopy. In this device, channels were formed in fused silica rather than PDMS. The device successfully trapped 3.3 µm polystyrene spheres for analysis by Raman spectroscopy. The successful implementation indicates the potential to use cDEP to isolate and identify biological samples on a single device.

There is more such research that shows design of an ideal cDEP based microfluidic device. For our work, we found data in [33] to be a good option for video based classification and other useful applications such as developing a model for calculating Young’s modulus of elasticity of the cells. Video based detection and classification would become much harder for cases where there are clusters of cells collected around the post or overlapping each other. The next section gives a brief introduction about the experimental setup and the devices used to collect raw videos that have been used in this thesis.

3.2 Experimental Setup

The experimental setup to obtain the data is similar to that shown in [28] (Figure 4). It mainly consists of a function generator, oscilloscope, amplifier, the fluid pump, microfluidic-device and a microscope. The function generator is used to generate signals of the required frequency and for turning on the DEP field. The frequency of the signal is monitored using an oscilloscope and is amplified using an amplifier. The pump station is responsible for inlet of the fluid and the required cells. The fluid flow rate is set at a value sufficient enough that the cells collide with the insulated pillars in the microfluidic device.

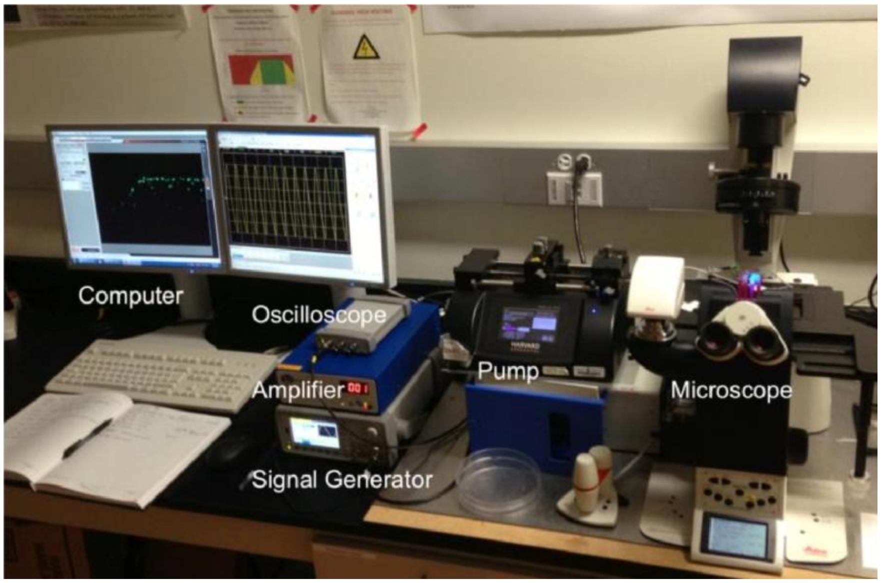


Figure 4. Image showing the system setup. Courtesy [28] (fair use).

The microfluidic chip is shown in Figure 5. The average dimensions of a chip are 50mm x 75mm. Internally, it consists of posts or pillars which are 20-100μm diameter [33]. The DEP field gradient is largest around these posts. To explain the response of cells in simple words, the ones that respond to the DEP field are dragged towards the posts and the cells that do not respond are simply carried away by the fluid flow. There are cell reservoirs or syringes that store the required type of cells and mix them with the fluid so that they flow through the device.

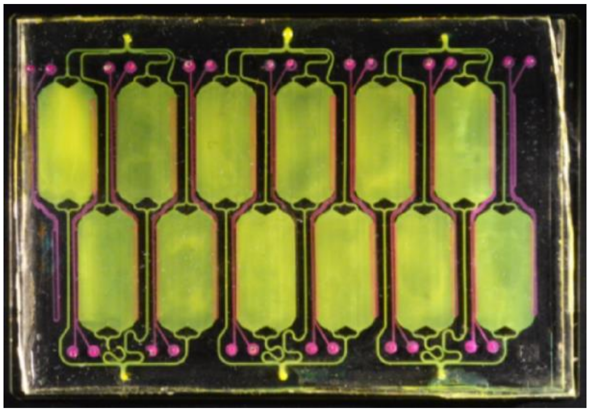


Figure 5. The microfluidic chip. The average dimensions of a chip are 50mm x 75mm. Courtesy [31, 32] (fair use).

The microfluidic device and the pillars are same as shown in [33]. Figure 6 shows the 100μm and 20μm pillars as visible in a microscope. It shows the instances when the DEP field is ON and cells are attracted to posts. The microfluidic chips are made layer by layer through lithography. Each layer has thickness in the order of micrometers. More details for the design can be found in [4, 33, 34].

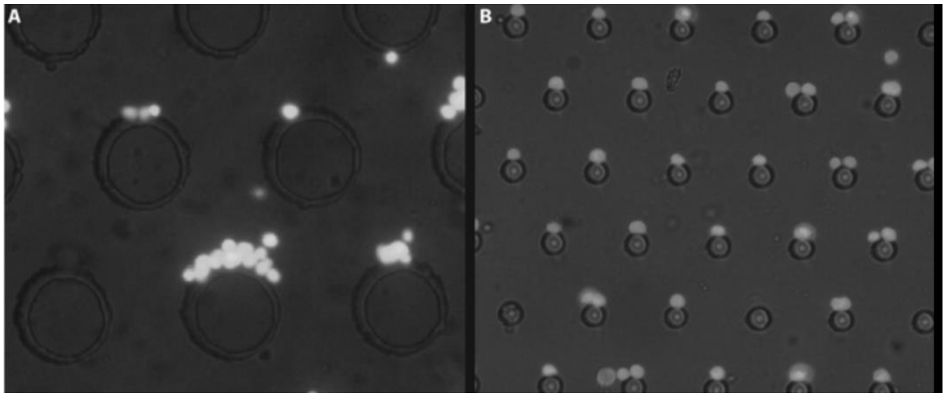


Figure 6. Image on the left shows 100μm pillars and the one on the right shows 20μm pillars. Courtesy [33] (fair use).

The inflow and outflow of the fluid and cells is controlled by a pump station. The pump regulates the fluid flow. In our experiments, fluid flow is set to a suitable value that causes the cells to move and hit the posts to cause some degree of deformation in their walls. During this process, videos are recorded from a microscope (shown in Figure 7) and used for further processing. The cells separate and exit out to a collector. There are two separate collection chambers in which two sorted populations of cells are collected [31].

Figure 7. The microscope used for video data collection [33] (fair use).

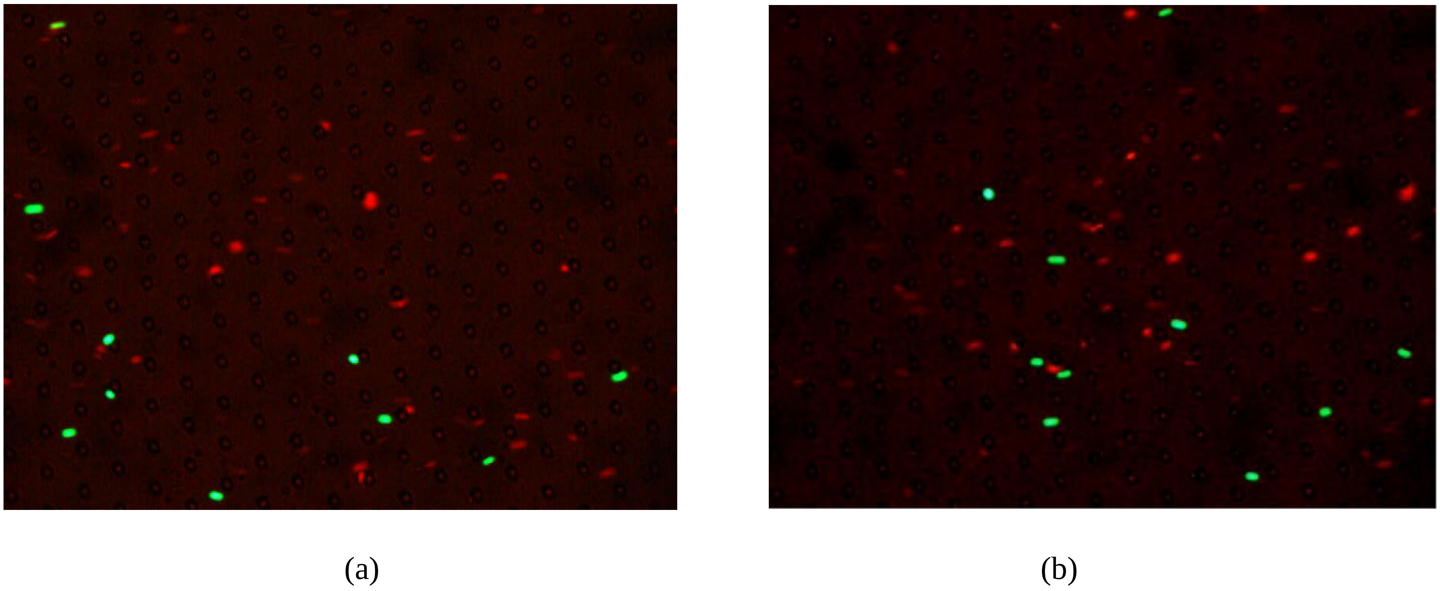
In this project, the camera quality of the microscope is of utmost importance. The accuracy of algorithms and software designed is dependent upon the resolution, frame rate and magnification of images recorded using this video recording device. The camera specifications are given below:

- Mega Pixels: 3.3
- Sensor: ½” CMOS
- Max Resolution: 2048x1536pixels
- Scan Area: 6.55mm X 4.6mm
- Pixel Size: 3.2umx3.2um
- Color: 24Bit
- Data Format: JPEG
- OS: Windows 7, 8.1
- Compatibility: USB 2.0
- Power supply: via USB 2.0
- Fps: 30 frames per second

3.3 Raw Data & Tools Used

The microscope is an integral equipment for raw data collection. In this thesis the cells that are being used are the PC-3 cells which are the human prostate cancer cell line. These cells are extensively used in investigating the biochemical changes in advanced prostatic cancer cells and in assessing their response to chemotherapeutic agents. These cells have a high metastatic potential. From a morphological point of view, electron microscopy revealed that PC-3 show characteristics of poorly-differentiated adenocarcinoma. They have features common to neoplastic cells of epithelial origins, such as numerous microvilli, junctional complexes, abnormal nuclei and nucleoli, abnormal mitochondria, annulate lamellae and lipoidal bodies.

The camera records videos of these cells moving through the microfluidic device, and we use these videos as our raw data for the framework. The videos were obtained from Dr. Davalos' lab (Bioelectromechanical Systems Lab at Virginia Tech). Example frames from these videos are shown in Figure 8. These videos are named 'video 1.avi' and 'video 2.avi'. They are in AVI format (AVI stands for Audio Video Interleave). Video 1 has 936 frames and video 2 has 190 frames. They are 30 fps videos with a resolution of 864x648. The camera captures JPEG (Joint Photographic Experts Group) images, which is a commonly used lossy image compression format. These cell sizes are in the range of 10-25 μm diameter. The two cell populations are colored green and red, which correspond to PC3+ and PC3− cells. The +/− symbols denote live (malignant) and dead respectively. The cells in the video are a heterogeneous set of cells which could have sub-populations of malignancy. However, in our work we only divide the data into cancerous (the green cells) and dead (the red cells). In our algorithms, we do not consider color (green and red) as a characteristic feature to differentiate the cells. Cell color is simply due to staining and is useful for enhancing contrast of the cell and its components when viewed under a microscope. One can extend this work to identifying sub-populations of cells (e.g, stages of malignancy) using other extensive properties such as the extent of response to DEP field. For accomplishing our task, we primarily used MATLAB in which most of the framework has been implemented. We make use of FIJI [43] for analyzing ground truth and evaluation of algorithms.



(a) (b)

Figure 8. Data sets used in this work: (a) Video 1. (b) Video 2.

Chapter 4

The proposed framework

4.1 Introduction

The framework proposed in this thesis is a software based system for identification and classification tasks, especially when combined with the experimental setup described in chapter 3. This framework is the crux of the whole thesis and can potentially take the existing DEP separation technology to a new level.

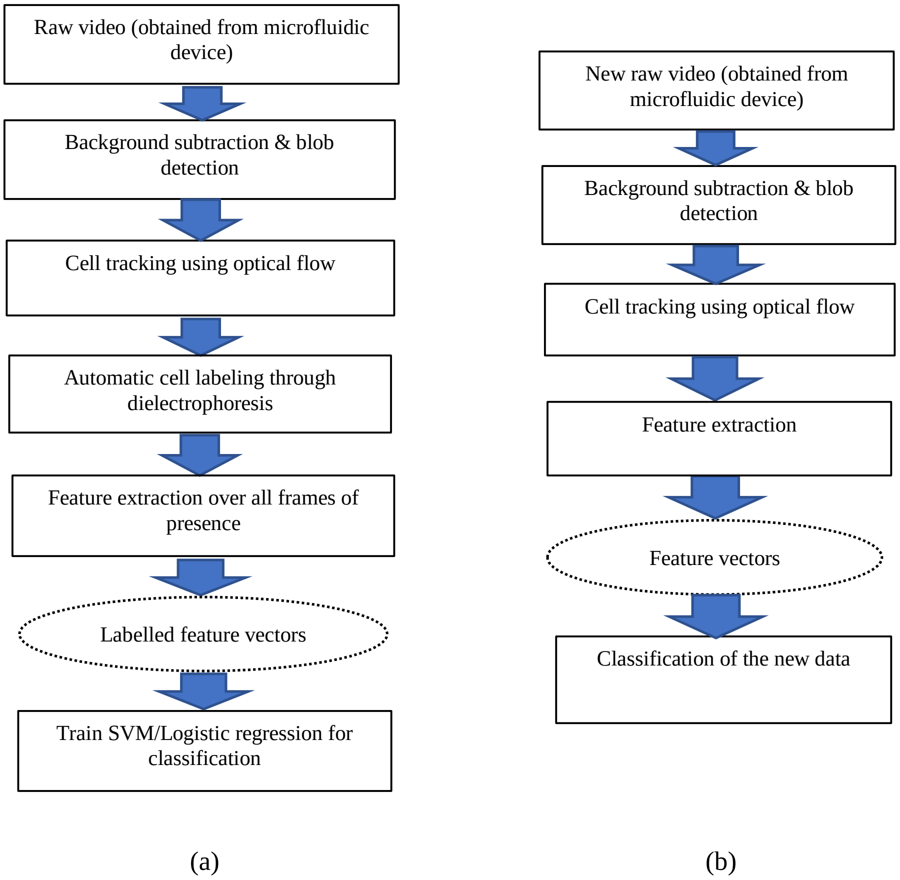
There are previous works that analyzed different types of microscopic image data on cellular organisms. One such work is shown in [8] where they study neuroblastoma – a cancer frequently occurring in children. Image analysis techniques were applied in this study to histological images of haematoxylin and eosin stained slides. Texture features were analyzed from the slides and 87.88% of classification accuracy was reported. Another recent work shown in [9] used the conventional deep neural network in label-free classification of cells. The work shows integration of feature extraction and deep learning with high-throughput quantitative imaging enabled by photonic time stretch [48], achieving an accuracy of 95.5% in label-free cell classification. They compare different classification algorithms like the logistic regression, support vector machine and the artificial neural network to classify white blood T-cells and colon cancer cells. Similar to [11], they make use of techniques in imaging flow cytometry to extract optical and geometric features. As compared to these works, we have shown that the pillar based microfluidic device allows for a less expensive and less time consuming means to extract image features temporally. More importantly, our work is the next generation of the most recent work shown in [32], in which comparison of blob detectors and trackers was performed. Our work not only shows improvements in the detection and tracking, but also extends the system setup to a complete image based phenotype analysis framework for cells.

A recent work of Bal et al. [10] has shown the usage of image processing for characterizing cells based on deformity. They make use of dielectrophoresis for stretching the cells to the elastic limits, and they ascertain deformity through actin cytoskeletal modeling. The device based on DEP is shown to be low cost and effective for such experiments and applications in this work. However, this work is time consuming as it is performed on one cell at a time. Another recent work by Di Carlo et al. [11] show how deformity in cells can be measured by using multi-parameter deformability cytometry (m-DC). They make use of high speed bright field photography to obtain very high frame rate data, claimed to be up to 500,000 frames per second. They further used image processing to obtain shape-size and morphological features. Finally, they make use of a support vector machine to classify cells using a combination of extracted features. They have a throughput of ~2000 cells per second. Practical application of techniques such as bright field photography can be very expensive. Nevertheless, it is useful in collecting reliable data.

Other works as in [12] show preliminary classification systems for feature extraction and classification of magnified leukemia cells from white blood cells. They make use of image segmentation techniques to identify and locate cells. Further, they extract features of the static cells in terms of shape-size, texture, color and statistical features and use reinforcement learning (Q-learning) to classify the cells in four sub-classes. [13] shows approaches to track motion of blood cells using optical flow techniques – template matching and then SIFT based matching and tracking. The standard optical flow technique is shown to work effectively when there is instability in velocity of cell being tracked. Meanwhile, SIFT is useful for tracking large displacements between frames.

There have been other such works [29, 30] that follow similar techniques in their respective data and have different applications. However, it is seen that almost all of these works aim at identifying and locating different types of cells for classification. Most of these works have either high resolution, magnified or high frame rate data for their analysis. Our work is shows a practical approach with less stringent requirements of resolution, frame rate or magnification. We show that such tasks are also achievable with a lower quality and easy to obtain data that is fast to process, has a good degree of reliability and suitable for real-time applications. Although video data with a high resolution, high frame rate or high magnification will be beneficial for calculation of physical quantities such as elasticity.

The work shown in this thesis is similar to previous works in terms of the primary motivation of detecting, tracking or classifying the cells. However, it is a combination of these methods for temporal feature extraction and more detailed work in terms of analyzing intrinsic physical properties of cells. In this chapter we show a complete classification framework for the given setup. It involves obtaining raw videos of cells from the microfluidic device, cell detection through blob analysis, tracking the cells, automatic labeling, feature extraction (temporally over frames) and finally classification. This framework is very suitable for real time applications which automates the collection of labelled quantitative data, unlike other works that manually label image data. It can also be applied to other video based cell analysis experiments due to the aggregation of all required methods such as detection, tracking, feature extraction and classification. The block diagram in Figure 9 shows a high level description of our proposed framework.



Raw video (obtained from microfluidic device)
Background subtraction & blob detection
Cell tracking using optical flow
Automatic cell labeling through dielectrophoresis
Feature extraction over all frames of presence
Labelled feature vectors
Train SVM/Logistic regression for classification
(a)
New raw video (obtained from microfluidic device)
Background subtraction & blob detection
Cell tracking using optical flow
Feature extraction
Feature vectors
Classification of the new data
(b)

Figure 9. The classification framework. (a) Block diagram for the training phase. (b) Block diagram for testing phase.

The system described in the figure above takes raw videos of PC-3 (live and dead) cells flowing through the microfluidic device. Static background subtraction is performed in every frame to approximately locate or isolate the cell blobs from the background. We then use Difference of Gaussian (DoG) filter for performing blob detection and locating positions of the blobs (cells) in every frame. The next step is to track the cells over the frames. Multiple object tracking is achieved through optical flow, i.e., using a Kanade-Lucas-Tomasi (KLT) [15] tracker in combination with Hungarian Algorithm [16] for data association. This method gives a good tracking accuracy on the given data.

The next stage of this framework is the automatic labeling and feature extraction. In our experiments one class of cells are attracted to the pillars in the microfluidic device when the dielectrophoretic field is turned on. We use this fact to collect labelled feature vectors of the two classes of cells. It is seen that the live-cancerous cells (or green cells) respond to the electric field and are attracted to posts, but the dead/benign (red cells) are not. We aim at analyzing differences between the two broad categories. After performing automatic class labeling, we extract features of each cell temporally over the frames until they disappear. We mainly focus on HOG features and geometric characteristics through ellipse fitting in this framework. Other features such as color are not considered as a part of this system, and we focus only on texture and shape based features for a better generalization. Through automatic labeling and feature extraction, we automatically generate labelled feature vectors which can be used for classification.

The final stage of this framework is to train a classifier from the obtained data. The HOG features combined with elliptic features are used by a support vector machine and logistic regression for classification. We discuss the quantitative and qualitative analysis of these classifiers in a later chapter in this thesis. We can now use the trained model on the test data set and identify two different classes of cells (Figure 8(b)).

4.2 Background Subtraction

To detect the blobs or cells accurately, we need to perform background subtraction so that the cells are highlighted. In this setup it suffices to use static background subtraction. However, techniques like mean filter, running Gaussian average, background mixture models, etc., may also be successfully applied. In this framework, we use static background subtraction and thresholding to separate the cell blobs from the background.

The block diagram shown in Figure 10 describes the simple steps followed in the process of background subtraction.

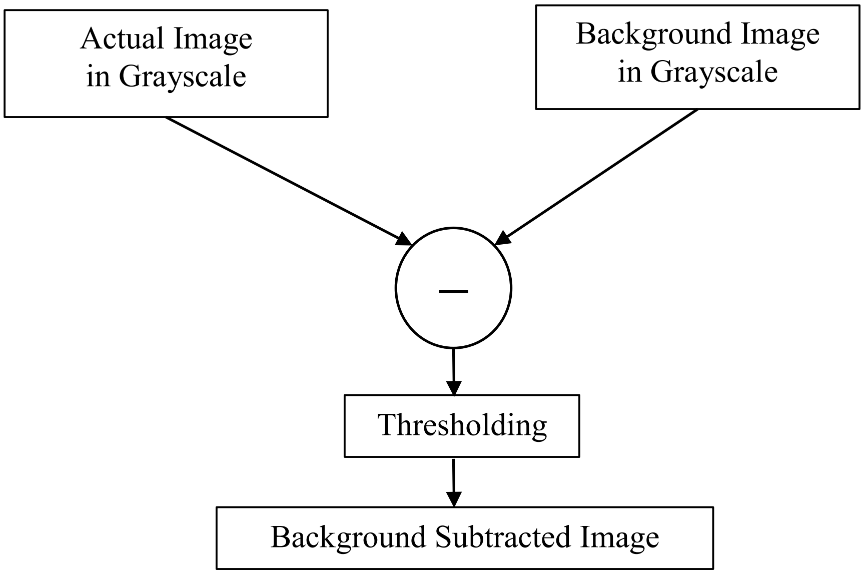

Actual Image in Grayscale
Background Image in Grayscale
–
Thresholding
Background Subtracted Image

Figure 10. Block diagram for static background subtraction.

This simple background subtraction process first converts both the color image and image of background to grayscale with pixel values in the range [0, 1]. The actual image is then subtracted from the background image to output an intermediate image with removed background and isolated blobs of cells. This image is further processed through static thresholding, in which all values below a threshold value are set to zero. We thus obtain an image with its background removed.

Let $B[x, y, T]$ be the grayscale background image obtained from the video at a fixed time instant T and $I[x, y, t]$ be the frame at a time instant t and (x, y) represent 2-D pixel coordinates. The background subtracted image at time instant t will be given by:

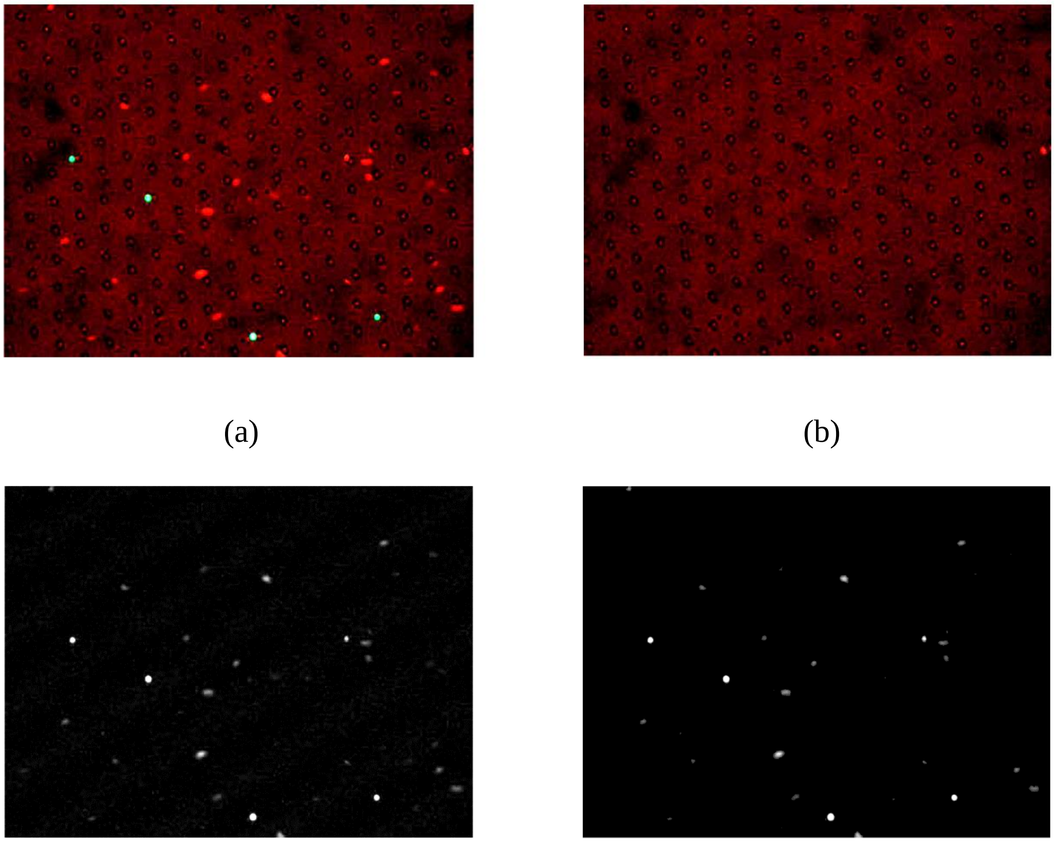
$$I_F[x, y, t] = I[x, y, t] - B[x, y, T] \tag{13}$$

The notation I_F with subscript F denotes foreground image. Now, we apply a threshold value of c on the pixel intensity to give the isolated foreground image as:

$$I_F[x, y, t] > c \tag{14}$$

The pixel intensities below this threshold will be set to a value of zero.

Figure 11 below shows the original frame in RGB and the images that appear in the process of background subtraction. These images are obtained from the raw videos.



(a)

(b)

(c)

(d)

Figure 11. The images shown above are increased in intensity for clarity. (a) Original image. (b) Static background image. (c) Isolated foreground image (in grayscale). (d) Thresholded foreground image.

An inherent disadvantage of this approach is that setting thresholds can remove objects that can actually be counted as cell blobs. This disadvantage is a concern for any technique that uses static thresholds. However, we set thresholds that are enough to detect sufficient number of cells for our analysis. The output image in Figure 10(d) is used as an input to the blob detection. The next section will describe the blob detection techniques used in this framework.

4.3 Cell Detection Techniques

After the isolation of cell blobs in a frame through background subtraction, we move towards detecting them and finding the locations of those blobs in the image. Blob detection is an important step to detect the cells. It helps in detecting the cells that are close to a required shape from the background subtracted image. At first we directly use the difference of Gaussian (DoG) approach to identify the blobs and then use connected component analysis on the filtered image with 8-connectivity to locate the position of centroids and draw bounding boxes. The block diagram in Figure 12 below explains the blob detection process through DoG.

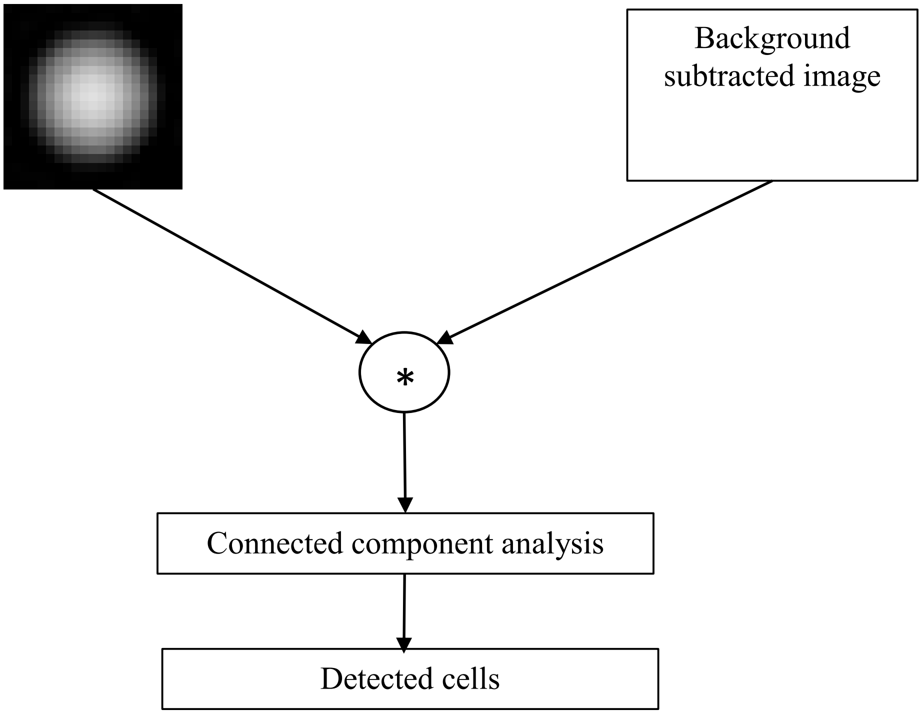


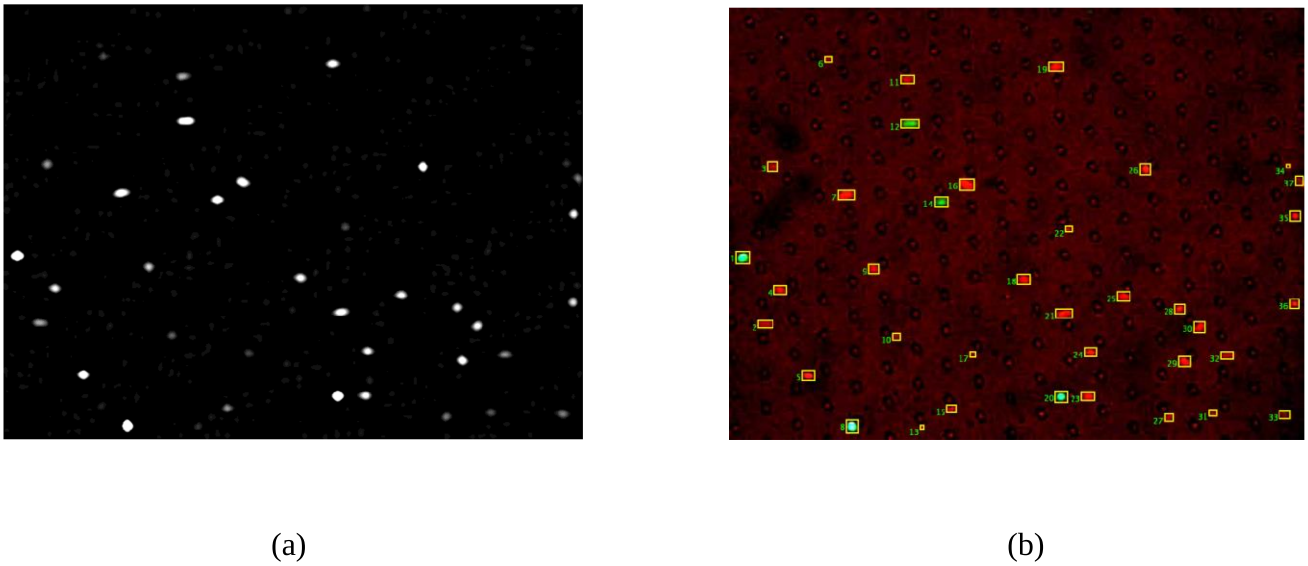
Figure 12. Block diagram showing blob detection using the Difference of Gaussian approach and connected component analysis.

In the block diagram above, the difference of Gaussian filter is convolved with the image in which the blobs are supposed to be detected or highlighted. The ‘*’ symbol stands for convolution in the block diagram and in equation (15). In case of grayscale images (as in our case), two Gaussian filters with different standard deviations are subtracted to create a DoG filter. We can apply the DoG filter to a background subtracted image without the removal by threshold step (to remove high frequencies). However, we would still need to give thresholds to convert it to appropriate binary image for connected component analysis.

For a grayscale foreground image $I_F[x,y,t]$ at time frame t the difference of Gaussian blob detection can mathematically be represented as follows:

$$I_b[x,y,t] = (G_{\sigma 1} - G_{\sigma 2}) * I_F[x,y,t] \tag{15}$$

where $G_\sigma = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x^2+y^2)/2\sigma^2}$ and $(G_{\sigma 1} - G_{\sigma 2})$ is the difference of Gaussian function. The result $I_b[x,y,t]$ will be a blurred image with the highlighted cells and ideal background value of zero. We use a filter size of 21x21 pixels with standard deviations $\sigma 1$ and $\sigma 2$ with values 5 and 10, respectively. These values were obtained empirically through multiple rounds of experimentation. Now, we use connected component analysis [14] to find the blob centroids, draw fit bounding boxes and extract other possible shape-size features from a single frame. Figure 13 below shows the output from the difference of Gaussian filter and the detected blobs with bounding boxes in the original image frame.



(a)

(b)

Figure 13. (a) Blurred image with highlighted blobs obtained through DoG filter. (b) Detected blobs shown in the image bounded by boxes.

We could have applied the connected component analysis over the background subtracted image directly. However, it is bound to catch erroneous pixels as required cell blobs. Erroneous pixels occur due to thresholds and are sometimes hard to avoid in techniques that use static thresholds for pixel exclusion. This is the reason DoG is helpful in detecting blobs or cells of the required shape and size. We fix the dimensions of the DoG filter used in this framework that leads to successful detection of cells.

As an add-on to the blob detection part, we also tried the k-means algorithm with binary quantization before DoG in hopes of improving the detection process. The k-means clustering analysis separates the cell blobs from the background or even two classes of cells based on pixel intensity. We can directly assign a value of zero to the background and the cluster centroid values to the cell blobs. This process will make the detection more precise. It will also help in removing erroneous pixels after the background process. However, it also removed much of the cell pixel information and we did not take it up further in the framework.

4.4 Multiple Cell Tracking

In this section, we discuss techniques used for multiple cell tracking from the given videos. For achieving a good tracking result, we implement a two level algorithm for tracking. In this framework, we did not handle certain cases such as occlusions and reappearances of cells in the frames. Thus, there is room for improvements in our tracking system. Nevertheless, the following tracker works well for the given data and suffices for achieving our current objectives.

The tracker works by detecting blob centroids in every frame and then utilizes optical flow (KLT tracker) [15] for locating the object position using feature matching in the next frame. We detect blobs in every frame using our blob detector and then find the matching centroids of those blobs in the next frame through KLT. We perform this for consecutive frames. To associate the detected blobs into tracks we use the Hungarian algorithm [16]. This cascade of techniques gives us a good and reliable tracking performance. It is necessary that the camera be stationary and this is in fact the case in our micro-fluidic device system. Further, we also use the Hungarian algorithm for detecting the frame of entry of new cells. Figure 14 below shows a block diagram of the KLT + Hungarian linker algorithm for multiple object tracking.

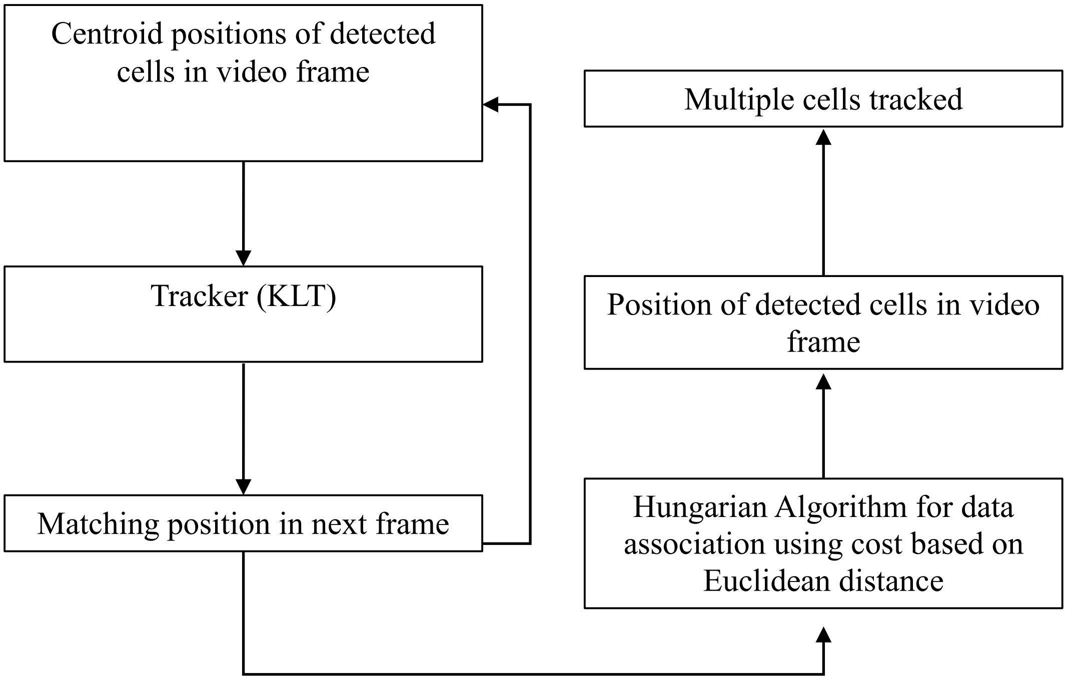

Centroid positions of detected cells in video frame
Tracker (KLT)
Matching position in next frame
Multiple cells tracked
Position of detected cells in video frame
Hungarian Algorithm for data association using cost based on Euclidean distance

Figure 14. Block diagram showing KLT+Hungarian linker for multiple object tracking.

The Kanade-Lucas-Tomasi (KLT) feature tracker works through template matching. It makes use of spatial intensity of the selected points to find the best match in the next frame. The input to the tracker is the centroid of the blobs in the current frame. The main assumption in the KLT tracker is that neighboring groups of pixels have similar intensities and relatively small movements. This assumption of similar intensities can be mathematically given as:

$$i(x, y, t) = i(x + u, y + v, t + 1) \tag{17}$$

where $i(x, y, t)$ is the intensity of a pixel at location (x, y) and at time t. The same pixel is displaced by (u, v) in the next frame which is at a time instance $t + 1$. The Taylor series expansion of the right hand side in the above equation gives:

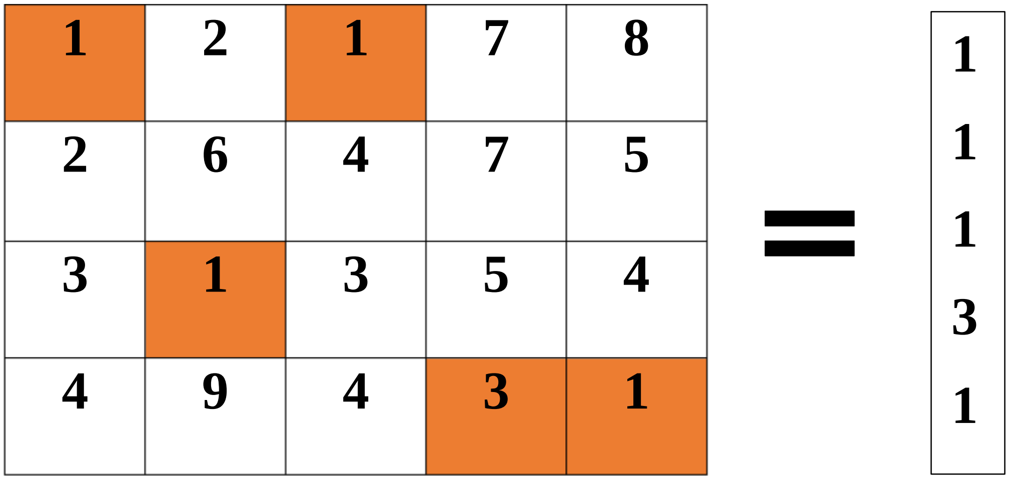
$$i_x u + i_y v + i_t \approx 0 \tag{18}$$

where i_x and i_y are image derivatives along horizontal and vertical axis respectively. i_t is the difference of consecutive image frames. An optimal value of (u, v) is arrived at by the following matrix notation under the constraints of existence of matrix inversion:

$$\begin{pmatrix} \sum i_x i_x & \sum i_x i_y \\ \sum i_x i_y & \sum i_y i_y \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = -\begin{pmatrix} \sum i_x i_t \\ \sum i_y i_t \end{pmatrix} \tag{19}$$

After the prediction of points in the next frame, we use the Hungarian algorithm as mentioned before. The Hungarian algorithm efficiently solves the track assignment and association problem. It also helps to identify the frames in which new cells appear. This helps in associating labels or IDs to each cell that is being detected and tracked. The Hungarian algorithm can be implemented by dynamic programming in $O(n^3)$ time and is fast enough for real time applications. The Hungarian algorithm aims at minimizing a cost, which in our case is the Euclidean distance.

In a matrix of distance values, we compare every element in the first column to find the Euclidean distance to the next column and the pair with lowest Euclidean distance are connected. This process is performed for consecutive columns until required and a chain of close relatives based on Euclidean distance is made. An example that explains the algorithm is shown in Figure 15.



$$\begin{bmatrix} 1 & 2 & 1 & 7 & 8 \\ 2 & 6 & 4 & 7 & 5 \\ 3 & 1 & 3 & 5 & 4 \\ 4 & 9 & 4 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 3 \\ 1 \end{bmatrix}$$

Figure 15. Example table showing distance values. The distances highlighted in red are the closest distances in consecutive frames in first iteration. They are connected to obtain a single matrix of associated distances.

Figure 15 explains the working of the Hungarian algorithm. In our case, the columns of the matrix are the frames and the rows are detected cell centroids. Distance between the centroids in first value of the first column to all the values of the second column is calculated and the pair with lowest distance are connected. This is done repeatedly till the cell blob disappears. Once the track is complete, it is removed from the distance matrix and stored in a connected track-matrix where each column corresponds to the track/path followed by a single cell. The disappearance of the blobs is handled by a threshold value of distance. If the minimum distance between a pair of centroids is more than the threshold, the track is closed and assigned to the cell.

The frames in which the cell blobs appear is calculated based on the size of the distance matrix and the track-matrix. Their sizes change dynamically as and when a new track for a given cell is made. The cell ID or labels are assigned through the column number of the track matrix which holds the tracks. The cell IDs are displayed on the image using a text inserter. The text inserter uses the top left corner of the cell bounding box as the pivot and we can insert the cell ID near it. In order to carry forward the ID of a cell in the whole path, we again use the shortest path algorithm for matching the ID to the corresponding cell in the next frame.

Figure 16(a) shows an image of paths followed by the cells. It also shows labelled cells and their propagation in the next frame using Hungarian algorithm for data association. It is seen that appearances of new blobs are also handled appropriately through their frame of entry. The cells in the recorded video data move from left to right edge of image frame.

A problem that arises in this algorithm is that of change in ID of the same cell. This happens due to: 1) absence of occlusion handling; 2) blob detector fails to detect the cell in a frame; or 3) the cell moves out of the image area and reappears after few frames. Out of these, 2) has been handled in this framework and the remaining two issues have been left for future work on improvements. Nevertheless, the tracker serves the purpose of this framework.

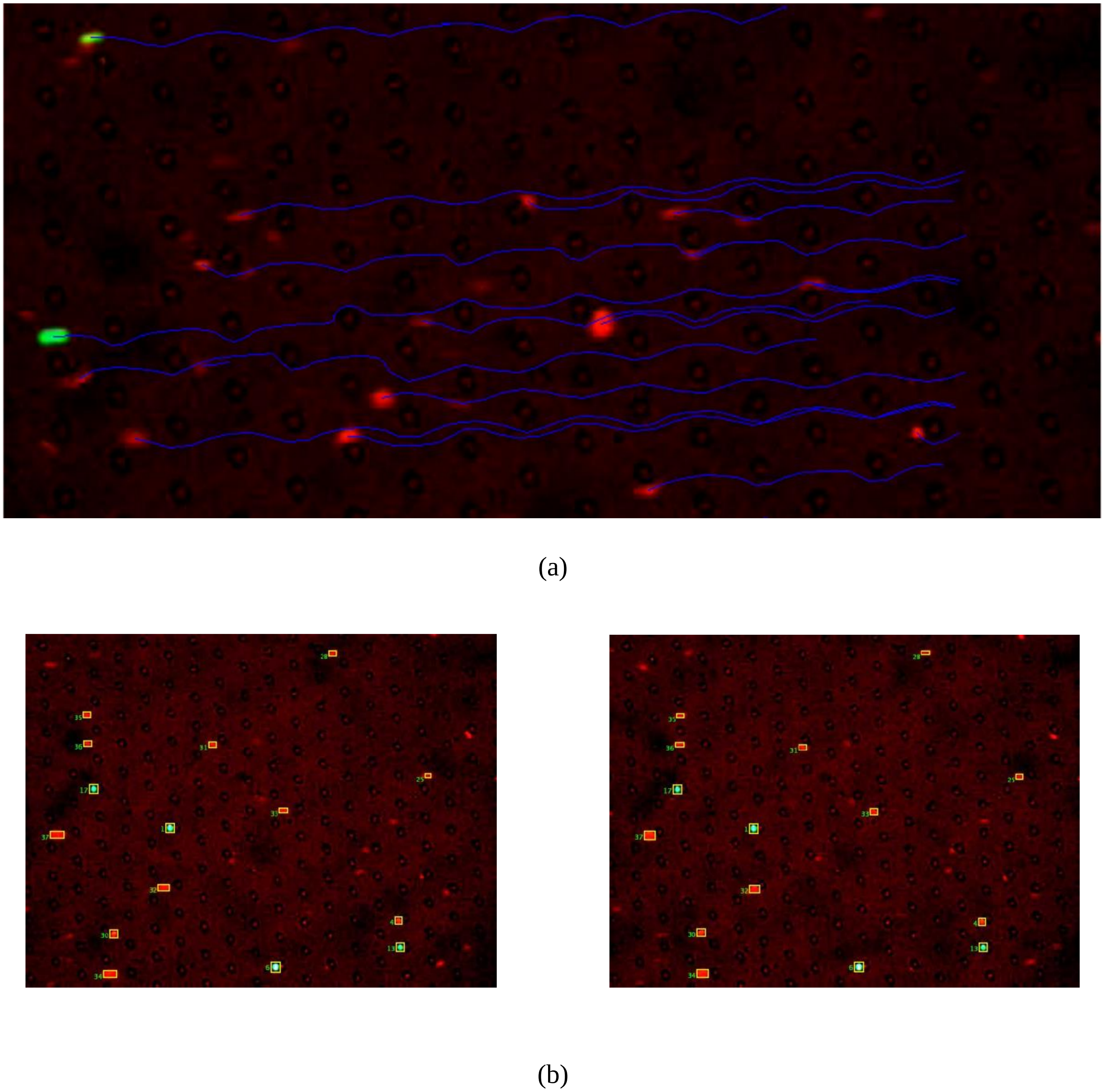


Figure 16. (a) Shows the paths (in blue) followed by the detected cells. (b) Shows two consecutive frames and a successful propagation of cell ID through tracking.

4.5 Automatic Class Labeling & Feature Extraction

One of the most important aspects of machine learning is data collection and class assignments to create a data set (or design matrix). Usually, class labeling (e.g., live vs. dead) of images is done manually and is in fact a time consuming process. We perform supervised learning only after we obtain a class labeled data set. However, the process of manual class labeling can be eliminated by knowing certain conditions that will allow us to automatically collect data. In our setup, dielectrophoresis is the phenomenon that helps us know the correct class of a cell. When the field is turned on, PC-3+ cells are attracted to posts, but the dead PC-3− cells are not. Thus we perform the class labeling during the time period when the electric field is ON. When the electric field is turned OFF, the class labels are still associated with the cells.

After associating class labels (1 or 0) to the cells, we perform feature extraction. The cell features are extracted in terms of Histogram of Oriented Gradients and based on ellipse fitting (elliptic features). The elliptic features are explained in the next chapter. In this section we describe HOG features and how they could be useful. The feature extraction process is initiated when the electric field is switched ON. It is performed on all cells that appeared in the frame when the field is ON. The processed and labelled feature vectors are now used as a data set for the classifiers. This data set can be used for supervised learning and will help us identify the different populations even in the absence of electric field. Through this approach our system serves as a great means for automating labelled training data collection.

The automatic labeling is performed by using a threshold value which can be the number of frames (or time) for which a cell has very little change in distance in consecutive frames. We use the number of frames as the threshold in our implementation. Mathematically, in our data, for a threshold frame value of f and distance travelled by cell between consecutive frames d, the class labeling can be defined as:

$$class(cell\ ID) = \begin{cases} 1, & d \leq 3.4, Frames > f \\ 0, & d > 3.4, Frames > f \end{cases} \tag{20}$$

The above conditions hold only when electric field is turned ON. The value of f is set to 30 frames.

Figure 17 show class assignments for the two populations considered here. A class of ‘1’ (PC-3+) is assigned to the cells attracted to the post and class of ‘0’ (PC-3−) is assigned to the cells not attracted to posts.

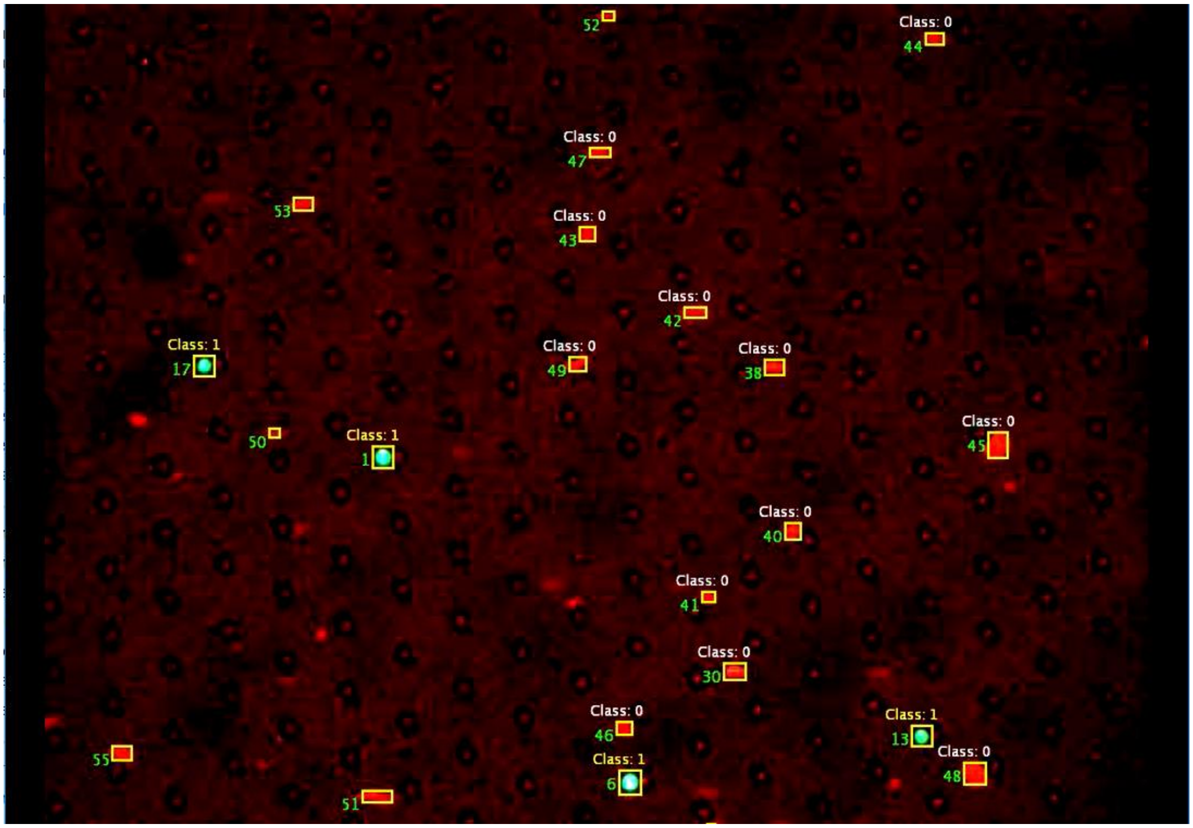


Figure 17. Image showing associated class labels. The cells attracted to posts are labelled as class 1 and the rest are labelled as class 0. The cells with no class assigned yet are newly emerged blobs and will soon be assigned a class.

Feature extraction is performed as soon as a class is assigned to a cell. Different types of features that can be extracted from the cell include color features, temporal shape changes, geometric features such as area, axis ratio, compactness and solidity, texture features and statistical features such as standard deviation or correlation between two features. Other features that can be extracted from images also include image moments (or Hu moments), HOG [19], shape context, SIFT, etc. [8, 11, 12, 17, 18].

In this work we focus mainly on features that would help us analyze and characterize cells based on shape, texture and statistics. The features that we extract from the cells are as follows:

- *Shape & texture features*: Histogram of Oriented Gradients [19] (HOG).
- *Geometric & elliptic features*: Area of ellipse fit, solidity, compactness, eccentricity, area of cell, spheroid volume, orientation, major axis changes, axis ratio.
- *Kinematic features*: Velocity (or speed – both are synonymous between two consecutive frames), acceleration.
- *Statistical features*: Correlation
- *Mechanical (intrinsic) properties:* Elasticity.

The above features are all indicative of phenotype of cells and can ascertain differences between two populations or sub-populations. The shape and texture features along with elliptic features are very useful as training data for classification algorithms such as support vector machine. The kinematic features and elliptic features are useful in calculating the mechanical properties such as the elasticity. Statistical features are useful for setting a benchmark for comparing two or more types of cells. For example, a high degree of correlation between features of two cells can indicate that they are of the same phenotype.

In this section we describe the Histogram of Oriented Gradient (HOG) features and show how they are applicable to our cell data. We dedicate a separate chapter to discuss the elliptic and statistical features.

Histogram of Oriented Gradients:

The Histogram of Oriented Gradients (or HOG) is a feature descriptor that counts the occurrences of gradient orientation in localized portions of an image. It is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy and invariance to illumination and shadowing. It was primarily developed for human detection and it surpassed the traditional wavelet based techniques. The HOG descriptors can be computed from a local point in an image or for a localized portion of an image, where the feature length (or dimensions) is dependent upon the image size, HOG cell (not to be confused with PC-3 cell) box sizes and the degree of overlap between HOG cells. The main idea behind the HOG descriptor is that local object appearance and shape in an image can be described by the distribution of intensity gradients or edge directions [19].

For computing the HOG descriptors, the image is divided into small connected regions called HOG cells and a histogram of gradient directions is calculated for each pixel in within the cell. This is a good way to characterize two categories of cells, say cancerous and non-cancerous. As discussed earlier, there have been numerous studies [3, 20] that explain many differences in cell morphology of non-cancerous and malignant cells. One of the key differences in the cell morphology of non-cancerous and cancerous cells is the cell membrane shape, cytoplasm color differences and irregular size, shape and arrangement of nucleus and organelles. These characteristic are prominent when viewed through a microscope or a recoded microscope video. HOG descriptors are a great way to describe such shape and texture characteristics and differences from an image of such cells. Resized images of PC-3+ and PC-3− images are shown in Figure 18. Some very recent studies apply HOG features for medical image analysis on magnified cancer cells [21, 22]. Work in [22] is done on magnified images of PC-3 cells which is also relevant to our data set. Our attempt is to record these features from a video data with low resolution and low magnification and, as we will see in later chapters, we are also successful in differentiating cells based on a concatenation of HOG and elliptic features. It is, however, important to mention that quality of features and amount of data collected is dependent on the image resolution, magnification and frame-rate.

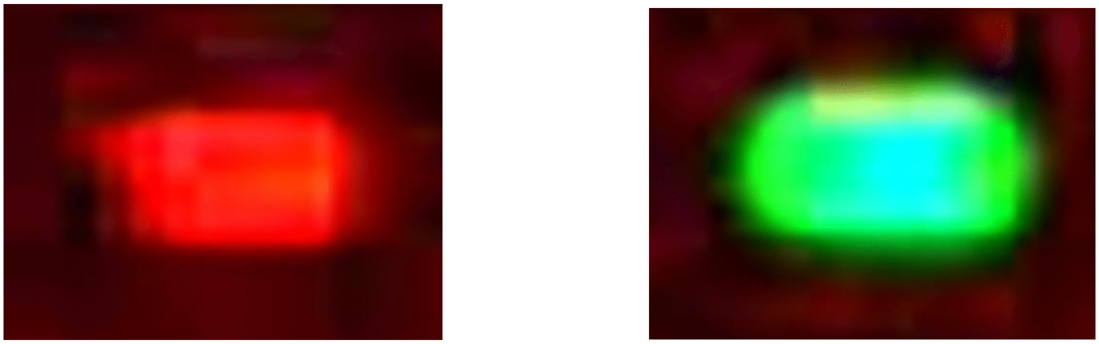


Figure 18. Image on the right shows a PC-3+ (malignant) cell and on the left is a PC-3− (dead). Both are stained and the shape differences are fairly visible in fluid flow (images have been resized for clarity).

As the PC-3 cells are of low resolution and small size in our images, it is necessary that we extract small scale features. Thus it is very important that we set the HOG parameters according to our requirements. The main parameters to consider while computing HOG features are number of orientation bins and the HOG descriptor block (R-HOG or C-HOG) which further includes number of cells per block, number of pixels per cell and number of channels per cell histogram. Other parameters include the image size, block overlap and orientation sign. All these parameters decide the HOG descriptor dimensions. We use the rectangular R-HOG blocks for feature extraction on our data. We use four HOG cell sizes, 1x1, 2x2, 3x3 and 4x4 for obtaining small scale features from cells. The block size is set to 2x2 with half overlap between adjacent blocks. Orientations of gradients are unsigned and number of bins is set to 9.

We extract grayscale patches from the images where the PC-3 cells are located through segmentation and tracking and then compute HOG features on these local patches (images of cell patches are shown in Figure 19). The HOG feature length (or dimension) is dependent on image size and different image sizes give different HOG feature lengths. Thus we have two options to maintain uniformity in HOG feature length: 1) Resize all the PC3 cell images to the same value; or 2) pad zeros on the images such that all are of same size. We resize all the cell images to a value of 14x15 – which was calculated as the mean size of all cell boxes (Figure 20). Although resizing images causes loss of image information, most of the key shape features are still preserved. Another disadvantage is that of the images being already compressed as JPEG format. This affects the quality of features extracted because of loss of information. We also perform another level of thresholding in the local cell patches to remove artifacts before computing HOG. We test four HOG cell sizes in our experiments. A scale of 1x1 and 2x2 had large feature length. This causes an increase in space and time complexity of the algorithm. HOG cells of size 4x4 have smaller feature length and show relatively better algorithmic performance. However, 4x4 HOG cells start losing small-scale features required to describe a PC-3 cell in the given data. HOG cell visualizations for each scale is shown in Figure 21.

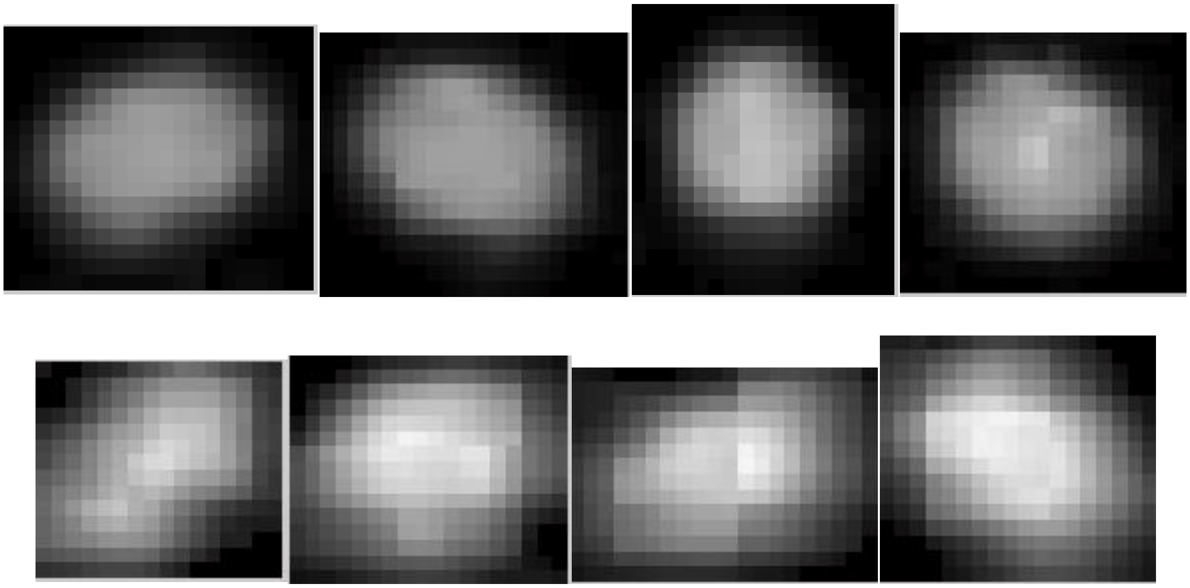


Figure 19. Figure showing cell patches. The top four images are of PC-3 malignant cells and bottom four are dead PC-3 (images are adjusted for brightness and clarity).

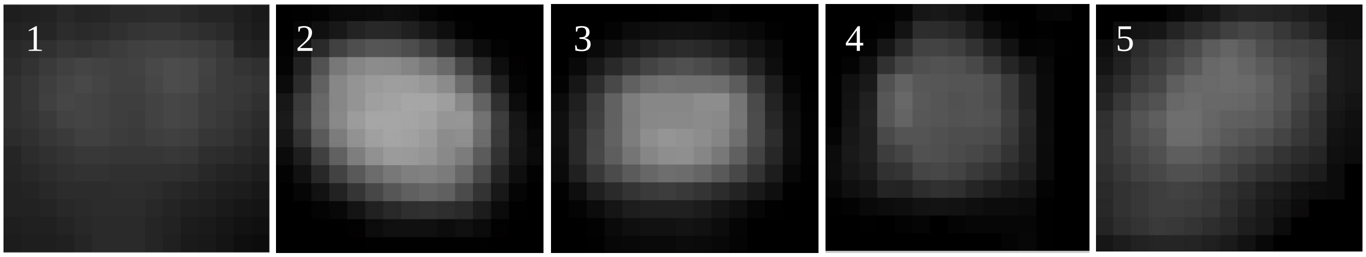


Figure 20. Figure showing cell image patches resized to 14x15. Images numbered 1, 4 and 5 are dead PC-3 cells. Images numbered 2 and 3 are malignant PC-3 cells (images are adjusted for clarity).

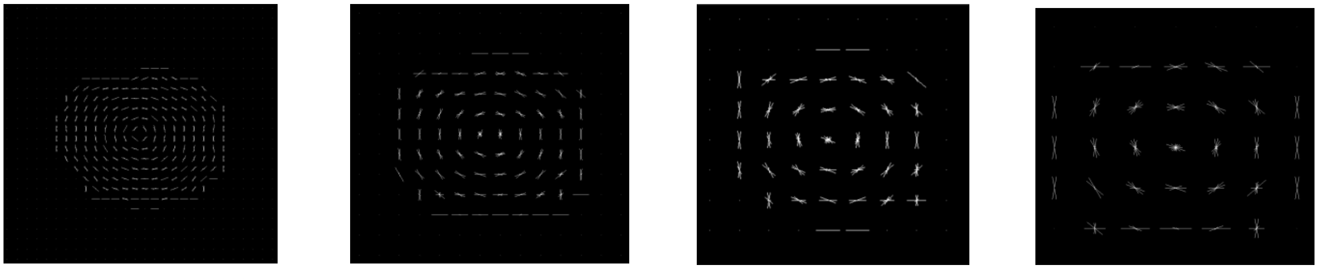


Figure 21. Histogram visualizations for a single patch of a malignant cell at different scales. From left to right– HOG for cell sizes of 1x1, 2x2, 3x3 and 4x4 respectively.

4.6 Classification Algorithms

For the task of classification, two main classifiers have been trained using the labelled feature data. The two classifiers that we use are logistic regression and a Support Vector Machine. We discuss these two classifiers in this section. We discuss the characteristic evaluation measures, cross validation, train and test errors in a later chapter.

Logistic Regression:

Logistic regression is a probabilistic classifier that separates data by a linear decision boundary. The dependent variable is the class label assigned to a data point (or a feature vector) and is thus a categorical variable. In our case this categorical variable would imply cancerous vs. non-cancerous or live vs. dead cells (binary class). The conditional probability of a class label assignment is given by a sigmoid function shown in Figure 22.

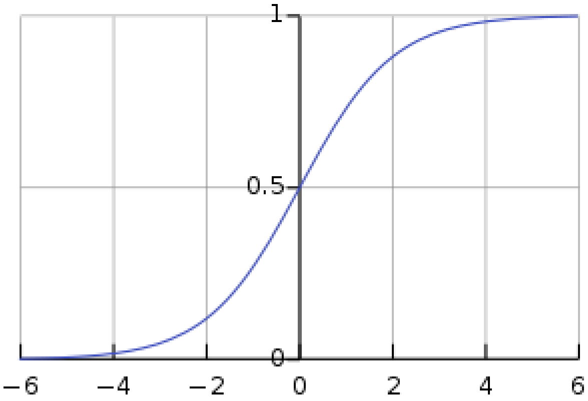


Figure 22. The sigmoid function. Source: Wikipedia (fair use).

The sigmoid function is given by:

$$f(x) = \frac{1}{1 + e^{-x}} \qquad (21)$$

For a binary classification problem, the conditional probability $P(Y_i \mid \boldsymbol{X}_i, \boldsymbol{w}, b)$ of categorical variable Y_i belonging to class 1, where $Y_i \epsilon \{0, 1\}$, is given by:

$$P(Y_i | \boldsymbol{X}_i, \boldsymbol{w}, b) = \frac{1}{1 + e^{-(\boldsymbol{w}^T \boldsymbol{X}_i + b)}} \qquad (22)$$

where $i = \{1, 2, \dots, n\}$, n is the number of feature vectors, $\boldsymbol{X}_i$ is a single feature vector of size m x 1 and $\boldsymbol{w} = [w_1, w_2, \dots, w_m]^T$ is the weight assigned to each feature in $\boldsymbol{X}_i$. The objective of logistic regression is to find the optimal set of weights or parameters $\boldsymbol{w}$ that separate the two classes through a decision boundary. Logistic regression can be solved in terms of linear algebra or as a probabilistic model where we compute the cost through maximum likelihood estimation or through log-likelihood. The class Y_i is assigned based on the sign of an optimized value of $\boldsymbol{w}^T\boldsymbol{X}_i + b$, where T denotes transpose and b is a constant or bias term.

The simplified cost function (log-likelihood) along with regularization term λ, that we need to minimize, can be represented in matrix expression as shown in (23).

$$J(\boldsymbol{w}) = -\frac{1}{n}\sum_{i=1}^{n}\left[Y_i \log\left(\frac{1}{1+e^{-(\boldsymbol{w}^T\boldsymbol{X}_i+b)}}\right) + (1-Y_i)\log\left(1-\frac{1}{1+e^{-(\boldsymbol{w}^T\boldsymbol{X}_i+b)}}\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{m}\boldsymbol{w}_j \qquad (23)$$

When performing optimization, we can also solve the cost function in matrix form in code by having $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_n]$ and $\boldsymbol{Y} = [Y_1, Y_2, \dots, Y_n]^T$. $\boldsymbol{X}$ is of size m x n, $\boldsymbol{Y}$ is of size n x 1, $\boldsymbol{w}$ is of size m x 1, where m is the length of feature vector $\boldsymbol{X}_i$. The corresponding weight update will be given by equation (24).

$$\boldsymbol{w}_{new} = \boldsymbol{w}_{old} - \alpha\left(\sum_{i=1}^{n}\boldsymbol{X}_i\left[Y_i - \log\left(\frac{1}{1+e^{-(\boldsymbol{w}_{old}^T\boldsymbol{X}_i+b)}}\right)\right] + \lambda\boldsymbol{w}_{old}\right) \qquad (24)$$

where α is the learning rate.

We use gradient descent to reduce the cost $J(\boldsymbol{w})$ above and obtain an optimal set of weights. For a test feature vector $\boldsymbol{X}_{test}$ the prediction labels will be given as:

$$Y_{test} = \begin{cases} 1, & \boldsymbol{w}_{new}^T\boldsymbol{X}_{test} \geq 0 \\ 0, & \boldsymbol{w}_{new}^T\boldsymbol{X}_{test} < 0 \end{cases} \qquad (25)$$

Logistic regression is a linear classifier because the decision boundary is linear in $\boldsymbol{w}$. In our framework the training data (or design matrix) $\boldsymbol{X}$ is the set of HOG features concatenated with some of the elliptic features (explained in a later chapter). $\boldsymbol{Y}$ is the column vector of class labels assigned during automatic labeling. The analysis and results of logistic regression on the data set is discussed in a later chapter.

Support Vector Machine:

Support Vector Machine [23] (or SVM) is a non-probabilistic classifier for supervised learning. Solving the cost function of an SVM is a convex optimization problem and usually requires a quadratic solver package or other algorithms like the sequential minimal optimization (SMO). SVMs can also be used for regression and unsupervised clustering problems, where it is called as the support vector clustering. The SVM can be used as a linear as well as a non-linear classifier. SVM utilizes the kernel trick to behave as a non-linear classifier, where the kernel maps the data to a higher dimension feature space, forms a decision boundary (which is a hyper-plane) and mapped back to the original feature space.

An important idea and objective behind the support vector machine is to separate the data by the largest possible margin (or gap). Thus it is also called as the large margin classifier. Figure 23 shows an example of data separation by the decision boundary made by an SVM. The SVM is divided into hard margin and soft margin which are different in terms of the cost function. The convex optimization problem of an SVM is divided into primal and dual forms.

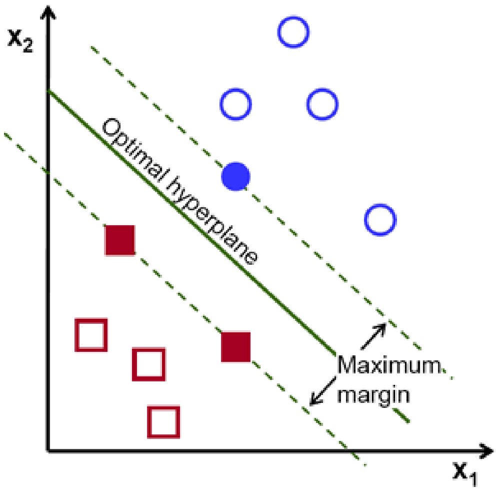


Figure 23. Maximum margin data separation by an SVM. Source: OpenCV 2.4.13.3 Documentation (fair use).

In a hard-margin SVM, the pair of parallel hyper planes that separate the data set can be written as:

$$\begin{aligned} P_1 &: \boldsymbol{w}^T \boldsymbol{X}_i - b = 1 \\ P_{-1} &: \boldsymbol{w}^T \boldsymbol{X}_i - b = -1 \end{aligned} \tag{26}$$

where $\boldsymbol{w}$, $\boldsymbol{X}_i$ and b mean the same as in previous section. Data points that lie on P_1 and P_{-1} are called the support vectors. Both these planes are separated by a distance margin given by:

$$D = \frac{2}{\boldsymbol{w}^T \boldsymbol{w}} \tag{27}$$

Maximizing the distance D in (27) is the objective of an SVM. Equation (27) can also be maximized by minimizing its denominator because its numerator is a constant. Thus we minimize the inverse of this expression which is given by $\min(\frac{\boldsymbol{w}^T\boldsymbol{w}}{2})$.

Now, if the label for a data point is given as Y_i which takes values $\{1, -1\}$, the separating hyper-plane can be written as shown in (28).

$$Y_i(\boldsymbol{w}^T\boldsymbol{X}_i - b) \geq 1 \tag{28}$$

This becomes the constraint for the SVM under which we minimize the expression discussed previously. The solution of $\boldsymbol{w}$ and b is obtained after solving the problem will become the model for classification. The class label will thus be dependent upon the sign of the value $(\boldsymbol{w}^T\boldsymbol{X}_{test} - b)$.

In a soft-margin SVM, we consider the cases when the data is not linearly separable and we introduce hinge loss. This introduces a slack variable instead. The hinge loss is given by: $\zeta_i = \max[0{,}1 - Y_i(\boldsymbol{w}^T\boldsymbol{X}_i - b)]$.

The new cost function thus becomes:

$$L = \frac{1}{n}\sum_{i=1}^{n} \zeta_{\mathrm{i}} + \frac{\lambda}{2}||\boldsymbol{w}||^2 \tag{29}$$

where λ acts as a regularization term. For a sufficiently small λ the soft-margin SVM will behave as a hard-margin classifier.

In the Primal SVM form, the constraint to the above cost minimization is given by

$$\begin{gathered} Y_i(\boldsymbol{w}^T\boldsymbol{X}_i - b) \geq 1 - \zeta_i \\ \zeta_i \geq 0, \forall\ i\ \epsilon\{1, 2, 3, \dots, n\} \end{gathered} \tag{30}$$

where ζ_i is the smallest non-negative number satisfying the above constraints. The expression $\boldsymbol{w}^T\boldsymbol{X}_i$ can also be represented as a dot product given by: $\overline{\boldsymbol{w}} \cdot \overline{\boldsymbol{X}}_i$. The value $\overline{\boldsymbol{X}}_i$ can be replaced by a kernel function to map it to a higher order or different feature space.

The SVM Dual form can be arrived at by differentiating the primal form loss with respect to $\overline{\boldsymbol{w}}$. We introduce a Lagrangian multiplier α_i to solve the primal quadratic optimization problem with the constraints. We see that the solution for $\overline{\boldsymbol{w}}$ can always be a linear combination of train data as shown in (31).

$$\overline{\boldsymbol{w}} = \sum_{i=1}^{n} \alpha_i Y_i \overline{\boldsymbol{X}}_i \tag{31}$$

Replacing this value in the cost function above changes it as a function of α_i, $\forall i\epsilon\{1, 2, \dots, n\}$, and as a maximization problem,

$$L(\alpha_1, \alpha_2, \dots, \alpha_n) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} Y_i \alpha_i (\overline{\boldsymbol{X}}_i \cdot \overline{\boldsymbol{X}}_j) Y_j \alpha_j \tag{32}$$

subject to the following constraints:

$$\sum_{i=1}^{n} \alpha_i Y_i = 0$$
$$0 \leq \alpha_i \leq \frac{1}{2n\lambda} \tag{33}$$

The above conditions also form the Karush-Kuhn-Tucker (KKT) conditions for defining the support vectors.

The offset b can be recovered by (34).

$$b = Y_i - \overline{\boldsymbol{w}} \cdot \overline{\boldsymbol{X}}_i \tag{34}$$

Now, we can introduce non linearity in an SVM by replacing the dot product with a kernel function. This is given in equation (35).

$$k(\overline{\boldsymbol{X}}_i, \overline{\boldsymbol{X}}_j) = \phi(\overline{\boldsymbol{X}}_i) \cdot \phi(\overline{\boldsymbol{X}}_j) \tag{35}$$

All expressions for $\overline{\boldsymbol{w}}$ in (31) and loss functions change accordingly. The kernels are usually polynomial, radial basis function (RBF) or hyperbolic tangent. Any new test point can be computed by the expression in equation (36).

$$class(\boldsymbol{X}_{test}) = sign\left(\left[\sum_{i=1}^{n} \alpha_i Y_i k(\overline{\boldsymbol{X}}_i, \boldsymbol{X}_{test})\right] + b\right) \tag{36}$$

In our experiments we use the same test features as we used in logistic regression. All results and discussion for the given data set using support vector machine are done in a later chapter.

Chapter 5

Morphological features

5.1 Introduction

In this chapter we discuss other features (such as geometric, elliptic, kinematic and statistical features) used to identify cells and their importance in differentiating morphological properties among two classes of cells. Works such as [12] show that blood disorders are identified through visual inspection of blood samples under a microscope. They show the importance of image processing and how it can be useful for speeding up the process of identification of a disease at an early stage so that treatment becomes easier. Their work primarily uses leukemia cell images and examine changes in texture, geometry, color and statistics. Further, they make use of reinforcement learning to automate the process of classifying these cells. Another work [17] makes use of histological image analysis to identify prostate cancer where they classify a tissue image into tumor or non-tumor class. They make use of texture, color and other morphometric features to identify the class of cells. Finally, they use Gaussian k-nearest neighbor and SVM for classification. They make use of fractal dimension and code features along with color histograms and wavelet transforms as a concatenated feature set. Work shown in [29] make use of extensive feature sets on microscopic biopsy images to classify cancer. They make use of magnified cell and tissue images to extract different types of features and make use of the standard classifiers to distinguish the data.

A recent work [11] makes use of high speed bright field photography to calculate deformity in cells under deformability flow-cytometry device. They achieve high throughput characterization of 20 different cell motions and morphology derived parameters. These parameters are extracted from videos and are mainly the size, deformity (in terms of major/minor axis change), deformation kinetics and morphology. Further, they make use of support vector machine to classify these cellular features from pluripotent stem cells, mesenchymal stem cells, neural stem cells, and their derivatives. Other works such as [30] use segmentation and tracking to examine kinematics and shape-size features to differentiate cells through Cartesian genetic program.

Our work is similar to the above works but a subtle combination of all. The features extracted in our work are mainly the ones that signify shape and size and are all extracted temporally. We also utilize a combination of these features and the experimental microfluidic chip design to derive models for physical quantity calculations such as the Young's Modulus of elasticity. We extract HOG features on a localized image rather than a local point/pixel location [29]. Moreover, our data is practical and inexpensive for feature extraction and classification tasks introducing bigger challenges. The data being relatively low resolution, low frame-rate and low magnification certainly affects the feature extraction and obscures some of the potentially very useful data such as shape and size of nucleus. Nevertheless, our classifiers show good performances in similar type of data.

5.2 Elliptic and Kinematic Features

This section shows some of the characteristics that have been used for classifying the cells and how they could be used for differentiating two classes of cells. The features discussed in this section are extracted temporally over every time frame are the geometric or elliptic features (axis ratio, goodness of ellipse fit, eccentricity, changes in major axis length in fluid flow and at collision, volume considering the cell as spheroid) and the kinematic features. The elliptic features are obtained through principal component analysis (PCA) of the cell image patches. Applying PCA on the image patches gives us ellipses fit on the cell blobs [25].

Physical features such as elasticity can also be deduced from these features which is another means of differentiating the cells and thus branches out the utility of our experimental setup. However, the cell elasticity can be calculated only when the cell collides with the pillars such that it forms an indentation on the cell membrane. The cells can also be differentiated based on mass, given mass density and cell volume. Other kinematic features that are also measured include velocity and acceleration that determine the force causing indentation in a cell. The HOG features and elliptic features are used by classifiers such as SVM and logistic regression for classification and are ideal for real time classification.

Principal Component Analysis (PCA):

The PC-3 cells in the given environment can be modeled in 2D as ellipses with continuously changing dimensions. We analyze phenotype of these cells through ellipse fitting on the cell blobs in images. Ellipse fitting can be achieved by more than one way. One of the methods is to use principal component analysis. Principal components through PCA are obtained in directions with the highest variance in descending order. Each principal component is linearly uncorrelated and orthogonal. Thus the number of components are dependent on the dimensions of feature space. For example, a 2D feature space will have a maximum of two principal components orthogonal to each other. Figure 24 shows an example of PCA applied on a data set. Such an application of PCA is used for fitting ellipses in image blobs.

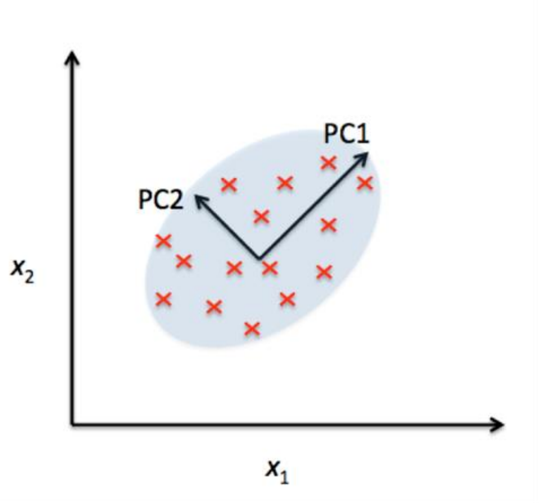


Figure 24. Image showing principal components obtained through PCA. Source: sebastianraschka.com (fair use).

For 2D data, the first component corresponds to the major axis of the ellipse and the second component corresponds to the minor axis of the ellipse. The components are obtained from the eigenvalues which are obtained from singular value decomposition of the data. PCA can also be used for dimensionality reduction. This is done by removing smallest the components or eigenvalues as they signify small variance.

Obtaining perfect ellipse fits is not an easy task [26]. However, we obtain close to perfect ellipse fits in most of the cases. Ellipse fits are dependent upon pixel intensity distribution and can catch up noise. Appropriate background subtraction is required in each cell image patch to obtain good fits. However, background subtraction based on static threshold still leads to presence of stray pixels that causes deviations from perfect fits in some cases. Stray pixels can often be removed by increasing the pixel intensity thresholds. However, this also starts to remove pixels from the cells, which causes loss of information. We set a fixed value of intensity threshold for localized background subtraction in individual cell image (grayscale) patches through experimental trials such that we get good ellipse fits around cells in most of the cases. The threshold value for local background noise removal was set to 0.07 for green cells (PC-3+) and 0.04 for red cells (PC-3−). In Figure 25 below are shown some example cases of ellipse fits that we obtain from our data. We now describe all the features that were obtained from each detected cell over the frames. All measured quantities are in pixel units. Similar measures were used by other researchers in previous works as discussed before.

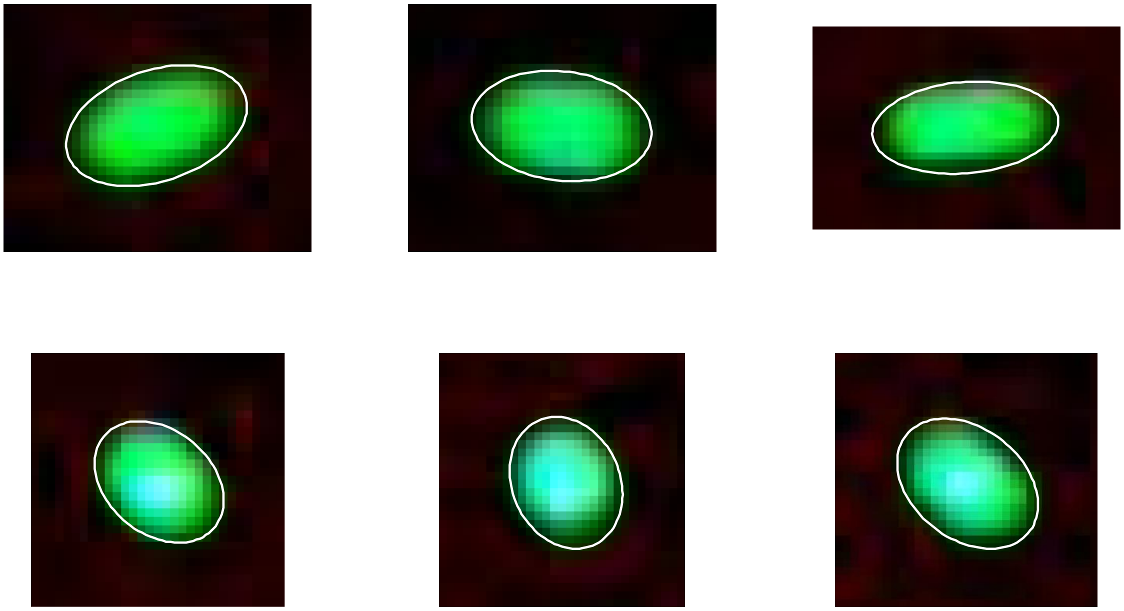


Figure 25. Images showing ellipse fits for the given cells.

a. Axis Ratio

The major and minor axes of the fit ellipses are approximately represented as major and minor axis of the spheroidal cells as described in section 2.2. The axis ratio is the ratio of major to minor axis of the ellipse (or the cell). This measure could also be represented as the elongation of the cell. The equation below represents the axis ratio of the cells.

$$Elongation\ (or\ Axis\ Ratio) = \frac{Major\ Axis}{Minor\ Axis} \tag{37}$$

Changes in this quantity in the fluid flow is a direct measure of geometric changes in the given environment. The experiments show that the axis ratio pattern differs significantly in the two populations of the cells being studied. The graphs in Figure 26 show that the patterns are different. There is a measurable trend and significant uniformity in graphs of PC-3+. However, this is not evident in the PC-3−.

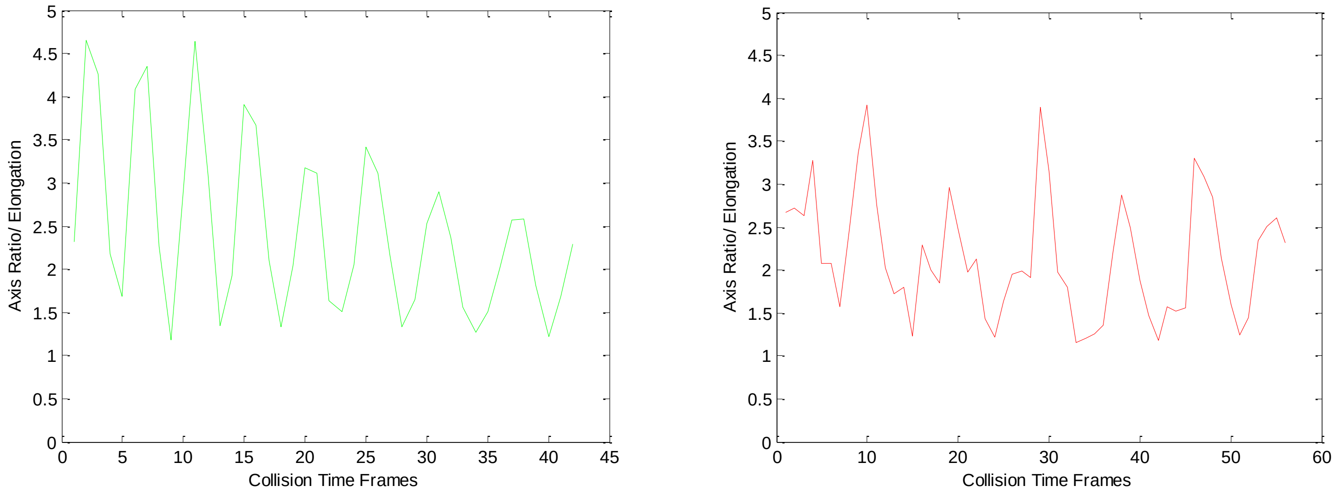


Figure 26. The graph on the left shows elongation changes for a PC-3+ (live cancerous) with cell ID of 108 and graph on the right shows the same for PC-3− (dead) with cell ID of 4 over all the frames of their presence. The collision instances occur at the local minima of the axis ratio curves.

b. Cell Membrane Compactness:

The cell membrane compactness or the goodness of ellipse fit is measured as ratio of areas of the cell with respect to the fit ellipse. It could also be represented as the relative solidity. This quantity determines how close the cell is within the ellipse and a value close to 1 corresponds to a good fit. Although a perfect fit to ellipse is not expected, the live cancerous cells have a definite/rigid shape and correspond close to ellipses in fluid flow. However, the dead cells have an undefined membrane shapes and do not have a good relative fit. The expression for compactness is given below:

$$Compactness = \frac{Actual\ Cell\ Area\ (in\ pixels)}{Fit\ Ellipse\ Area\ (in\ pixels)} \tag{38}$$

The temporal analysis of these fits also show significant differences in uniformity and trends for PC-3+ cells and PC-3− cells. An example is shown in Figure 27.

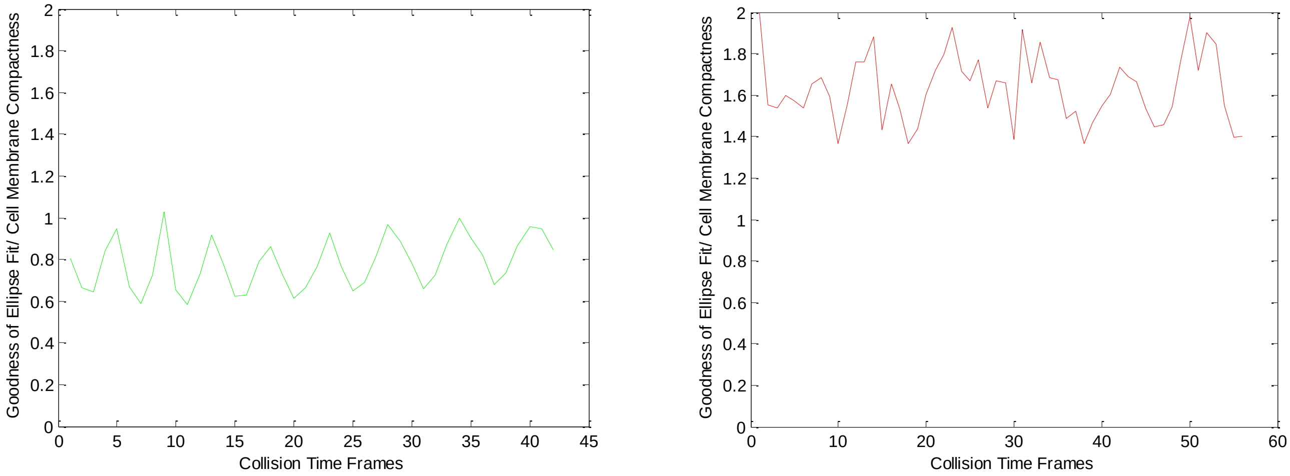


Figure 27. Graphs showing the cell membrane compactness or goodness of fit patterns for the two classes of cells (Left: PC-3+ with ID 108 and Right: PC-3− with ID 4).

c. Eccentricity

Eccentricity is a measure of roundness of a cell. It is given by:

$$Eccentricity = \sqrt{1 - \left(\frac{Minor\ Axis}{Major\ Axis}\right)^2} \qquad (39)$$

It is observed that the live cells tend to take nearly circular shapes, especially when there are head-on collisions. However, this is not the case with dead cells.

d. Major Axis Changes:

Changes in major axis of cells (or fit ellipse) are very prominent as compared to changes in minor axis. In other words, the variance along major axis is much higher than minor axis. In case of PC-3+ cells the changes are uniform and show similar trends as axis ratio. However, this is not the case for PC-3− cells. PC-3− cells show irregular pattern over time indicating that there are significant differences in shape and cytoplasm distribution. The following graphs (Figure 28) show the trends for two classes of cells.

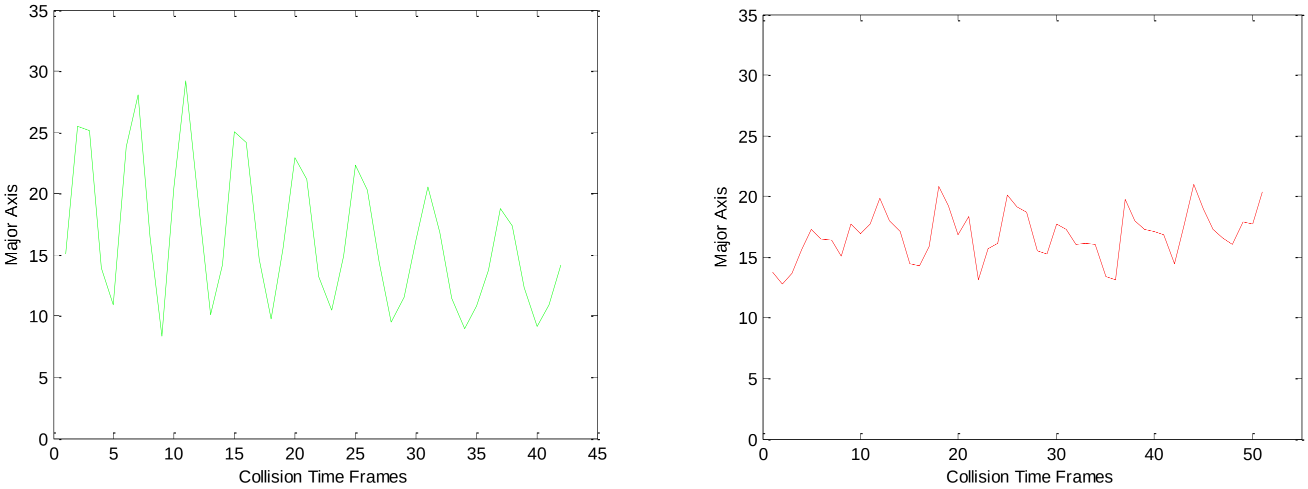


Figure 28. Graphs showing major axis (in pixel units) changes in both types of cells (left: PC-3+ with ID 108, right: PC-3− with ID 94).

e. Volume:

If we consider the shape of the cell as a spheroid with two axes being equal to major axis and the third one being minor axis, the volume will be given by $\frac{4}{3}\pi a^2 b$. A sphere of equivalent volume will have a radius $R = (a^2 b)^{\frac{1}{3}}$. We can also calculate an approximate cell mass, given mass density for a type of a cell.

f. Orientation:

The orientation of the cell is obtained from the eigenvector $[E_x , E_y]$ through ellipse fitting or PCA. It is given by:

$$Orientation = \left[\left(\frac{180}{\pi}\right) * \tan^{-1}\left(\frac{E_x}{E_y}\right)\right]^{\circ} - 90^{\circ} \qquad (40)$$

This can be used to automate the cases of oblique collisions and obtain the corresponding properties during head-on collisions.

The above were the mainly studied elliptic features in this thesis. Next we discuss the kinematic features which are mainly the velocity and the acceleration.

g. Velocity

The cell velocity is calculated as the rate of change of centroid position and is calculated through tracking the cell centroid across consecutive frames. Velocity is given by

$$\boldsymbol{V}_c = \frac{|\boldsymbol{P}_1 - \boldsymbol{P}_2|}{\left(\frac{1}{f}\right)} \tag{41}$$

where $\boldsymbol{P}_1$ is the initial position or the position in frame 1, and $\boldsymbol{P}_2$ is the position of the centroid in next frame (or frame 2). f is the frame rate of the video data. Magnitude of velocity and speed are synonymous, especially when measured between two consecutive frames. This is because the cells are assumed to move in a nearly straight line between the cylindrical posts.

The velocity of the cell is assumed to be caused by the fluid flow rate. By modelling the cell dynamics and free body diagram, we can obtain the velocity from a first order differential equation as shown in chapter 2. Figure 29 shows the velocity trend for cell 108 and cell 4.

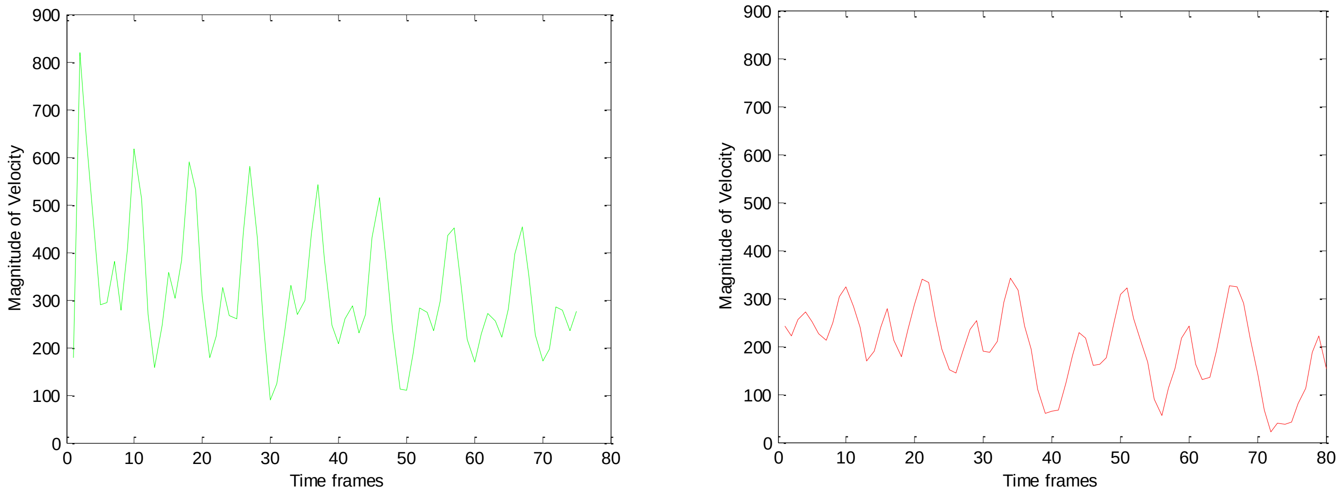


Figure 29. Graphs showing velocity (in pixels per second) magnitude changes in both types of cells (left: PC-3+ with ID 108, right: PC-3− with ID 4).

h. Acceleration

Acceleration is the rate of change of velocity. It can be obtained across three consecutive frames.

$$\boldsymbol{a} = \frac{|\boldsymbol{V}_{\mathrm{c}1} - \boldsymbol{V}_{\mathrm{c}2}|}{\left(\frac{1}{f}\right)} \tag{42}$$

where $\boldsymbol{V}_{\mathrm{c}1}$ and $\boldsymbol{V}_{\mathrm{c}2}$ are velocities between three consecutive frames.

i. *Correlation*

Correlation between two measurements can be a good factor to assess similarity of a pair of cells. For example, we can use the axis ratio and velocity measurements for a single cell to calculate the correlation. If the value crosses a threshold, say 90%, we can conclude that the cell under consideration is class 1. Further, if we have a set of labelled set of data, we can use correlation over time as a feature for classification.

The sample correlation coefficient between two measurements is given by,

$$R = \frac{\sum_{i=1}^{n}(u_i - \bar{u})(v_i - \tilde{v})}{(n)s_x s_y} \tag{43}$$

where n is the number of samples collected dynamically, ū is the dynamic mean of the samples obtained from first cell and ṽ is the dynamic mean obtained from samples of other cell. s_x and s_y are the sample standard deviations.

It is clear from the above discussed properties that most of these properties are indicative of shape and size features of a cell. We concatenate some of the elliptic features along with HOG features and discuss the classification performance of SVM and logistic regression in a later chapter.

Chapter 6

Physical quantity estimation

6.1 Introduction

This chapter proposes a model for estimating Young's modulus of elasticity through image processing. It shows an additional utility of the experimental setup that can be exploited. Current techniques that are used to calculate Young's modulus of elasticity make use of the Atomic Force Microscopy (AFM) [1, 2, 3, 27]. AFM is an expensive and time consuming procedure. As a way around this technique we propose a model for calculating the elasticity from the given setup. Although a rudimentary model, the expressions are similar to the one obtained from the AFM through the Hertz equation [1]. In the AFM papers, the authors show various types of indentation tips such as spherical, conical, parabolic and pyramidal. Our model is a collision of a cell with a cylindrical post. Collision of a cell with a cylindrical post has an elliptic contact surface [5]. This contact surface mathematically arrives at an elliptic integral. However, as a simpler approach, we can approximate the model as a sphere-sphere collision model in terms of the Hertz equation. Other complicated models that can be worked on are an ellipsoid and cylinder contact model. These models have an elliptic contact surface and have non-closed from expression and are not considered as a part of this thesis.

Recently conducted works [10, 11] use elongation as a measure of deformation. However, we give a more detailed perspective by proposing a simple model for of the deformity in terms of Young's modulus of elasticity. Our system is potentially suitable for estimating the Young's modulus for cells in a short period of time, unlike the methods used in AFM and [11] which are time consuming and expensive procedures. It has been shown in many similar works that elasticity for a cancer cell and for a benign cell have a significant difference. These studies demonstrated that cancerous cells are softer than cells from normal or non-malignant or even less-differentiated cancers. However, despite these differences, not many clinicians believe that it is possible to detect cancerous cell changes by mechanical properties [3].

Young’s modulus can be estimated from our setup when the cells collide with the micro-pillars with a force sufficient enough to create an indentation. The main force assumed to be acting on the cells is the Stokes’ drag due to the fluid flow. As shown in previous chapters, the Stokes’ force is dependent upon the geometry of the cell, viscosity and velocity of the cell. These can be extracted through image processing and tracking as shown in the previous chapters. The elasticity of the cell is also dependent upon the indentation caused upon collision. The previous works on AFM poke indentation probes at different locations on the cell membrane and then arrive at a value of elasticity. However, in our experiments, the elastic modulus is assumed to be a homogenous bulk property of the cell. The indentation can also be measured through video processing. It requires image patches of cells with sufficient resolution such that the indentation is perceivable. The next section describes the model through which elasticity can be calculated.

6.2 Hertz Contact Model for Cell Elasticity

In this section we explain in detail our contact model for estimating Young’s modulus of elasticity. The resultant force acting on the cell (in the absence of DEP force) by the fluid flow decreases with respect to time and finally reaches a constant value of zero at infinite time. The indentation caused on the cell membrane during the collision is a direct result of the Stokes’ drag. Figure 30 shows the instants before and after collision of a cell with the pillar in the microfluidic device. The time instant $t-1$ denotes the force on the cell just before collision. To measure the indentation from the image with a good accuracy, a high resolution video is preferable.

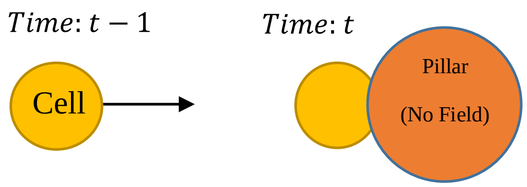


Figure 30. Cell flowing in the DEP experimental setup and collision with pillar.

Some important assumptions are made in the dynamic modeling of the cell in the given environment. These assumptions are listed below:

- The Naiver-Stokes' equation is solved for a creeping flow.
- Effects of gravity are considered to be negligible.
- Mass of each cell is distributed uniformly.
- Rotation of the cell is ignored.
- From a top-view of the setup, the electrode posts are circular and electric field is turned off.
- There is no loss of mass from cell during a collision with post.
- Adhesive forces or friction on posts are absent.
- It is a closed system.
- Poisson ratio for all cells is taken to be 0.5 [3].
- Forces on the cell are considered to be along the x-axis. Thus any changes that occur are along x-axis.

The Stokes' force for a creeping flow is dependent upon many factors that are related to properties of the cell and the fluid being used to flush the cells through the micro-fluidic device. A list of such quantities is given as follows:

- Density of flowing fluid $-$ ρ_F $(\mathrm{g/ml}^3)$ $-$ needs to be predetermined.
- Semi-major axis of cell – a (μm) $-$ calculated through image processing.
- Semi-minor axis of cell – b (μm) $-$ calculated through image processing.
- Equivalent radius of cell $-$ R (μm) $-$ same as a.
- Velocity of the cell $-$ v_c $(\mu\mathrm{m/sec})$ $-$ calculated through image processing.
- Dynamic viscosity of the fluid $-$ γ_F (Pascal-seconds or Pa · s) – needs to be predetermined.
- Velocity of the fluid $-$ v_F $(\mu\mathrm{m/sec})$ $-$ needs to be predetermined.
- Density of the cell $-$ ρ_C $(\mathrm{g/ml}^3)$ $-$ needs to be predetermined.
- Radius of the post $-$ R_p (μm) $-$ needs to be predetermined.

We use R as the equivalent radius of the cells for mathematical convenience. We use R assuming that the mass density ρ_c of the cells is preserved – this in turn keeps the overall force equation to be consistent. The equivalent radius R is simply taken to be equal to the semi-major axis of a cell, i.e., a just before collision.

$$R = a \tag{44}$$

As discussed earlier, the only force we consider to be acting on the cells is the Stokes' force along the horizontal axis. Thus only cases with nearly head-on collision are considered in this discussion. Cases of oblique collisions are not considered. Thus, the free body diagram for the cells just before colliding to the pillars can be represented as shown in Figure 31.

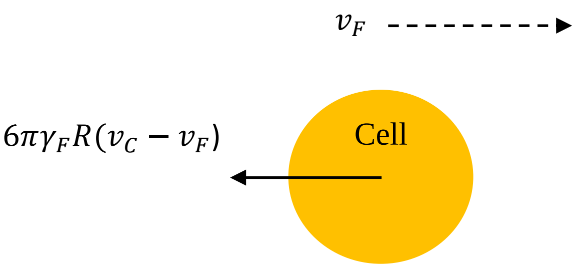


Figure 31. Free body diagram of the cell before collision.

Figure 31 shows the direction of fluid flow v_F and the total Stokes' force $6\pi\gamma_F R(v_C - v_F)$. Here the external force F_s, as previously shown in equation (1), is assumed to be zero. The resultant force on the cell is $m_c \frac{dv_c}{dt}$. In this equation, R, v_c and $\frac{dv_c}{dt}$ can be estimated through image processing at a time instant just before the collision. The term $6\pi\gamma_F R(v_c - v_F)$, is the viscous drag (Stokes' force) caused due to the viscous fluid medium flowing at a rate of v_F.

After hitting the post, the cell membrane experiences an indentation x as shown in Figure 32(a). This indentation can be explained by the Hertz contact model [1, 2, 3]. The indentation x can be calculated through image processing as given in (45).

$$x = R - R' \tag{45}$$

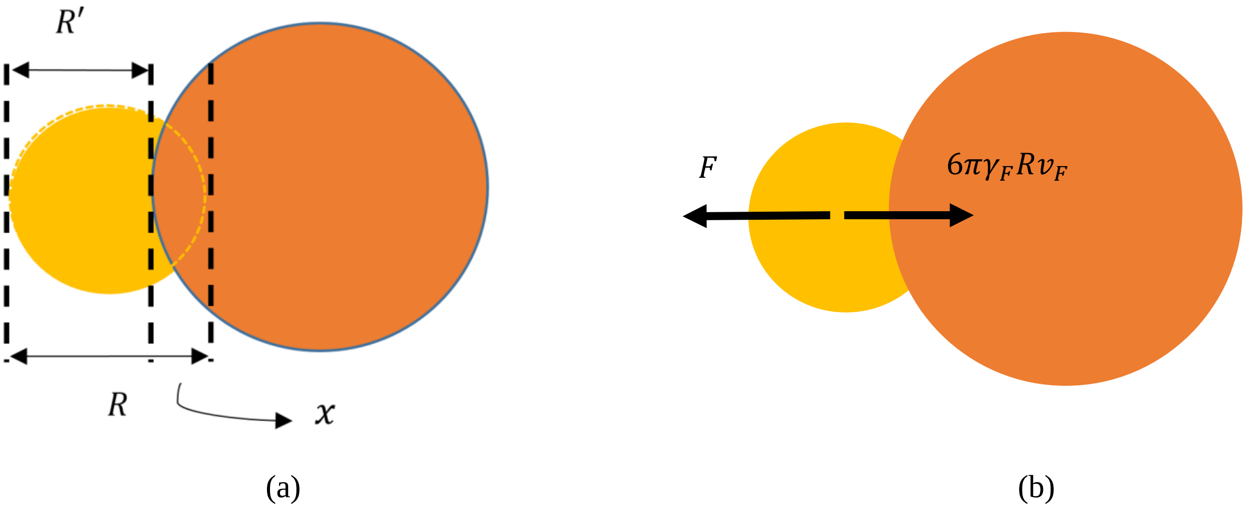


Figure 32. (a) Shows indentation x caused in the cell due to collision. (b) Free body diagram of the cell after collision.

Indentation x is caused due to the normal force on the cell upon collision to the post. Although normal force is also an external force on the cell form the walls of the pillar, it is not included as a part of F_S. The Young's modulus of elasticity (E) of the cell with Poisson ratio of 0.5 can be estimated by Hertz model, as discussed earlier and given in (8):

$$F = \frac{16}{9} E \sqrt{\frac{RR_p}{R + R_p}} x^{3/2}$$

Figure 32(b) shows the free body diagram of the cell when it comes to a complete rest after hitting a pillar. When the cell hits the pillar, it comes to a complete rest. Thus the resultant force $m_C \frac{dv_C}{dt}$ and the velocity v_C of the cell become zero at the collision instance. The two major forces acting on the cell upon collision with the pillar are the normal force from the wall of the pillar and the force due to viscous fluid flow. From Figure 32(b), the normal force on the cell will be same as (8). Upon applying these conditions (i.e., zero resultant force on the cell) in (1) and (8), we obtain an expression given by:

$$6\pi\gamma_F R v_F = \frac{16}{9} E \sqrt{\frac{RR_p}{R + R_p}} x^{3/2} \tag{46}$$

E can now be calculated from the equation given v_F and γ_F. Velocity of the cell v_C and acceleration $\frac{dv_C}{dt}$ become zero. Under the assumption that other external forces (F_S) are negligible, the above equation can be rearranged and simplified as follows:

$$E = \frac{9x^{-3/2}}{16} \sqrt{\frac{1}{R} + \frac{1}{R_p}} \, [6\pi\gamma_F R v_F] \tag{47}$$

The value of elasticity may be calculated in images with high resolution, high frame rate and higher magnification as we will have a good observation for value of indentation x.

The above work shows an idealistic means to calculate the cell elasticity from the given setup through image processing. The Young's modulus of elasticity can be calculated when there is a collision, and it will be calculated as a bulk property. For the dead (PC-3−) cells, this can be challenging and more error prone due to the ambiguous shapes they attain. In such a case, fitting active contours in high resolution images is a viable option for analysis.

Rewriting the above expression as a discrete approximation (from Figure 32(a)) will result in the expression given below:

$$E = \frac{9(R - R')^{\frac{3}{2}}}{16}\sqrt{\frac{1}{R} + \frac{1}{R_p}}\ [6\pi\gamma_F R v_F] \tag{48}$$

Once we obtain a value of the cell radius and indentation through images, we can convert it to the real word units through camera parameters given in chapter 3. For calculating the Young's modulus for a collision with a flat surface, we can simply replace $\sqrt{\frac{1}{R} + \frac{1}{R_p}}$ in (48) by $\sqrt{\frac{1}{R}}$ and calculate indentations similar to the procedure discussed above. We require a high resolution and high magnification image data to observe the indentation. Collision with a flat surface is practically easier to model and calculate. However, the sphere-sphere collision model would require different techniques that could give us information about the third dimension. Measuring indentation upon collision with a cylindrical pillar is practical but the solution is not available in closed form.

It has been shown [3] that Young's modulus of elasticity of cells usually range between $1 - 100$ kPa. This range was established through studying different types of cells under the AFM, both cancerous and non-cancerous. Typical indentation value on the cell by the AFM indenter ranges between $100 - 1000$ nm. In [3], the Young's modulus was calculated for a 200nm indentation on the cell wall. Usually, the spherical indenters are shown to give good estimates for the Young's modulus through the Hertz contact model [1, 2, 3]. Typical diameter of the indenter is between $1 - 15$ um, as shown in these works. Indentation of this small scale is not observable in our existing image data. The observable changes in the existing data are seen in the major axis of the cells as discussed in section 5.2. These changes signify the elongation of the cell where the force will be directly proportional to changes in cell dimensions (or major axis) as per the Hooke's law [49]. In such as case, the cell needs to be stretched up to the elastic limits to have a reliable measurement of elasticity.

Currently, the DEP based microfluidic device system has a 20 um diameter cylindrical pillar. As we showed above, contact between two non-conforming shapes does not give a closed form solution and may not be an effective approach for calculating Young's modulus from the experimental setup. Moreover, the cell has to collide nearly head-on, and at the same place, with the pillars for a reliable measurement of indentation. In reality, the cells rotate in the buffer fluid and may not hit head-on facing the same region with subsequent collisions. Moreover, image resolution is insufficient to observe the actual indentations. Techniques such as super resolution imaging can be useful to upsize low resolution images. However, they require multiple image samples of the same instance of head-on collision. For the existing setup, using an electron microscope, or at least super resolution microscopy can be a good alternative to observe the indentation which can resolve down to a scale of ~ 250 nm.

Chapter 7

Framework evaluation and results

7.1 Introduction

In this chapter we discuss the performance, accuracy and results of the algorithms that constitute our cell phenotype analysis framework. We start with establishing metrics which will be used for evaluation of the algorithms. The performance of the framework is dependent upon the performance of its constituents which are primarily object detection, tracking and classification. The performance of classifiers is dependent upon feature extraction and data collection – a pure noise free data usually gives better results. The analysis of the classifiers is purely based on the data that we have collected through tracking and automatic labeling. It is important to note that each component in the framework is directly dependent upon each other. Error or inaccuracies in blob detection are propagated to tracking. Inaccuracies in tracking are propagated to the feature extraction and thus classification.

7.2 Evaluation Metrics

We define evaluation metrics for detection, tracking and classification, each separately in this section. Performance of the framework will thus be established based on the performances of each of its constituents.

Object Detection:

The object detection is a cascaded process that uses background subtraction followed by Difference of Gaussian (DoG). We follow evaluation guidelines similar to [32, 39] for our blob detector. Our DoG based object detection is a threshold dependent process. These fixed thresholds were set for background subtraction after thorough manual inspection of the grayscale images and the DoG filter dimensions. These thresholds are different for different video data sets. Performance of the detector is dependent upon these thresholds and automating the threshold selection process should be considered for future works. Our evaluation of object detection is done by manually calculating the metrics for video 1 (discussed in section 3.3). The objects under consideration are cells, and evaluation metrics used for our object detector are as follows:

- Correct Detections (CD): Number of correctly detected cells (with bounding boxes: True positives).
- False Detections (FD): Number of regions incorrectly detected as cells (False positives).
- Detection Failure (DF): Number of regions not detected but is actually a cell (False negatives).
- Merged Region (MR): Number of detections of two or more cells as a single object.
- Partial/Split Region (SR): Number of detections of a single cell as two or more objects.
- Ground Truth (GT): The actual number of cells. This is same as CD+DF+MR.

Using the above evaluation parameters, we can now calculate the precision (P), recall (R) and F measure for object detection.

$$Precision\ (P) = \frac{CD}{CD + FD}$$

$$Recall\ (R) = \frac{CD}{GT}$$

$$F\ measure\ (F) = \frac{2PR}{P + R}$$

Multiple Object Tracking:

We track cells in the video dataset through KLT and Hungarian algorithm. Although a robust tracker for the given data that handles changes in the object sizes (bounding box), much of its performance is dependent upon the performance of the blob detector. We follow evaluation metrics similar to [40, 41] for tracking along with some of our own metrics. We robustly handle changes in bounding box sizes by detecting the blobs and connected component analysis in every frame rather than propagating the boxes over frames. In this algorithm, KLT feature tracker is used such that we detect and obtain key feature points of cells (centroids) in current frame and use KLT to locate corresponding feature points in the next frame. Also, the Hungarian algorithm is used for connecting the feature points obtained from KLT to form a track. The algorithm uses its cost as Euclidean distance and has an upper threshold value for associating tracks. Considering these factors, we use the following set of evaluation metrics for multiple object tracking:

- Detected Tracks (DT): Number of tracks of an object (in two consecutive frames) as detected by the algorithm.
- False Tracks (FT): The number of tracks assigned as cells when they are not actually cells.
- Change in ID (CI): Number of times the track ID changes from the one assigned originally. This is equivalent to number of missed detections/tracks. This is primarily due to inaccuracies in blob detection such as merged detections, split detections or object detection threshold. This also includes the cases where a cell in current frame gets matched to a different cell in next frame.
- Correct Tracks (CD): Number of tracks per consecutive frame detected correctly.
- Ground Truth (GT): Number of frames with actual tracks for each cell.
- Object Tracking Error (OTE): The average error in all tracked points with respect to centroid in terms of Euclidean distance (pixel units):

$$OTE = \frac{1}{f}\sum_{j=1}^{f}\left[\frac{1}{N_j}\sum_{i=1}^{N_j}\sqrt{\left(x_{pj}^{i} - x_{cj}^{i}\right)^2 + \left(y_{pj}^{i} - y_{cj}^{i}\right)^2}\right] \tag{49}$$

where N_j is number of cells in jth frame, f is number of frames, $\left(x_{pj}^{i}, y_{pj}^{i}\right)$ and $\left(x_{cj}^{i}, y_{cj}^{i}\right)$ are the position obtained through KLT and centroid for the ith cell in jth frame respectively.

Thus the precision (P) and recall (R) for tracking can be given by:

$$P = \frac{CD}{CD + FT + CI}$$

$$R = \frac{CD}{GT}$$

F-measure can be calculated using the precision and recall given above.

Classifiers:

Classifier performance is dependent upon the feature data used, number of samples of a given class, and proportion of data of different classes (for example, having just one feature point for a class will not suffice for classifying similar test points). We use the following performance measures for the classifiers:

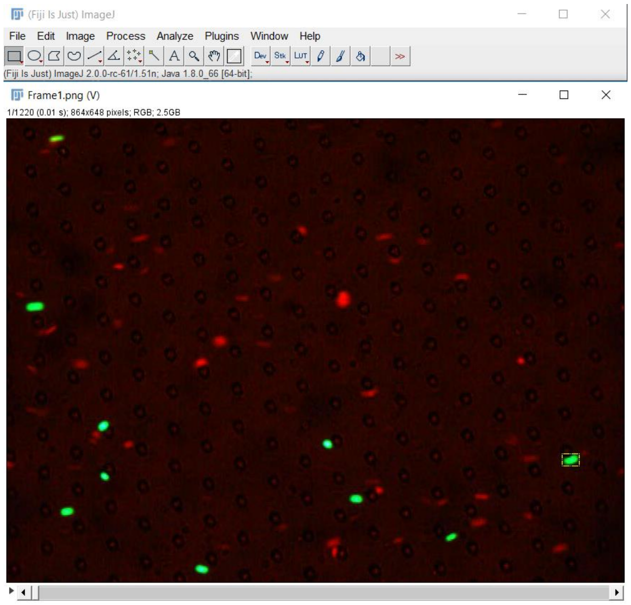
- Training Error: The error obtained on the training class labels when the training data is used as the test data on a fully trained model. It is defined using the zero-one loss function I, where y_i is the actual class label and $\hat{y}_i$ is the predicted class label:

$$Training\ Error = \frac{1}{N}\sum_{i=1}^{N} I(y_i \neq \hat{y}_i) \tag{50}$$

- 10-fold Cross validation: We divide the training data into 10 equal parts. We take 9 parts as training data and 1 part as the validation data. We keep rotating the validation data 10 times and average the error from each fold to find the final error.
- Test Error or Accuracy: Error on test data. Similar to training error, this will be a zero-one loss as well.
- Precision, recall, false positive rate and F1 score (or F measure) for the test data were also calculated.

7.3 Ground Truth

Various image processing and annotating tools were tried on the video data sets for generating ground truth. One problem upon using these software was that some of them used the same algorithm for detection. Measuring against these tools would introduce some degree of ambiguity. As a way around we used the FIJI image processing and annotation package [43]. Although a time consuming process, we manually determined ground truth for detection, tracking and classification that could help us in calculating the performance metrics of our algorithms. Figure 33 below shows the FIJI (Fiji Is Just ImageJ) framework interface that was used for ground truth through manual inspection. We discuss the performance of our algorithms in the upcoming sections.



(Fiji Is Just) ImageJ 2.0.0-rc-61/1.51n; Java 1.8.0_66 [64-bit];

1/1220 (0.01 s); 864x648 pixels; RGB; 2.5GB

Figure 33. FIJI image processing framework interface.

7.4 Blob Detection Performance and analysis

We perform analysis and evaluation of our 'DoG + connected components' based blob detector as per the evaluation metrics defined in the previous section. Video data set (video 1) shown in Figure 34 has been used in evaluation of blob detector. These videos have been extracted from the experimental setup shown in chapter 3. The video data shows PC-3+ (green) and PC-3− (red) cells. Evaluation of the blob detector was performed manually by counting the detected blobs on the data set over many frames using the FIJI image processing package [43]. After obtaining the evaluation metrics for each frame, average precision, recall and F measure for the algorithm were calculated.

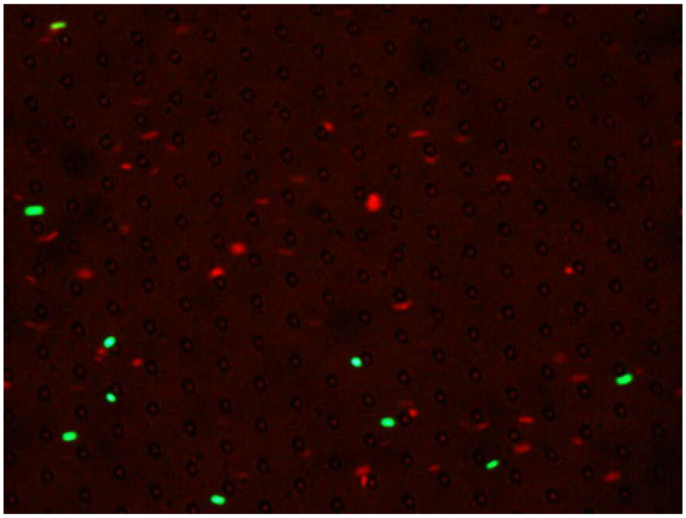


Figure 34. An example image frame from the video data set (video 1) used for analysis.

The evaluation metrics were calculated for each frame through manual inspection. We obtained the evaluation data shown as an example in Table 1. This data was collected for over 40 randomly chosen frames from video 1 shown in figure 34.

Table 1. Shows an example of evaluation metrics and precision-recall values for cell detection calculated for nine frames.

Frames	Detections	False Detection	Correct Detections	Detection Failure	Merged Detection	Partial Detections/Split detections	Ground Truth	Precision	Recall	F-Measure
1	63	0	63	3	2	0	68	1	0.926	0.962
2	62	0	62	2	4	0	68	1	0.912	0.954
3	65	1	64	1	2	1	67	0.985	0.955	0.969
4	64	0	64	2	2	0	68	1	0.941	0.969
5	67	1	66	1	1	1	68	0.985	0.971	0.978
6	62	0	62	2	3	0	67	1	0.925	0.961
7	63	0	63	1	3	0	67	1	0.940	0.969
8	60	0	60	2	2	1	64	1	0.938	0.968
9	61	0	61	3	4	2	67	1	0.910	0.953

After obtaining average values of the evaluation metrics, the average precision, recall and F measure were calculated as the performance of the blob detector. Table 2 shows the average performance of the blob detector.

Table 2. Average performance of the DoG based blob detector over all 40 frames.

Precision	Recall	F-Measure
0.9884	0.9610	0.974

The blob detector shows reasonable performance as per the results obtained for video data (result for a single frame shown in Figure 35). The performance of the blob detector is highly dependent upon the DoG filter size and the intensity of the blobs. The filter size was selected empirically such that minimal false positive blobs be detected in the given set of video frames. Despite this value of the filter size, some blobs remain undetected or are detected intermittently in a few frames. This could be mainly due to the dynamic behavior of the cell shapes, illumination and the depth at which they travel. The algorithm is a combination of DoG and connected components and uses thresholds for detecting the cell blobs in the process of converting grayscale to binary. Any deviations in filter size or thresholds from the currently set value results in either detecting a lot of false blobs or in missing out many of the correct blobs. This setting of the detection parameters causes much of the error and inaccuracies in identifying cell blobs in the frames.

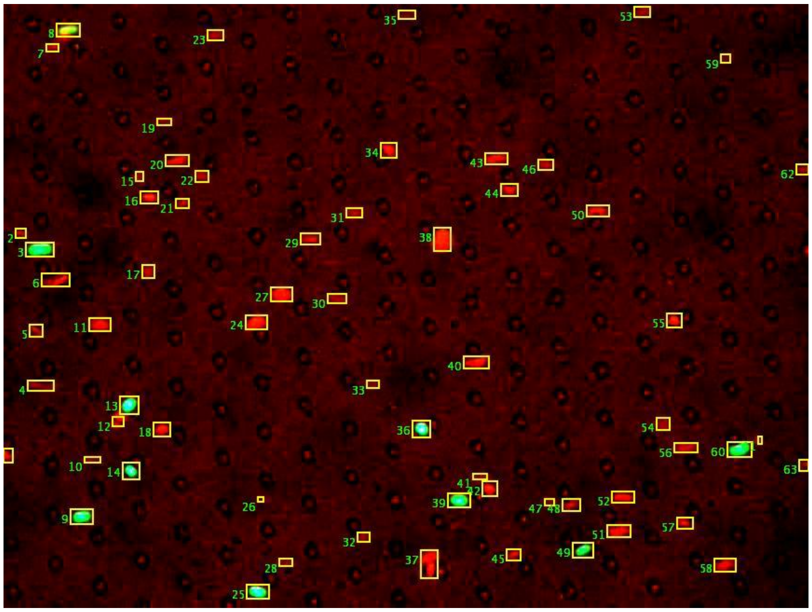
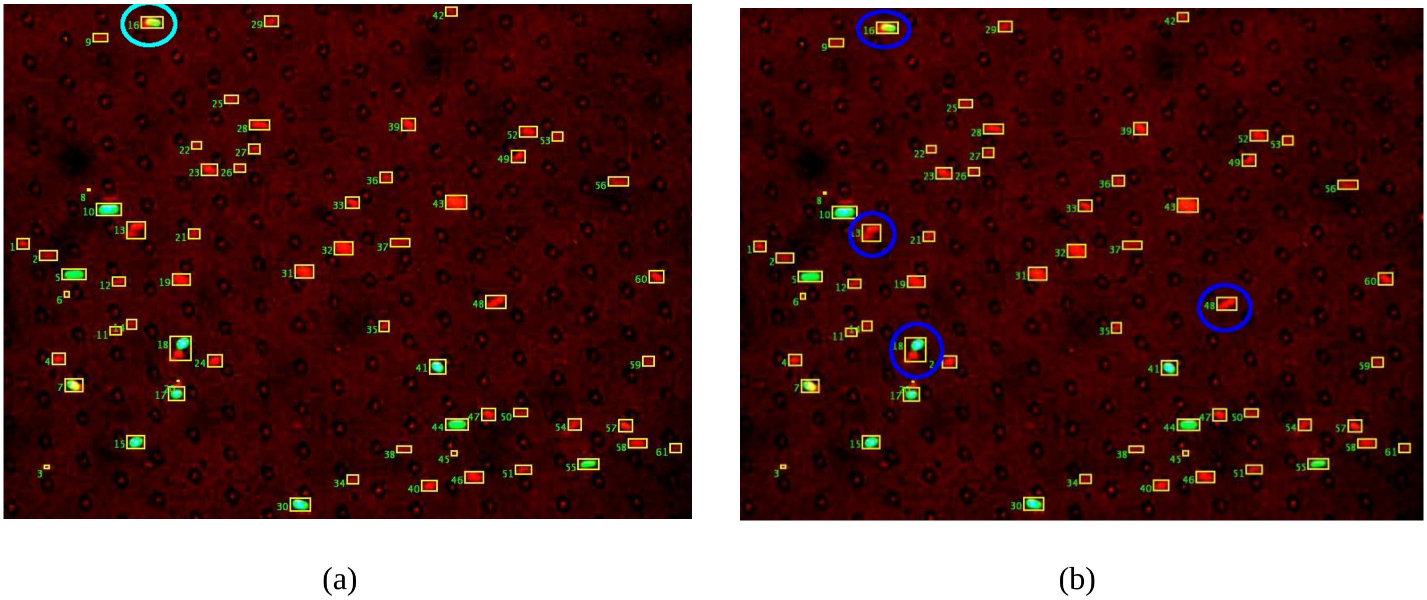


Figure 35. Example blob detection result for a single frame.

Other issues that occur in our detector are the occlusion, merging and partial detections (or bounding boxes). Detecting occlusion cases is almost always mishandled in this algorithm. There are cases where one cell almost completely occludes another cell in a few frames. This makes detecting the hidden cell very difficult. Figure 36(a) shows an example image of nearly complete occlusion of a red cell below a green cell. Figure 36(b) shows merged detections. Merged cells are considered to be the same blob by the detector and gets treated as the same object.



(a)

(b)

Figure 36. (a) A complete occlusion case circled (in cyan). (b) Merging and occlusion cases have been detected as single blobs (in dark blue). The same frame has been shown in (a) and (b) to show the two cases separately.

There were no cases where a single blob is detected as two or more. However, there are cases where the blobs are detected partially such as bounding boxes partially covering the blobs. These cases arise mainly due to the thresholds governing the background subtraction. Thresholds have been set at a fixed value empirically. Any changes in threshold can cause increased cases of incorrect detections or missed detections. Further, most of the bounding boxes are tightly bound on the cells. There are very few cases where the bounding boxes miss the cell boundary by the width of a pixel or two. This is yet again accounted for the thresholds and filter size used in the Difference of Gaussian. Figure 37 below shows loose detection cases and incorrect detections.

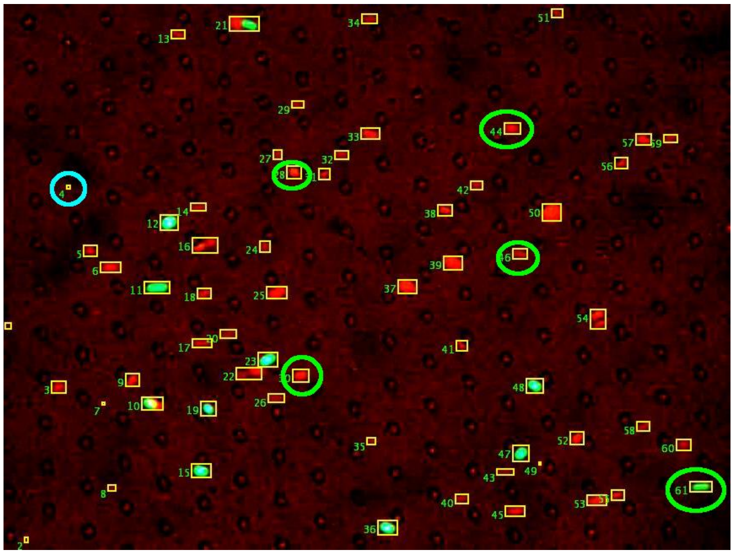


Figure 37. Example image showing false detection encircled in cyan and loose detection boxes in green.

7.5 Tracking Performance and Analysis

To evaluate our KLT + Hungarian Algorithm based tracker, we obtain the tracking evaluation metrics through manual inspection of all the objects detected in video. Performance of our tracker is dependent upon the performance of the detection process. Also, the tracking is important for the feature extraction process from the cells. In the given video dataset, dead cells are much more in number than the live cells. In order to extract proportional amounts of feature data from the live (green) and dead (red) cells, the threshold and filter size of the blob detector were empirically kept at a desired value during the training phase on top of which tracking is implemented. Extracting features for all detected cells results in an excessive data for dead cells than the live cells. This negatively affects the classifier performance.

The tracker performance is evaluated in conditions such that not all blobs are detected, unlike in evaluation of our blob detector where the best performing parameters were chosen. We use a filter size of 21x21 pixels with standard deviations $\sigma 1$ and $\sigma 2$ with values 15 and 20, respectively, to detect the blobs for tracking. The tracker was analyzed on whole video 1 which had 99 objects in total. The evaluation metrics recorded for tracking are shown in Table 3. These metrics were further used to calculate the precision, recall and F measure for all the 99 tracked objects (15 live cells and 84 dead cells).

Table 3. Shows the example evaluation metrics and precision-recall values calculated for 99 tracked objects.

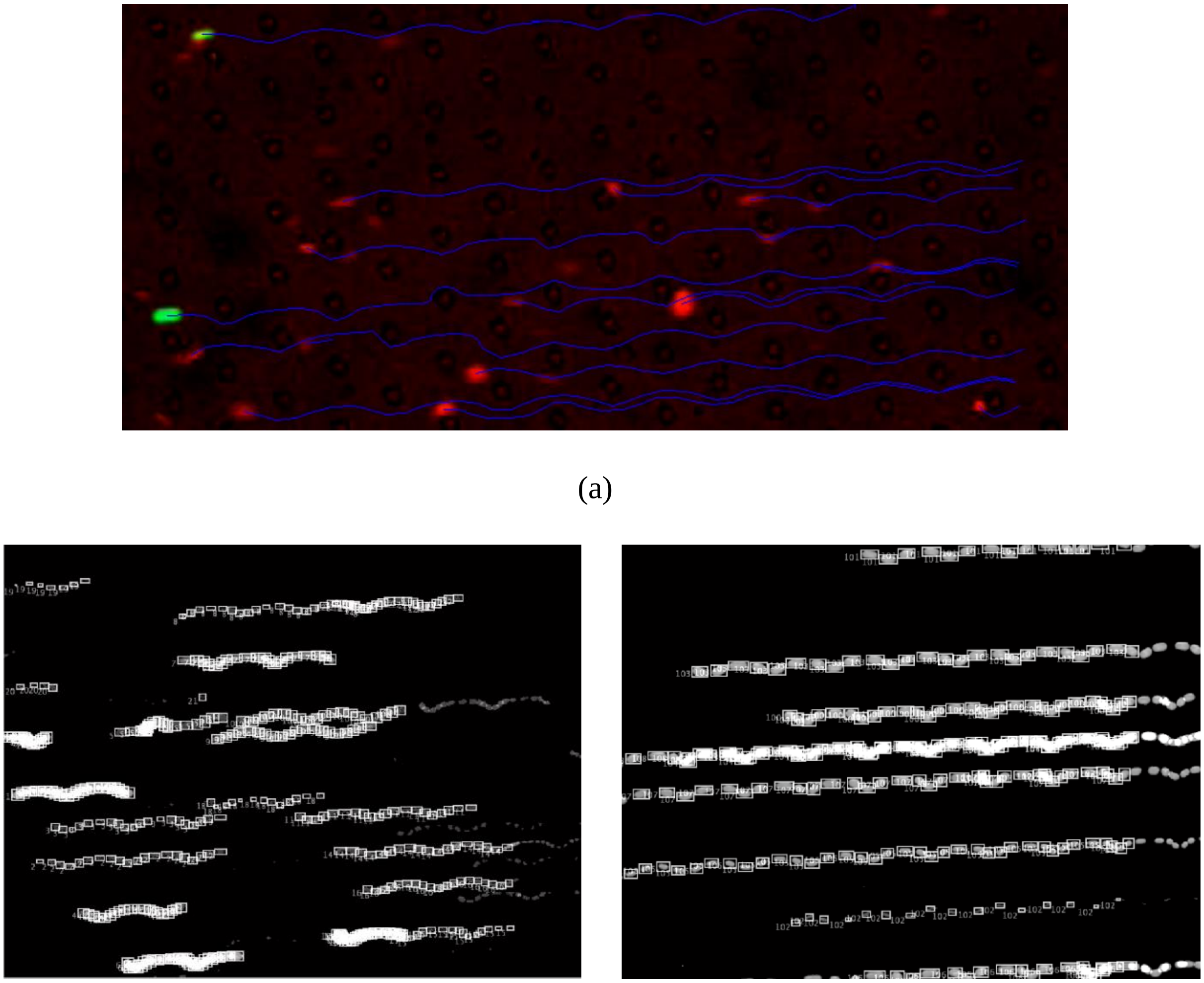
Object Number	Detected Tracks	False Tracks	ID Change	Ground Truth	Correct Detections	Precision	Recall	F-Measure
1	742	1	1	742	741	0.997	0.999	0.998
2	54	0	0	54	54	1	1	1
3	54	0	0	54	54	1	1	1
4	101	0	0	101	101	1	1	1
5	49	0	0	49	49	1	1	1
6	732	3	1	733	729	0.995	0.995	0.995
7	57	0	1	58	57	0.983	0.983	0.983
8	34	0	0	34	34	1	1	1
9	36	0	0	36	36	1	1	1
10	33	0	0	33	33	1	1	1
11	24	0	0	29	24	1	0.828	0.906

Similar to the evaluation of blob detection, we calculated the average performance measures in terms of precision, recall and F-measure for tracking. We also calculate the object tracking error (OTE) for which the ideal value (or the ground truth) is 0. Table 4 shows the results obtained for tracking over all frames in video 1.

Table 4. Average performance of the KLT based tracker with respect to ground truth.

Precision	Recall	F-Measure	OTE
0.9898	0.9388	0.9552	5.05

Our tracking algorithm is completely based on optical flow (KLT). As compared to previous results obtained through optical flow [32], we show robust and remarkable improvements. Our tracker also efficiently handles changes in size of bounding boxes. Most of the inaccuracies in this tracker are due to inaccuracies in the blob detection. OTE is the error due to the KLT tracker itself and is obtained with respect to blob centroids as the ground truth. Figure 38 shows example tracking results of detected cells.



(a)

(b)

Figure 38. Example tracking results. (a) Successfully tracked paths (in blue). (b) Successfully tracked size varying bounding boxes for a sequence of frames.

Most of the tracking failure cases appear due to inaccuracies in blob detection. The most common reasons (which also lead to inaccuracies in blob detection) are merging of two cells and failure to detect the cell in a single frame. This leads to tracking the incorrect cell or change of cell ID or both. Once the blob detector misses out a cell in one of the frames and then detects it again in some other frame, the cell ID changes, despite the track being correct. This causes a single cell to be treated as two or more different objects. We found 5 cases of merging/occlusion in the chosen video (video 1). Mostly, the red cells were beneath the green ones or in parallel.

Cases of complete occlusion are hard to detect and may require additional information about the depth axis. In some cases, the detector does not show these problems even after merging.

Other reasons for inaccuracies in following the track of a cell are cost thresholds given to the Hungarian linker. Sometimes, despite the KLT predicting the correct position in the next frame, the cost threshold in Hungarian data association is less than the displacement of the cell. This causes the tracks to change or simply cease. The cell in the missed track is then treated as a new object with a new track. These inaccuracies related to change of cell ID are shown in Figure 39 for one example case, which also shows automatic class labeling.

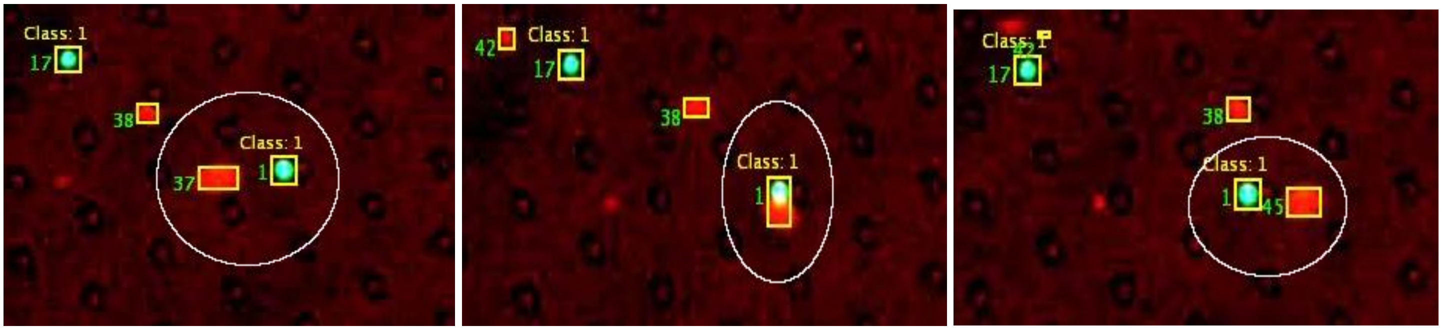


Figure 39. An example showing the change of cell ID. Image on the left shows cell 37 approaching cell 1. Image in the middle shows a merge in two cells causing some deviation in track for cell 37. Image on the right shows cell 37 changed to cell 45 (all encircled in white). The ID of cell 1 remains unaffected. Cell 42 in the middle image also faces a similar situation.

Figure 40 shows example inaccuracies caused due to two moving and merging cells and due to the threshold of Hungarian data association.

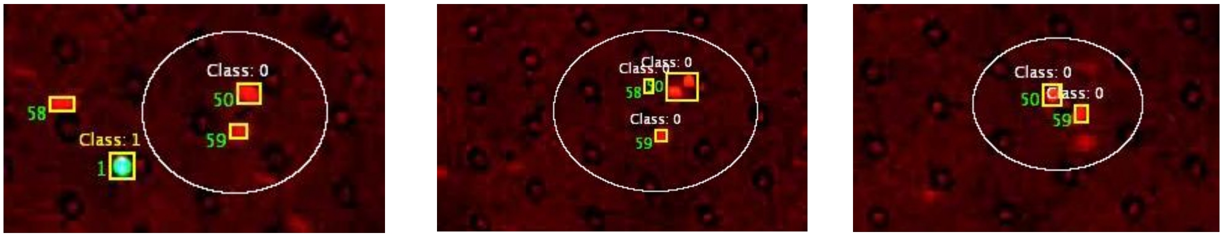
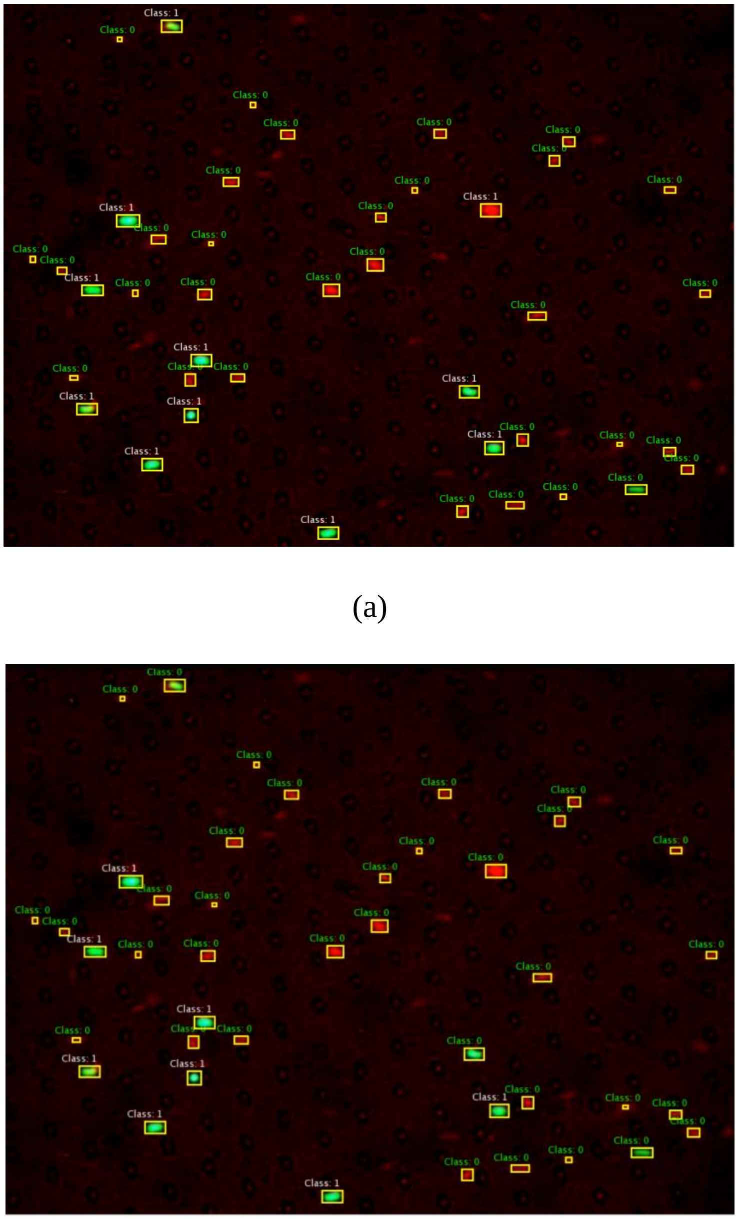


Figure 40. Image sequences of cell 50 and cell 59 moving side by side, cell 58 merging with cell 50, and ID 59 being incorrectly assigned to a different cell. Cell 50 still manages to come out consistently labeled.

7.6 Classifier Performance and Analysis

We primarily use linear support vector machine and logistic regression for feature based classification in our work. The features extracted are described in section 4.5 and chapter 5. Cells are automatically labeled during the period when the DEP field is turned ON, and features are extracted until the cells disappear from the frame.

The classification results obtained from linear SVM and logistic regression are evaluated based on test error, train error, 10-fold cross validation and precision-recall metrics. Figure 41 below shows some classification results for SVM and logistic regression on a test frame. We use two videos of live and dead PC-3 cells where one is used to collect train data (video 1, discussed in section 3.3) and the other is used as the test data (video 2, discussed in section 3.3). The data set used for classification has ~500 data points belonging to green cells (PC-3+) and ~700 data points for red cells (PC-3−).



(a)

(b)

Figure 41. Example classification results from (a) logistic regression and (b) support vector machine. Class 1: Live and Class 0: dead.

We used four different combinations of HOG cell sizes for training and testing our classifiers. Tables 5 and 6 show the average performance measures of logistic regression and Support Vector Machine respectively when trained and tested for different HOG cell sizes. The largest HOG cell size we use is 4x4 and the smallest being

1x1. Logistic regression was trained using gradient descent for about 5000 iterations. The learning rate was fixed to 0.01 and the regularization parameter was set to 0.8. The SVM was trained using sequential minimal optimization (SMO) for about 15000 iterations.

Table 5. Results for logistic regression for different HOG cell sizes. Highest accuracy in bold.

HOG Cell Size	Recall	Precision	False Positive Rate	Accuracy	F1 Score
1x1	0.722	0.960	0.013	0.912	0.816
2x2	0.782	1	0	**0.936**	0.866
3x3	0.744	1	0	0.928	0.850
4x4	0.620	0.904	0.026	0.876	0.732

Table 6. Results for Support Vector Machine for different HOG cell sizes. Highest accuracy in bold.

HOG Cell Size	Recall	Precision	False Positive Rate	Accuracy	F1 Score
1x1	0.724	0.960	0.0130	0.914	0.822
2x2	0.798	0.926	0.0254	0.922	0.854
3x3	0.814	0.922	0.0266	0.930	0.860
4x4	0.782	0.976	0.0062	**0.934**	0.864

From the above table, logistic regression performs best with a HOG cell size of 2x2 with an accuracy of 0.936, whereas SVM performs best with a HOG cell size of 4x4 with an accuracy of 0.934. A HOG cell size of 2x2 has more feature points and more dimensions as compared to a 4x4 cell size. The size of a feature vector affects the computational efficiency of a classifier– more the size, more the computations. The difference in performance of logistic regression on HOG 2x2 is very less compared to SVM on a HOG cell size of 4x4. Thus it is preferable to use an SVM for HOG cell size of 4x4. The training error for logistic regression is found to be 3.33%. Meanwhile for SVM we obtain a 6.67% training error. 10-fold cross validation errors are 5.3% and 6.4% for logistic regression and linear SVM, respectively. 10-fold cross validation error is calculated for 10 batches of randomly shuffled training data, and then all of them are averaged.

It is seen that the elliptic and geometric features obtained in chapter 5 show differences in pattern, range and a different degree of variance between two classes. We concatenate elliptic features along with the HOG features and note for changes in performance of the two classifiers. We primarily use elongation, membrane compactness and the eccentricity as additional features. Table 7 and 8 show the performance of the two classifiers using HOG features combined with elliptic features.

Table 7. Results for logistic regression for different HOG cell sizes combined with elliptic features. Highest accuracy in bold.

HOG Cell Size	Recall	Precision	False Positive Rate	Accuracy	F1 Score
1x1	0.744	0.962	0.0128	0.922	0.828
2x2	0.776	1	0	0.940	0.868
3x3	0.814	1	0	**0.948**	0.892
4x4	0.506	0.882	0.0024	0.846	0.638

Table 8. Results for Support Vector Machine for different HOG cell sizes combined with elliptic features. Highest accuracy in bold.

HOG Cell Size	Recall	Precision	False Positive Rate	Accuracy	F1 Score
1x1	0.766	0.928	0.0256	0.914	0.830
2x2	0.872	0.946	0.0190	**0.950**	0.906
3x3	0.852	0.956	0.0124	0.948	0.900
4x4	0.856	0.942	0.0184	0.946	0.892

From the above two tables, SVM has an increased classification accuracy and perhaps the highest accuracy overall for a HOG 2x2 cell size combined with the elliptic features. Comparing the results for HOG + elliptic features and HOG alone, the accuracy increased overall when the elliptic features are combined with the HOG. As the HOG cell size increases, the feature vector size decreases and large scale features become prominent. The cells in the given data are small and small scale features are necessary for reliable classification. Using a HOG cell box size of a large value reduces the prominence of small scale features and we can expect a reduction in classifier accuracy. Overall, the best HOG cell size for the given data is 3x3 which is computationally easier to process and at the same time gives a good accuracy. Despite having higher accuracy of classification, false positive rates are higher in SVM than logistic regression. However, the F1 scores are higher for the SVM.

The classifiers perform reasonably well using the obtained features, as evident from Tables 5-8. Addition of features such as axis ratio and compactness have a high variance between the two classes and thus give some boost to the classification accuracy. Much of the inaccuracies in classification are accounted for inaccuracies in tracking and feature extraction. Extraction of features from image patches is directly dependent upon the amount of information present in the image. We will be able to extract more information with higher resolution data.

CONCLUSION

Our work showed a multifaceted framework for phenotyping of cells under the DEP experimental setup. It successfully established a technique for identification, tracking, automatic class assignments, feature extraction and classification of PC-3 cells using the microfluidic device and DEP. We also showed a model for calculating Young's modulus of elasticity of cells in a similar approach as done in AFM, but potentially achieving the same for a batch of cells by consuming less time and being less expensive. Unfortunately, the data did not have enough information to provide us with a value of indentation. Another unique aspect of our work is the temporal feature extraction. It reveals dynamic changes in features of cells and classification results show that similar type of cells show similar dynamic behavior. Moreover, automatic class labeling enables us to assign class labels automatically for supervised learning rather than manually spending time to annotate images. It is shown that optical flow techniques perform well for tracking the cells in the video data sets with an average F-measure of 0.95 and an OTE of 5.05 pixels. The DoG and connected component combination used for blob detection also showed an average F-measure of 0.97. The classifiers also show a good performance in the test data. Logistic regression has an accuracy of 0.948 and SVM has an accuracy of 0.950 with the chosen parameters. Our framework can also be used in real-time for tracking, data collection and classification along with physical property calculations. We also use RBF and polynomial kernels in SVM, however, the linear SVM gives the best results for our extracted features.

IDEAS FOR FUTURE

This work has a huge scope and lots of room for improvement. As we discussed in our work, there are differences between cancerous and benign cells. These differences have been established theoretically and also based on practical analysis from microscopic images. The next step will be to obtain data for cancerous vs. benign cells and apply the same framework to analyze differences. Background subtraction and detection algorithms might be improved by adaptive thresholding or other advanced background subtraction techniques. However, lack of differences in illumination of the cells and background can again lead to inaccuracies. Tracking can be a complete framework in itself which can handle occlusion, merging and cases of blob reappearances. Most importantly, an improved image resolution will give enhanced features which can improve classification accuracy instead. Usage of deep learning is highly recommended to look forward for very accurate classification results on high resolution images. But increasing resolution will have also increase computational complexity in its processing. Other features such as nucleus shape and size will also be a useful way to characterize the cells. Our model for elasticity shows an idealistic equation with many assumptions. This model is dependent upon the design of the microfluidic device pillars and to calculate its value through video processing, a high frame rate will improve accuracy and make the indentations on the cell wall more clear. For the purpose of sub-phenotyping response of the same type of cells under DEP in terms of frequency response, speed at which a cell is attracted to the post etc. can be a useful measure. Further, shape matching statistical techniques will also be a useful way to compare two kinds of cells. Scaling down the HOG features further and performing dimensionality reduction might also improve classification results. It is highly recommended that a higher resolution, frame rate and magnified data be used for the phenotyping framework. It is advisable that values of elasticity be calculated and measured against a standard device such as the AFM. Moreover, our framework can also be incorporated as a part of multi-physics software such as COMSOL. Further, our work has futuristic applications and can potentially give a new direction to the current treatment methods of diseases such as blood cancer. This setup can potentially be used as a device for selective killing of malignant cells at a very fast rate rather than current methods of localized treatment such as radiation therapy that also affects the benign cells. However, technical challenges such as effects of blood being directly passed through our system setup is something to be investigated upon.

REFERENCES

[1] Nataliia Guz, Maxim Dokukin, Vivekanand Kalaparthi and Igor Sokolov, “If cell mechanics can be described by elastic modulus: study of different models and probes used in indentation experiments”. *Biophysical Journal*, vol. 107, pp. 564-575, August 2014.

[2] R. Kiss, H. Bock, S. Pells, E. Canetta, A. K. Adya, A. J. Moore, P. De Sousa and N. A. Willoughby, “Elasticity of human embryonic stem cells as determined by atomic force microscopy”. *Journal of Biomechanical Engineering*, National Center for Biotechnology Information, October 2011.

[3] Małgorzata Lekka, “Discrimination between normal and cancerous cells using AFM”. *Bio-Nanoscience*, Springer, pp. 65-80, Jaunary 2016.

[4] Hadi Shafiee, Michael B. Sano, Erin A. Henslee, John L. Caldwell and Rafael V. Davalos. “Selective isolation of live/dead cells using contactless dielectrophoresis (cDEP)”. *Lab on a Chip*, Royal Society of Chemistry, vol. 10, pp. 438-445, Jaunary 2010.

[5] M. J. Puttock and E. G. Thwaite, “Elastic compression of spheres and cylinders at point and line contact”. *National Standards Laboratory*, Technical Paper No. 25, Commonwealth Scientific and Industrial Research Organization, Australia, 1969.

[6] P. R. Gascoyne, J. Noshari, T. J. Anderson and F. F. Becker, “Isolation of rare cells from cell mixtures by dielectrophoresis”. *Electrophoresis*, Wiley Online Library, pp. 1388-1398, April 2009.

[7] Michael B. Sano, Rafael V. Davalos and Paul Gatenholm, “Dielectrophoretic microweaving: Biofabrication of aligned bacterial nanocellulose for regenerative medicine”. *Proceedings of the ASME Summer Bioengineering Conference*, 2009.

[8] J. Kong, O. Sertel, H. Shimada, K. L. Boyer, J. H. Saltz and M. N. Gurcan, “Computer-aided evaluation of neuroblastoma on whole-slide histology images: Classifying grade of neuroblastic differentiation”. *Pattern Recognition*, Elsevier, December 2007.

[9] Claire Lifan Chen, Ata Mahjoubfar, Li-Chia Tai, Ian K. Blaby, Allen Huang, Kayvan Reza Niazi and Bahram Jalali, “Deep learning in label-free cell classification”. *Scientific Reports*, www.nature.com, March 2016.

[10] Guohua Bai, Ying Li, Henry K. Chu, Kaiqun Wang, Qiulin Tan, Jijun Xiong and Dong Sun, “Characterization of biomechanical properties of cells through dielectrophoresis-based cell stretching and actin cytoskeleton modeling”. *BioMedical Engineering OnLine and Open Access*, www.nature.com, 2017.

[11] Jonathan Lin, Donghyuk Kim, Henry T. Tse, Peter Tseng, Lillian Peng, Manjima Dhar, Saravanan Karumbayaram and Dino Di Carlo, “High-throughput physical phenotyping of cell differentiation”. *Microsystems & Nanoengineering*, www.nature.com, 2017.

[12] F. Kasmin, A. S. Prabuwono and A. Abdullah, “Detection of leukemia in human blood sample based on microscopic images: a study”. *Journal of Theoretical & Applied Information Technology*, vol. 46, no. 2, 2012.

[13] D. Guo, A. L. V. de Ven and X. Zhou, “Tracking and measurement of the motion of blood cells using optical flow methods,”. *Journal of Biomedical and Health Informatics*, IEEE, vol. 18, 2014.

[14] Michel B. Dillencourt, Hanan Samet and Markku Tamminen, “A general approach to connected-component labeling for arbitrary image representations”. *ACM Digital Library*, vol. 39, pp. 253-280, April 1992.

[15] B. K. Horn and B. G. Schunck, “Determining optical flow”. *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[16] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[17] A. Tabesh, M. Teverovskiy and H.Y. Pang, “Multifeature prostate cancer diagnosis and Gleason grading of histological images,” *IEEE Transactions on Medical Imaging*, vol. 26, no. 10, pp. 1366–1378, 2007.

[18] S. Liao, M. W. K. Law and A. C. S. Chung, “Dominant local binary patterns for texture classification”. *IEEE Transactions on Image Processing*, vol. 18, no. 5, pp. 1107–1118, 2009.

[19] Navneet Dalal and Bill Triggs, “Histograms of oriented gradients for human detection”. *Computer Vision and Pattern Recognition*, IEEE, June 2005.

[20] A. I. Baba and C. Câtoi. *Comparative oncology, Chapter 3: Tumor cell morphology*. The Publishing House of the Romanian Academy, 2007.

[21] Klaus D. Toennies. *Guide to medical image analysis: methods and algorithms, Chapter 5: Feature detection*. Springer, 2017.

[22] Tuomas Tikkanen, “Cell detection from microscope images using histogram of oriented gradients”. MS Thesis, Tampere University of Technology, Finland, November 2014.

[23] C. Cortes and V. Vapnik, “Support-vector networks”. *Machine Learning*, Springer, vol. 20, pp. 273-297. September 1995.

[24] Google Brain, https://www.tensorflow.org/. A Machine Learning Library, Apache 2.0 open source. November 2015.

[25] S. Wijewickrema and A. P. Paplinski, “Principal component analysis for the approximation of a fruit as an ellipse”. *International Conference in Central Europe on Computer Graphics*, Visualization and Computer Vision, 2005.

[26] Sung Joon Ahn, Wolfgang Rauh and Hans-Jurgen Warnecke, “Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola”. *Pattern Recognition*, Elsevier, vol. 34, Issue 12, pp. 2283−2303, December 2001.

[27] Schanila Nawaz, Paula Sa´nchez, Kai Bodensiek, Sai Li1, Mikael Simons and Iwan A. T. Schaap, “Cell visco-elasticity measured with AFM and optical trapping at sub-micrometer deformations”. *Public Library of Sciences ONE*, vol. 7, Issue 9, September 2012.

[28] Lisa Mae Anders, “Lab on a chip rare cell isolation platform with dielectrophoretic smart sample focusing, automated whole cell tracking analysis script, and a bioinspired on-chip electroactive polymer micropump”. Master’s Thesis, Virginia Tech, 2014.

[29] RajeshKumar, Rajeev Srivastava and Subodh Srivastava, "Detection and classification of cancer from microscopic biopsy images using clinically significant and biologically interpretable features". *Journal of Medical Engineering*, Hindawi Publishing Corporation, July 2015.

[30] Zhen Zhang, Matthew Bedder, Stephen L. Smitha, Dawn Walker, Saqib Shabir and Jennifer Southgate. "Characterization and classification of adherent cells in monolayer culture using automated tracking and evolutionary algorithms". *Biosystems*, Elsevier, pp. 110-121, June 2016.

[31] phenoCHIP LLC. [Online]. Available: https://startupcompete.co/startup-idea/lifesciences/phenochip-technologies-llc/35477

[32] Divya Chandrakant Bala, "Cell phenotype analyzer: automated techniques for cell phenotyping using contactless dielectrophoresis". MS Thesis, Virginia Tech, Spring 2016.

[33] Jaka Cemazar, Temple A. Douglas, Eva M. Schmelz and Rafael V. Davalos, "Enhanced contactless dielectrophoresis enrichment and isolation platform via cell-scale microstructures". *Biomicrofluidics*, February 2016.

[34] Alireza Salmanzadeh, Elizabeth S. Elvington, Paul C. Roberts, Eva M. Schmelz and Rafael V. Davalos, "Sphingolipid metabolites modulate dielectric characteristics of cells in a mouse ovarian cancer progression model". *Integrative Biology*, vol. 5, pp. 837–912, June 2013.

[35] H. A. Pohl. *Dielectrophoresis: The behavior of neutral matter in non-uniform electric fields*. Cambridge University Press, Cambridge, 1978.

[36] Hadi Shafiee, John L. Caldwell, Michael B. Sano and Rafael V. Davalos, "Contactless dielectrophoresis: A new technique for cell manipulation". *Biomedical Devices*, Springer, 2009.

[37] Cynthia Hanson and Elizabeth Vargis, "Alternative cDEP design to facilitate cell isolation for identification by Raman spectroscopy". *Sensors 2017*, Molecular Diversity Preservation International and Multidisciplinary Digital Publishing Institute, February 2017.

[38] Saurav Basua, Soheil Kolouria and Gustavo K. Rohdea, "Detecting and visualizing cell phenotype differences from microscopy images using transport based morphometry". *Proceedings of National Academy of Sciences*, vol. 111, pp. 3448–3453, March 2014.

[39] J. Nascimento and J. Marques, "New performance evaluation metrics for object detection algorithms". *Proceedings of IEEE PETS Workshop*, 2004.

[40] Y. Wu, J. Lim and M. H. Yang, "Online object tracking: A benchmark". *Computer Vision and Pattern Recognition*, pp. 2411–2418, 2013.

[41] Faisal Bashir and Fatih Porikli, "Performance evaluation of object detection and tracking systems". *Computer Vision and Pattern Recognition*, June 2006.

[42] Oliver Otto, Philipp Rosendahl, Alexander Mietke, Stefan Golfier, Christoph Herold, Daniel Klaue, Salvatore Girardo, Stefano Pagliara, Andrew Ekpenyong, Angela Jacobi, Manja Wobus, Nicole Töpfner, Ulrich F Keyser, Jörg Mansfeld, Elisabeth Fischer-Friedrich and Jochen Guck, "Real-time deformability cytometry: On-the-fly cell mechanical phenotyping". *Nature Methods*, pp. 199–202, February 2015.

[43] J. Schindelin, I. Arganda-Carreras and E. Frise, "Fiji: An open-source platform for biological-image analysis", *Nature Methods*, pp. 676-682.

[44] Hu Zhang and Kuo-Kang Liu, “Optical tweezers for single cells”. *J. R. Soc. Interface*, pp. 671–690, 2008.

[45] John Happel and Howard Brenner. *Low Reynolds number hydrodynamics: With special applications to particulate media*. *Springer*, Netherlands, 1983.

[46] Gianni Medoro, Roberto Guerrieri, Claudio Nastruzzi, Roberto Gambari and Nicol Manaresi, “Lab on a chip for live-cell manipulation”, *IEEE Design and Test of Computers*, vol. 24, pp. 26-36, January-February 2007.

[47] A. Castellanos, A. Ramos, A. González, N. G. Green and H. Morgan, “Electrohydrodynamics and dielectrophoresis in microsystems: Scaling laws”. *Journal of Physics D: Applied Physics,* pp. 2584, 2003.

[48] A. S. Bhushan, F. Coppinger, and B. Jalali, “Time-stretched analogue-to-digital conversion," *Electronics Letters*, vol. 34, no. 9, pp. 839–841, April 1998.

[49] Robert Hooke. *De Potentia Restitutiva, or of Spring. Explaining the Power of Springing Bodies*. London, 1678.