Automatic Feature Extraction and Classification of Cell Images for Cytopathology

A thesis submitted
in partial fulfillment for the degree of

Doctor of Philosophy

by

GOPAKUMAR G

Department of Earth and Space Sciences
INDIAN INSTITUTE OF SPACE SCIENCE AND TECHNOLOGY
Thiruvananthapuram - 695547

December 2017

CERTIFICATE

This is to certify that the thesis titled **Automatic Feature Extraction and Classification of Cell Images for Cytopathology**, submitted by **Gopakumar G**, to the Indian Institute of Space Science and Technology, Thiruvananthapuram, for the award of the degree of **Doctor of Philosophy**, is a bona fide record of the research work done by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Gorthi R.K. Sai Subrahmanyam
Supervisor & Associate Professor
Department of Avionics
Indian Institute of Space Science & Technology

Place: Thiruvananthapuram
December 2017

Counter Signature of HOD with Seal

Dr. Anandmayee Tej

DECLARATION

I declare that this thesis titled **Automatic Feature Extraction and Classification of Cell Images for Cytopathology** submitted in fulfilment of the Degree of Doctor of Philosophy is a record of original work carried out by me under the supervision of **Dr. Gorthi R.K. Sai Subrahmanyam**, and has not formed the basis for the award of any degree, diploma, associateship, fellowship or other titles in this or any other Institution or University of higher learning. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

G Gopakumar
SC12D013

Place: Thiruvananthapuram
December 2017

ACKNOWLEDGEMENTS

I offer my salutation to the lineage of Guru Parampara starting with lord Sadasiva and continuing up to all my immediate teachers for constantly showering their grace upon me without which my thesis 'Automatic Feature Extraction and Classification of Cell Images for Cytopathology' would not have materialised. I submit this thesis to my beloved teacher Rajamma for her guidance and support without which I would have been a school drop-out.

I am grateful to Almighty to have my Guide Prof. Gorthi R.K. Sai Subrahmanyam for his academic guidance and emotional support throughout the course.

I sincerely acknowledge the guidance offered by Dr. Sai Siva Gorthi of IISc, Dr Deepak Mishra & Dr. Rama Rao Nidamanuri of IIST. I also acknowledge the support that they have extended to use their lab facilities. I am grateful to all my doctoral committee members Dr. A N Rajagopalan, Dr. N Padmanabhan and Dr. R Krishnan for their valuable suggestions. I sincerely thank Dr. Anandmayee Tej, Head of the Department and Dr. Vinay Kumar Dhadhwal, Director for extending their support and facilities for my research.

I am grateful to my beloved parents for their will to educate their children in the best possible way they can, using their day-to-day small earnings, even though they did not have any formal education. My sincere thanks to my younger brothers who had sacrificed a lot especially during our schooling. I also thank my wife, Divya who had supported me to quickly overcome many hurdles by giving new hope and brought me back to the mainstream.

I, on this occasion, remember the valuable suggestions and support offered by my friends Gangadhar, Jayesh, Kalyan, Rajesh and Swetha at IISc as well as the support from Arun, Bharath, Hari and Rahul at IIST.

ABSTRACT

Cytopathology is the analysis at cellular level for disease diagnosis. Every cell has standard morphology and typical count in unit volume constituting the cell signature. Depending on the pathological state of the individual, the signature may change and is the subject of cytopathology. Manual microscopic examination is the gold standard for cytopathology but is a tedious, skill demanding job and suffers from low throughput. Automated microscopy and more recently imaging flow cytometry (IFC) emerged to overcome these difficulties and to standardise the result. However these systems used extensive robotic handling and/or expensive fluid handling mechanisms, making them bulky, expensive and not suitable for resource limited clinics. In our research, we strive for developing very cost-effective point-of-care diagnostics platforms by using off-the-shelf, low-cost components. However the low-cost instrumentation has introduced great challenges in processing the acquired data such as dealing with the focus shift, unlabeled, unstained data and imaging artefacts. We have overcome these challenges by designing, developing and employing sophisticated image analysis and advanced machine learning algorithms. We have proposed processing frameworks for both microscopy and IFC: a framework to automate malaria diagnosis in microscopy and a general framework for processing and classification of cells in IFC. The frameworks include feasible preprocessing, novel cell segmentations, feature extraction as well as classification. We have explored both the possibility of using conventional classifiers (like support vector machine and nearest neighbour) and trending deep learning based classifiers (based on restricted Boltzmann machine and convolutional neural network) and proposed classification techniques even when the availability of labeled data for training is limited. The feasibility of the IFC framework is established by classifying leukaemia cell-lines (K562, MOLT, and HL60).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

ATCC	American Type Culture Collection
CD	Contrastive Divergence
CNN	Convolutional Neural Network
CTC	Circulating Tumour Cells
DBN	Deep Belief Network
FFN	Feed Forward Neural Network
FNAC	Fine Needle Aspiration Cytology
FoV	Field of View
GLCM	Gray Level Co-occurence Matrix
IFC	Imaging Flow Cytometry
LBP	Local Binary Pattern
Mf-Ms	Microfluidic Microscopy
mIFC	Microfluidics Imaging Flow Cytometry
MRI	Magnetic Resonance Imaging
NA	Numerical Aperture
NN	Nearest Neighbour
PCA	Principal Component Analysis
RBC	Red Blood Cells
RBM	Restricted Boltzmann Machine
ReLU	Rectified Linear Unit
SIFT	Scale Invariant Feature Transform
SML	Soft-Max Log-loss
SVM	Support Vector Machines
WBC	White Blood Cells
MCC	Matthews Correlation Coefficient

CHAPTER 1

Introduction

1.1 Overview of the Research Work

Cytopathology is the study and diagnosis of diseases at the cellular level. The microscopic examination remains as the gold standard for cell analysis due to its low cost and widespread acceptance. However, the manual examination is a laborious task involving both slide preparation (fixation and staining) and analysis. It is a time consuming, repetitive and tedious job. Above all, the results may vary for the same sample among the clinicians depending on their level of expertise. In order to overcome these drawbacks, several efforts have been made in the recent past to automate the process of cytopathology. Research efforts in this direction have mostly been constrained to two approaches: automation of slide preparation and automation of slide analysis. Instruments which can carry-out automated slide preparation employ extensive amount of robotic handling, rendering the commercially available whole slide scanning and analysing systems bulky and expensive (Rojo et al. (2006); Pantanowitz et al. (2011)). One of the cheaply available automated microscopy systems is PathScope (PathScope (2016)) but costs around 25000 US $ making them not that affordable in resource limited clinics especially in low-income group countries.

The automated microscopy systems such as the PathScope (PathScope (2016)) has considerably improved the throughput (number of cells that can be processed in unit time) when compared to manual microscopy but is still well behind flow cytometry. Flow cytometry has become an indispensable tool for clinicians and biologists for counting, analysing and identifying cells with typical throughput of the order of a few thousands cells per second. It uses a flow cell architecture where the cells are interrogated using lasers while they are in flow. A typical flow cytometry system measures the forward as well as side scatter profiles of the lasers. The forward scatter is a measure of the size of the cell and the side scatter is a

measure of the complexity of the cell (refer Fig. 1.1), and this knowledge is used to identify and count different cells under study.

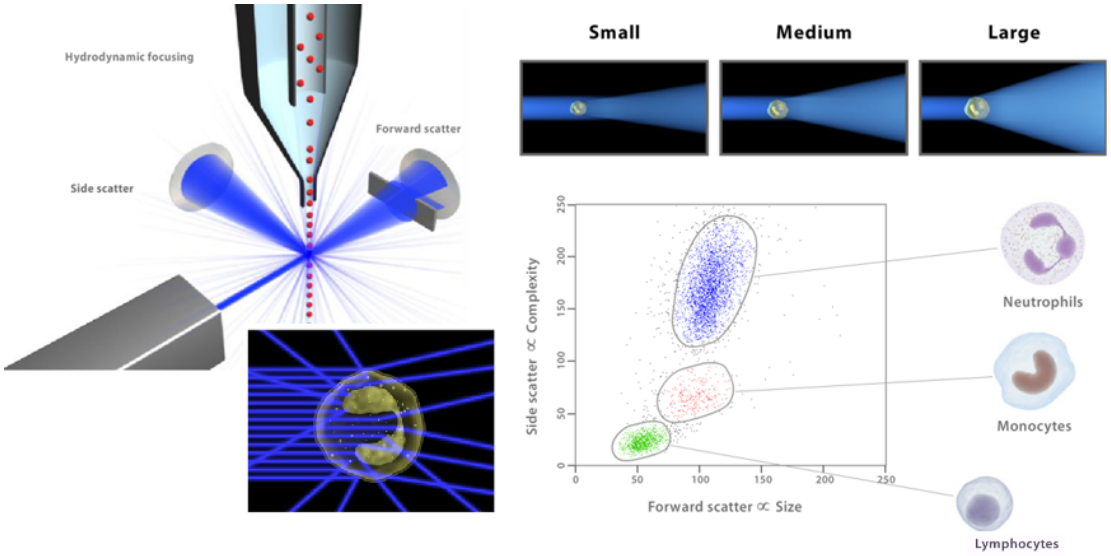

Figure 1.1: Analysis of white blood cells by flow cytometry (Flow (2016))

While the acquisition speed of flow cytometry is extremely high, the amount of information it provides per cell is usually low. The reason is that the flow cytometry will not capture specific morphological features other than the amount of scatter. On the other hand, traditional microscopic examination offers detailed images with spatial localisation of sub-cellular components but has drawbacks in terms of enumeration and speed. Imaging Flow Cytometry (IFC) (Basiji et al. (2007); Schonbrun et al. (2012)) is a nascent technology that combines the speed of flow cytometry and the power of digital microscopy in providing the capability to analyse morphological features. However, the current commercially available IFC systems are bulky and expensive. For example, the Amnis IFC by Merck Millipore Inc (Amn (2016)) costs around 199,000 US $. These systems use bulk fluid handling mechanisms for automating the process of sample image acquisition and employ sophisticated and expensive image acquisition schemes which use time-delay integration detectors and multiple laser sources. The machines are too expensive to afford in resource limited settings. The recent trend has been to employ microfluidic sample handling in combination with different microscopy imaging modalities to enable high-throughput imaging of cells in flow. These systems, which we refer as microfluidics microscopy (Mf-Ms), combine the statistical power of flow cytometry with spatial and quantitative morphology of digital

microscopy.

In this research, we propose processing frameworks for the data acquired from cost-effective whole slide analysing system and a reasonably high-throughput prototype microfluidics microscopy system (Jagannadh et al. (2016)) developed by our collaborators. Unlike the commercially available systems, the developed systems employ inexpensive off-the-shelf optical components and fluid handling mechanisms. This has enabled us to set-up portable, automated disease diagnostic/screening platforms for resource limited settings. Thus my research thesis has dual objectives. i) Analysis of stained blood smear images/videos captured using custom built automated full slide scanner, thereby providing a cost-effective solution for malaria diagnosis in conventional gold-standard microscopy. ii) Analysis and classification of cells from unstained IFC data. The common goal is to design and develop necessary framework containing advanced image analysis and machine learning techniques to operate on the data from very cost-effective instruments thereby moving in a direction to make portable diagnostic/screening systems for resource limited settings.

In this thesis, first we put forward necessary image analysis and classification algorithms for the custom-built automated whole slide scanner for detecting and quantifying textitPlasmodium falciparum infected malarial cases. This is accomplished by cell localisation by a proposed cascaded segmentation strategy and parasite detection using a custom-designed Convolutional Neural Network (CNN) on focus stack of slide images. Subsequent chapters discuss about the data analysis from a prototype Mf-Ms system. As noted, the Mf-Ms systems employ inexpensive optics, as well as polymer/plastic based microfluidic devices (around 1 $). Thus the total cost of the components of the microfluidics IFC system developed (Jagannadh et al. (2016)) is only about 1500 $ when compared to 199,000 $, the cost of Amnis (Amn (2016)) IFC. I have proposed, as part of this research, a general framework capable enough to automatically analyse the cells captured using custom-designed microfluidics microscopy systems and have also built a prototype for signature based as well as hand engineered feature based classification of unstained unlabeled Leukaemia cell-lines K562, MOLT, and HL60.

In the last chapters of this research report, we propose the use of deep learn-

ing based cell classifiers for better accuracy. The proposed framework is capable enough to deal with limited availability of labeled data for building a supervised classification system. The Restricted Boltzmann Machine (RBM) based deep belief network makes use of all available data (both labeled as well as unlabeled) to learn the underlying structure of the training data so that the subsequent supervised training needs only very few training samples for learning the classifier. Also, we propose to use the transfer learning capability of CNN to extract sensible and discriminative features to produce better accuracy for the leukaemia cell-line classification.

1.2 Motivation and Scope

There is a great demand and huge medical value for cost-effective cytopathology. As discussed in last section, the available automated cytopathology systems are bulky and unaffordable to many clinics especially in poor-income group countries. This has motivated us to define the scope of our research and we set our goal to design and develop necessary image analysis and pattern recognition techniques for setting up low-cost, portable instruments for point-of-care diagnostics.

1.3 Contributions of the Thesis

The goal of this research work, as noted in section 1.2, is to propose necessary image analysis and pattern recognition techniques for low-cost automation for cytopathology. Towards this goal, we have made the following contributions through this research.

- A fully automated quantification system for the diagnosis of malaria due to protozoan of type *Plasmodium falciparum* from focus stack of blood smear images collected using a cost-effective, custom-built, portable whole slide scanner is proposed. A custom designed convolutional neural network is used to detect the malaria infected cells. Use of CNN operating on focus stack for the detection of malaria is first of its kind, and it not only improved the detection accuracy both in terms of sensitivity and specificity

but also favoured the processing on cell patches and avoided the need for hand-engineered features. The proposed approach (portable slide scanner and the CNN algorithm together) is suitable for point-of-care Diagnostics and eliminates the need for a pathologist to manually examine the slides using a bright-field microscope.

- The proof-of-concept of a diagnostic framework as well as a prototype for signature based analysis and classification of leukaemia cell lines (K562, MOLT, HL60) imaged using custom fabricated, cost-effective microfluidics imaging flow cytometry (mIFC) is proposed. The mIFC is an emerging technology that combines the statistical power of flow cytometry with spatial and quantitative morphology of digital microscopy. We have analysed the feasibility of a cell signature based approach for cancerous cell identification analogous to face recognition systems implemented with help of surveillance cameras. We have also proposed a way by which cells difficult to classify can be identified, which will open up an opportunity to clinicians to correctly identify the true class of the cell rather than going for a wrong classification. Altogether, such a platform would facilitate affordable mass screening camps in the developing countries and therefore help to decrease cancer mortality rate.

- We Proposed a general framework for the processing/classification of cells in mIFC. The framework includes computationally feasible cell preprocessing methods and a proposed graph based cell segmentation strategy to find the contour of the cell. Once the cells are localised, a set of features reflecting the size, shape, and complexity of the cells are extracted and are used to classify the cells. The usefulness of the framework is established by performing the classification of leukaemia cell-lines. The proposed system is a significant development in the direction of building cell analysis platform that would facilitate affordable mass screening camps looking cellular morphology for disease diagnosis.

- We have also explored and proposed the feasibility of using deep learning networks for cytopathology by performing the classification of leukaemia cell-lines. The capability of Restricted Boltzmann Machine (RBM) based

systems in learning the structure of the data rather than learning labels is utilized to build a Deep Belief Network (DBN) for classification. The transfer learning capability of deeply trained CNN on extensive non-medical image database such as ImageNet is made useful to successfully classify the leukaemia cell-lines K562, MOLT and HL60. These capabilities of both RBM and CNN are very useful in medical image domain where often large dataset is available for training but only a small fraction is labeled, limiting the capability of building supervised deep learning based classifiers. In our investigation, we have found that the proposed methods outperformed the conventional systems in the classification of these cell lines. To the best of our knowledge, such a reporting on cytopathology images is first of its kind and we believe that it holds great promise in terms of enabling cancer screening in resource-poor settings.

1.4 Organisation of the Thesis

Chapter 2 explore the cellular image diagnosis in general and microscopy as well as microfluidic microscopy based cytopathology in particular. Chapter 3 describes the image analysis and classification system proposed for automated quantitative malaria diagnosis, using a custom developed whole slide scanning system. Chapter 4 introduces the nascent imaging flow cytometry and covers the details of the prototype developed for signature based leukaemia cell-line analysis. Chapter 5 provides the general framework proposed for the analysis and subsequent classification of cells captured using the custom-built microfluidic microscopy systems. Chapter 6 proposes the use of deep learning systems for better classification accuracy. It provides necessary details for using RBM and CNN for the use in cytopathology to build deep learning based classification system even with limited labeled training data. Each chapter starts with the goal that it is trying to accomplish, and covers necessary literature, discusses the contributions. Chapter 7 summarizes the thesis and suggests directions for future works.

CHAPTER 2

Cellular Image Analysis for Cytopathology

Cytopathology is the study and diagnosis of diseases at the cellular level. In Cytopathology, often free cells are analysed unlike histopathology where the tissue as a whole get analysed. Each cell under investigation has a signature constituted by the morphology of the cells as well as their behavioural characteristics. The cytologists look for deviation from standard cell signature to report pathological state of the subject. The cells for analysis are either prepared as a smear on a glass slide or as a fluid suspension. Sometimes cyto-centrifugation is used to concentrate the cells under investigation without altering the morphology of the cells.

Often cytopathologic tests are conducted to assist cancer diagnosis/screening. One of the most popular and successful examples is in the screening of cervical cancer where cells are scraped from the cervix and are subjected to analysis using a microscope. Cells from neck, thyroid and breast are also analysed to detect the pathological state of the organ. In such cases the cells are often collected using a needle attached to a syringe and are analysed (fine needle aspiration cytology (FNAC)). Though the primary concern of cytopathology is to assist in the diagnosis of cancer, the techniques are also used to detect microbial infections : viral, bacterial, and/or parasitic infections.

2.1 Blood Cytology

Blood cytology is cytopathology analysis of blood cells for the diagnosis of diseases. A number of diseases can either be diagnosed or screened by analysing blood cells and it includes the parasitic infections like malaria, allergic reactions (eosinophilia) and leukaemia. The important cells in blood are red blood cells (erythrocyte), white blood cells (leukocyte) and platelets (thrombocytes). Each of these cells has dedicated roles : erythrocyte's primary responsibility is transporting oxygen to tissues and taking back carbon dioxide. The leukocytes fights

Table 2.1: Typical Blood Cell Count

Type	Man (in μl)	Woman (in μl)	Child (in μl)
Erythrocyte	4.5 – 5.5 million	3.8 – 4.8 million	3.7 – 5.3 million
Leucocytes	4000 – 10000	4000 – 10000	5500 – 15000
Thrombocytes	0.15 – 0.41 million	0.15 – 0.41 million	0.15 – 0.41 million

Table 2.2: Differential Leucocyte Count (in %)

Neutrophil	Eosinophil	Basophil	Monocyte	Lymphocyte
40 – 80 %	1 – 6 %	<1 – 2 %	2 – 10 %	20 – 40 %

against infections and they include neutrophil, basophil, eosinophils, monocytes and lymphocytes. The monocytes and neutrophils are bacterial phagocytes. Lymphocytes deal with foreign objects such as bacteria, virus and toxic substances by producing antibodies and by triggering antigen-antibody reactions. Eosinophil deals with the allergic reactions while basophils triggers inflammatory reactions.

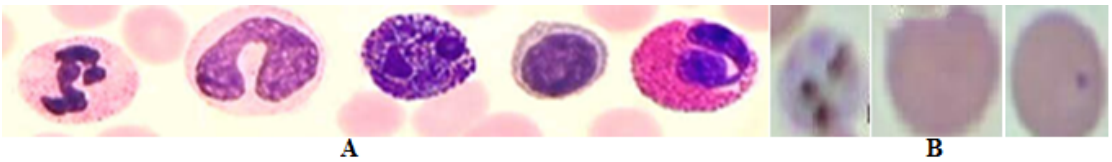

Figure 2.1: Types of blood cells : a) Leucocytes (neutrophil, monocyte, basophil, lymphocyte & eosinophil); b) Erythrocytes

The different blood cells follow standard range in their count in unit volume of blood and hold standard morphological characteristics. For example, one micro litre of blood has around 5 million RBCs and 7000 WBCs in a healthy man and the differential count of WBCs is also in standard range (Table. 2.1). Around 60% of total WBCs are neutrophils and 30% will be lymphocytes (Table. 2.2) (Lewis et al. (2012)). Each of these cells also has standard morphology determined by features such as cell size, shape of the nucleus, amount of cytoplasm, presence of granules and number of nuclear lobes (Fig. 2.1). For example, the lymphocytes are small cells (~$8\mu m$ in diameter) having round nucleus and scarce cytoplasm with no granules, while neutrophils are bigger cells (~$15\mu m$ in diameter) having multi-lobular nucleus and abundant cytoplasm with granules. These

morphological characteristics and standard count in unit volume of blood form their signatures, and any deviation may signal a pathological state. For example, a person with fever and having high neutrophil count suffers most probably a bacterial infection whereas having high lymphocyte count signal viral infection. The morphology of the cells may also undergo a change. For example, the last 3 cells in Fig. 2.1 **b** are red blood cells (RBCs) of which only the middle one is healthy and others are infected by malarial parasite. Healthy RBCs are small cells ($\tilde{}6\mu m$) with rich haemoglobin content, and it can be noted that the characteristics of the infected cell are quite different.

2.2 Microscopy

The most popular cytopathology examination is using brightfield microscope. For brightfield microscopic examination, specimen is prepared by smearing samples on a glass slide and is followed by fixation and staining. The fixation hardens the cells on the glass surface but keeps them in a 'life-like' state. This is achieved by the use of a fixation agent (e.g. formaldehyde) that kills micro-organisms and prevents enzyme activities. Sufficient contrast is then introduced between the components of interest (nucleus, cytoplasm) and the background by a process called staining. Well established fixation and staining (Clark et al. (1981)) protocols are used for the purpose.

The microscopic examination of cells is cost effective and it opens up the opportunity to closely inspect cellular details including morphology. Thus it became the gold standard in cytopathology. However, the manual microscopic examination is a quite labour involved task. The process of slide preparation requires skill and it typically takes about 30 minutes. Similarly the staining also takes about 30 minutes to few hours depending on the cells being analysed and stain being used (Bancroft (2008)). Finally, the analysis of statistically significant number of cells requires a few hours turning the manual microscopic examination tedious, time demanding, repetitive. The results are also highly dependent on the level of expertise of the cytopathologist and hence can lead to non-standard results across clinicians even for the same sample.

In general, the process of cytological investigation is quite intricate and time taking due to the manual processing involved at various stages of the test. In the case of manual slide examination, the slide has to be manually moved using a mechanical translation stage to visually assess different regions of the slide for the presence of cells with abnormal morphology. This would involve several focus and magnification readjustments, making the overall process quite tedious and time-taking and it also reduces the throughput. The natural choice to overcome these drawbacks is automated microscopy.

2.3 Automated Microscopy

Research efforts in automating microscopy have mostly been constrained to two approaches: automation of slide preparation and automation of slide analysis. As noted earlier in section 2.2, slide preparation involves both fixation and staining and are quite involved process. A number of automated slide preparation systems are proposed (Oud et al. (1986); Zahniser and Hurley (1996)) and are commercially available (de Bitencourt et al. (2013); Tabe et al. (2014); PathScope (2016)). However they employ extensive amount of robotic handling rendering them bulky, expensive and inaccessible to resource limited settings.

Sophisticated image analysis algorithms are used to analyse the captured slide images and the analysis is often performed only on the best focused image. In order to find the best focused plane, a number of auto focus measures are available (Firestone et al. (1991); Hamm et al. (2010); Junior et al. (2010)) and are often built-in to the hardware (He et al. (2010)). However, the images acquired in automated microscopy tend to have cells very closely spaced (or overlapping) increasing the computational complexity of image processing algorithms, essential for extracting signature of different cells. Therefore the image analysis algorithms must include preprocessing stages to improve contrast and to suppress noise, segmentation techniques to localise each cell, feature extraction techniques to extract signature specific features followed by an inference engine to make the decisions. The inference engine often makes use of machine learning techniques to accomplish the intended task. Computerized methods have been rapidly evolving to

assist pathologists by automating the segmentation of cells (and its components such as nuclei (Di Cataldo et al. (2008)) and cytoplasm), feature extraction and classification (Thiran and Macq (1996)). A good review on each of these techniques can be found in Irshad et al. (2014).

The automated microscopy has considerably improved the throughput when compared to manual microscopy. However, the research efforts to facilitate wider deployment of cytopathology have mostly been constrained to automation with very little focus on cost optimization. While robotic sample handling has been used to automate slide preparation, image processing algorithms have been used to automate/semi-automate slide analysis (Walts and Thomas (2002)). Despite recent efforts on automation, slide based cytopathology fails to reach out to resource-poor settings. This is mainly due to the trade-off that exists between automation and system cost.

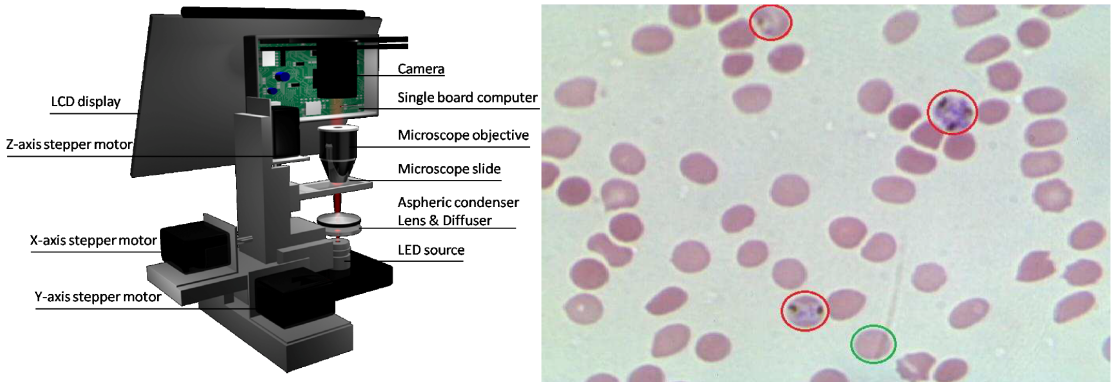

Figure 2.2: Schematic of the whole slide scanner and the best focused image for an arbitrary FoV

There is huge interest for automated microscopy for low-resource settings which is essential for operability without the need for skilled labour and equipped lab resources. Low-cost automation can be achieved by employing off-the-shelf, inexpensive components and the schematic of one such system developed by our collaborators is shown in Fig. 2.2. The overall cost of the components used was less than $1500 making it an affordable platform for disease screening or diagnosis. However the low-cost instrumentation has introduced great challenges in terms of processing. For example, due to the vibrations of the motors, the focus may shift across the field-of-views (FoV) and the image at right focus for each FoV has to be decided in software (by analysing focus measure for each frame). The best focused

image for a FoV showing malaria infected RBCs can be found in Fig. 2.2. Note that, all parasites (infected RBCs are encircled in Red) may not appear in single best focused image and depending on the stage of the life cycle of the parasite, they may appear differently. Also, there can be staining and imaging artefacts. For example, an artefact is encircled in green. This artefact is due to a tissue strand from the paper tissue used to clean camera sensor. These facts make it necessary to analyse a number of images at different focus level to decide between a real parasite and an artefact. Along with parasite detection, cell counting is also need to be addressed to produce quantitative malaria diagnosis. This necessitates the need for cell segmentation. With the knowledge of the parasite locations (from detection) together with the exact boundary of the cells (from segmentation), we can determine the number of infected and healthy RBCs which in turn lead to quantitative result. The cell segmentation becomes a difficult task particularly when there are overlapping cells, staining and imaging artefacts. These issues have to be dealt and we will discuss them in detail in chapter 3.

In fact, the necessity for the costlier robotics based automation is a consequence of slide-based sample handling architecture inherent in conventional imaging methods. Whereas there exist other flow based sample handling such as flow cytometry (Flow (2016)) which completely avoids the slide based imaging and offers higher-throughput compared to automated microscopy. High throughput is desirable as it offers the capacity to analyse statistically significant number of cells in reasonable amount of time before making any inference.

2.4 Flow Cytometry

Flow cytometry is a powerful technique for the analysis of individual cells within a heterogeneous population to statistically report cellular characteristics such as size, complexity, and phenotype. Figure 2.3 shows the important components of a typical flow cytometry system. It includes the fluidic system which presents samples to the interrogation point, the lasers which is the light source for the scatter and fluorescence, the optics which gather and directs the light, the detectors which receive the light, and the electronics and peripheral computer for converting

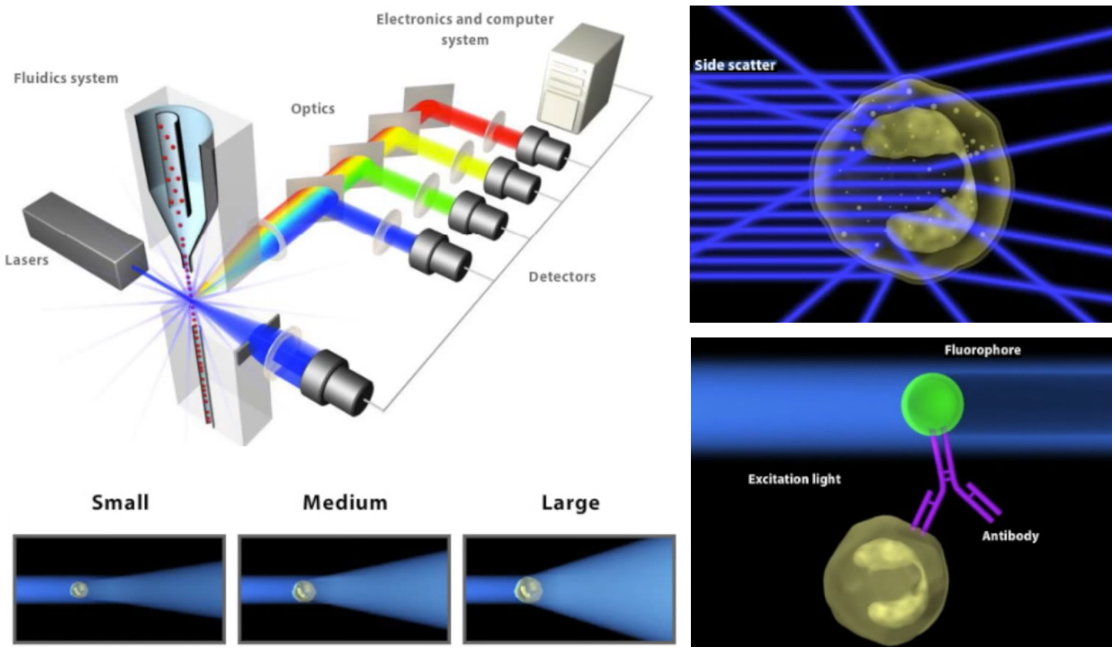

Figure 2.3: Schematic of flow cytometry system (Flow (2016))

the signal from the detectors into digital data and performing necessary analysis.

The cells are injected into a flowing fluid (sheath fluid/saline water) and ensures a laminar flow. The fluidic system takes the sample across the interrogation point one cell at a time. This is achieved by compressing the sample stream to roughly one cell in diameter (hydrodynamic focussing). The cells are interrogated by the laser and as it passes through the cell, the light undergoes scatter. The width of the forward scatter is a measure of size of the cell; the wider the scatter the bigger the cell. Similarly the side scatter is a measure of cell complexity as the side scatter is caused by granularity and structural complexity inside the cell. The side scatter light is guided and is often collected by placing detectors placed right angle to the laser path. The forward scatter as well as the side scatter thus can be used to study heterogeneous population of cells and is usually done using 2 dimensional dot plots. For example, in the dot plot shown in Fig. 1.1, each dot corresponding to the high forward scatter and high side scatter will be a neutrophil which has the highest complexity and larger size among the WBC populations. Similarly, the lymphocytes will be having low forward and side scatter profiles. In addition to making use of forward as well as side scatter, flow cytometry systems make use of fluorescence to get cell structure and functional characteristics of the cells. One of the most common ways to study phenotype characteristics of a cell

is, use fluorophore labeled antibody along with the cell samples. The antibody then binds to a specific molecule of the cell (Fig. 2.3). When interrogated using lasers, the fluorophore gets excited to a high energy state and then returns back to its ground state after releasing the energy. The energy of the emitted light depends on the energy level to which the fluorophore was excited, and hence the emitted light will be having a particular dominant wavelength which determines the colour of the emitted light. The fluorescent light also takes the path of the side scatter, and by placing a number of mirrors and lenses the emitted lights can be captured by the appropriate detector. Thus the mirror and lens assembly do a kind of spectral decomposition to guide the right wavelength (due to side scatter and fluorescence) to the appropriate detector. Once the forward, side scatter as well as fluorescence profiles are captured, these are used to study the size, complexity and phenotype of the cells with in a heterogeneous cell population. The basic principle and clinical applications of flow cytometry can be found in Laerum and Farsund (1981); Brown and Wittwer (2000); Henel and Schmitz (2015); Adan et al. (2017).

With flow cytometry thousands of cells per second can be analysed improving the throughput considerably when compared to automated microscopy. But it lacks in providing the capability to offer morphological analysis on the cells, as it only collects the scatter/fluorescence profiles of the cells. With the advent of advanced imaging techniques and sophisticated image analysing platforms, new flow based imaging came into practice; the most popular is imaging flow cytometry.

2.5 Imaging Flow Cytometry

Imaging Flow Cytometry (IFC) is a nascent technology that combines the statistical power of flow cytometry and the quantitative morphology of digital microscopy. Unlike flow cytometry, in IFC cells are imaged at higher rate and at good resolution so as to capture the morphological features while they are in flow. The cells are flown one at a time across the interrogation point and hence the cells are isolated (rarely clumps). This reduces the computational complexity of analysing the cells as the segmentation is relatively easier compared to slide based imaging system. One of the popular IFC systems available in market is Amnis ImageStream

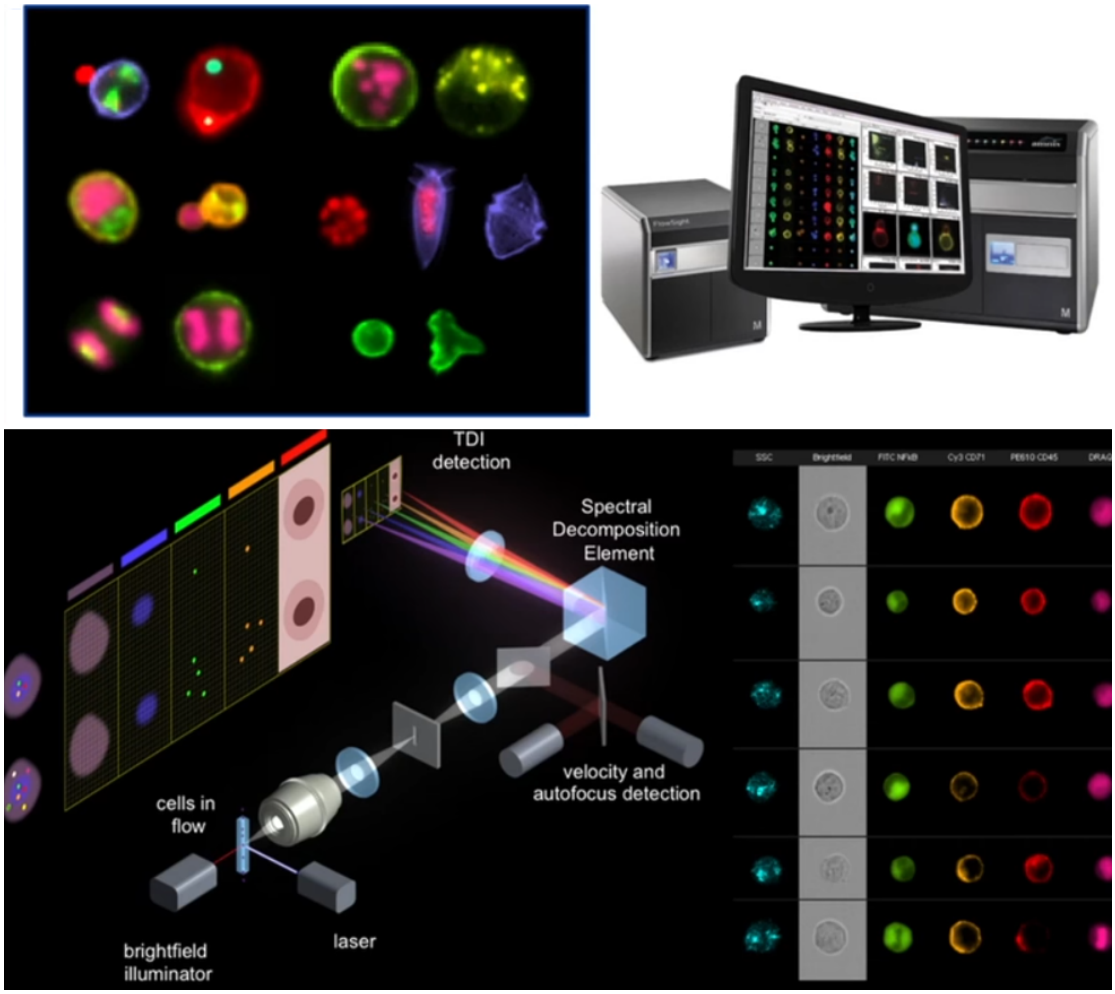

Figure 2.4: Schematic of imaging flow cytometry system (Amn (2016))

(Amn (2016)). It can produce bright-field, dark-field as well as several fluorescent images at high speed. In dark field images, the direct beam is blocked by the aperture while one or more diffracted beams are allowed to pass the objective aperture. These diffracted beams must have strongly interacted with the specimen, and produces useful information about the intra-cellular complexity. The system shown in Fig. 2.4 can get 6 images per cell. Just like the traditional flow cytometer, the cells are introduced into the flow and are interrogated using both bright field and laser illumination. The only difference is in the detector which is CCD camera in IFC. The fluorescence and the bright field imagery are then captured using the objective lens and collimated to the spectral decomposition element. The spectral decomposition element is a series of long pass filters (analogous to electronic low pass filters) set at precise angle such that the spectra is spread out so that when sends to the CCD camera the light is captured at discrete location in the camera. This helps to take different wavelength in different channels. The 6 channels shown

in Fig. 2.4 respectively hold the side scatter dark image, the bright field image, the NF-kB (Nuclear Factor Kappa B) image, orange and red wavelength surface stain images, and the nucleus image in pink wavelength. The NF-KB image can provide useful information as it image the changes due to NF-kB pathway which regulates many cellular processes including proliferation and apoptosis.

Imaging flow cytometry (Basiji et al. (2007); Schonbrun et al. (2013); Gorthi et al. (2013)) recently became the state-of-the-art technology for accurate cellular phenotyping and real-time image analysis (Ng et al. (2015, 2016); Otto et al. (2015b); Yan et al. (2016)). However, these systems employ bulk fluid handling mechanisms for automating the process of sample image acquisition. Further, these systems employ sophisticated and expensive image acquisition schemes, which employ Time-delay integration detectors and multiple lasers. Thus the available IFC systems (~199,000 US $ for Amnis IFC (Amn (2016))) are bulky and expensive and hence are not suitable for many resource limited clinics.

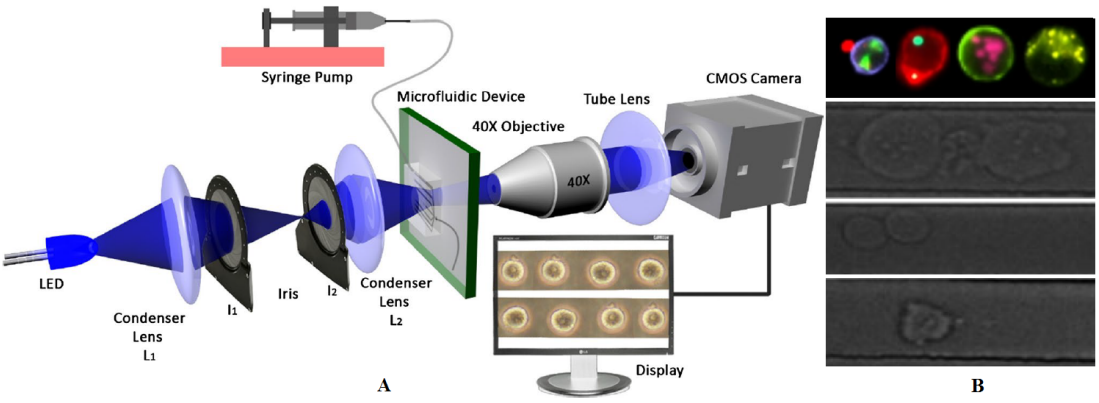

Figure 2.5: a) Schematic of developed mIFC system; b) Labelled and unlabeled cell images

More recently microfluidic sample handling in combination with microscopy became the practice; offering flow based imaging at high-throughput and at low-cost. We refer these systems as microfluidic microscopy and have employed one such system with inexpensive components developed by our collaborators (Jagannadh et al. (2016)) to facilitate cost-effective point-of-care diagnostics. The schematic of the system is shown in Fig. 2.5 **a**. The developed system employs microfluidic conveyor belts (channels) to enable automated sample handling. This has significantly lowered the setup cost compared to commercially available IFCs, which

employ an expensive image acquisition scheme and bulk fluid handling mechanisms. The overall cost of the components of the system developed is around 1500 US $ making it a good choice for resource poor settings. In the context of developing diagnostic system for poor resource settings, the most important parameter for feasible deployment is the cost per test. While regular cytological testing requires the usage of large sample and lower dye volumes, the presented system uses microfluidic devices for sample handling, and thereby requires significantly fewer sample and dye volumes, which brings down the overall cost per test. Performing the test a single time would require simply running the sample through the microfluidic device and analysing the acquired images. Thus, the cost per test would include the costs of microliter volumes of reagents and a replaceable plastic chip. One more factor affecting the cost per test is the usage of fluorochrome for labelling. We can highlight the components of the cell and study phenotype by selectively labeling using fluorochrome as shown in first row of Fig. 2.5 **b**. This makes the processing of images simpler when compared to processing the gray scale unlabeled images shown in subsequent rows. However the fluorescent labelling is costly and adds up the cost per test. There was no existing published framework for processing unlabeled unstained images in mIFC and we have proposed the complete framework which includes preprocessing, segmentation, feature extraction and classification algorithms. We will discuss this framework in chapter 5.

2.6 Summary

The gold standard for cytopathology is microscopy due to low-cost and wide acceptance. However, it suffers low-throughput and the process is tedious, repetitive and time demanding. Above all, the results vary depending on the level of expertise of the analyst. The natural choice to overcome these difficulties is automation. However the automated systems available in the market are bulky and expensive, limiting their use in resource limited settings. This has generated huge interest for automated affordable point-of-care diagnostics systems. We have come up with sophisticated image analysis and machine learning frameworks for recently developed low-cost microscopy/mIFC systems and will be discussed in this thesis.

CHAPTER 3

Focus Stack based Quantitative Malaria Diagnosis for Automated Microscopy

This chapter discusses a focus stacking based approach for automated quantitative detection of *falciparum* malaria from blood smear. For the detection, a custom designed convolutional neural network (CNN) operating on focus stack of images is used. The cell counting problem is addressed as the segmentation problem and we propose a cascaded segmentation strategy. Use of CNN operating on focus stack for the detection of malaria is first of its kind, and improved the detection accuracy (both in terms of sensitivity (97.06%) and specificity (98.50%)). This approach also favoured the processing on cell patches and avoided the need for hand-engineered features. The slide images are acquired with a custom-built portable slide scanner made from low-cost, off-the-shelf components and is suitable for point-of-care diagnostics. The proposed approach of employing sophisticated algorithmic processing together with inexpensive instrumentation can potentially enable disease diagnosis without manual intervention at the site of testing.

3.1 Introduction

Malaria is a deadly infectious disease transmitted by female Anopheles mosquitoes. In 2015, 214 million malaria cases were reported worldwide causing an estimated death toll of 438,000. Five Malarial species of the protozoan of genus *Plasmodium* (W.H.O. (2010)) (*falciparum, vivax, ovale, malariae, and knowlesi*) infect humans. Among those five, *Plasmodium falciparum* is the most common species infecting humans (˜75%) and accounts for the majority of deaths. The next major share (˜20%) of infection is by *vivax* while the *knowlesi* rarely infects humans. Typically the diagnosis of malaria is done by inspecting the Giemsa/Leishman stained blood smears under a bright field microscope. Though there are other methods (Zimmerman and Howes (2015)) such as antigen-based rapid diagnostic tests and

the use of the polymerase chain reaction to detect the parasite's DNA, microscopic examination remains as the gold standard for the diagnosis of malaria due to the low cost and widespread acceptance. However, the manual examination of the slides is cumbersome and demands highly experienced clinicians. Often the result varies from clinician to clinician, and produces non-standardized results. Thus, it is highly beneficial and can add great diagnostic value to have an automated system to aid the clinician in efficient detection of malaria. Further, the development of an affordable and automated diagnostic platform can minimise dependency on the expert for conducting these tests and facilitate in performing these diagnostic tests in resource limited conditions. This has generated huge interest in affordable point-of-care (Zhu et al. (2013)) diagnostic systems.

In the direction of developing automated system for point-of-care malaria diagnosis, quite a large number of works have been done both from thick and thin smear slide images(thick: Elter et al. (2011); Purnama et al. (2013); Pinkaew et al. (2015))(thin: Ravendran et al. (2015); Makkapati and Rao (2009); Mehrjou et al. (2013); Purwar et al. (2011)). A good review on the computer vision based procedures towards malaria diagnosis till 2009 can be found in Tek et al. (2009). The research work proposed in Elter et al. (2011) uses a two stage algorithm for the automatic detection of the parasite in thick blood films. The sensitivity of the system is maximised in the first stage and the cells identified as infected are further processed in the second stage to reduce false positives. There are studies that use statistical features such as mean, variance, kurtosis at the suspected location followed by classification based on genetic programming (Purnama et al. (2013)) and SVM (Pinkaew et al. (2015); Savkare and Narote (2015); Preedanan et al. (2016)). The intensity level differences as well as features extracted from morphology of the parasites are studied for its capability in detecting the presence of parasites in thin films (Anggraini et al. (2011); Kareem et al. (2012); Zou et al. (2010)). Das et al. (2013) uses SVM and Bayesian learning on statistical and textural features to detect 3 important life cycle stages of parasitic infection (*P. falciparum* and *P. vivax*). The work done in Nugroho et al. (2015) extracts histogram based texture features and uses neural network to classify the different stages of parasite infection while Linder et al. (2014) uses SVM classifier (Boser et al. (1992)) on scale invariant feature transform (SIFT) (Lowe (2004)). These

methods operate on different dataset (differ in stain & dataset size) and report sensitivity in the range 81.7% to 95% and specificity in the range 92.59% to 100%. For example, the work in Makkapati and Rao (2009) reports sensitivity of 83% and specificity of 98% on a small dataset of 55 slide images that used Leishman stain.

In this work, we address the detection as well as counting of plasmodium falciparum infected RBCs from Leishman stained microscope slide images. The choice of Leishman's stain is motivated from the recent study in Sathpathi et al. (2014) which compared the use of Leishman and Giemsa stains for malaria diagnosis and suggested Leishman as a good alternative. A comparative evaluation of conventional staining methods and immunological techniques for the diagnosis of malaria can be found in Samir et al. (2013). Though there are many attempts to automate the malaria diagnosis, almost all of them inspect only the single best focused image to identify the infected cells. This has not only increased the cost of the device as they employ sophisticated methods to generate the in-focus image but also introduced the possibility of misclassifying dark dust particles on image as parasite (due to dust on camera/relay lens). Also, depending on the life-cycle stages, the parasites appear differently on slides and is often hard to get all of them in a single focus. We address these problems by capturing focus stack of images. The additional information being acquired as part of the focal stack is not being discarded but being made useful in the subsequent image processing. By avoiding the sophisticated methods to generate the in-focus image, we reduced the cost of the setup and by making use of focus stack in subsequent processing, we have increased the overall detection efficiency. Recently our collaborators have brought the cost of the instrument further down by making use of off-the-shelf cost-effective components. As of now, the least price of an automated slide scanner (PathScope (2016)) is $25,000 (US), whereas the bill of materials for the instrument employed here is as low as $1500. For the image analysis, CNN that operates directly on focus stack of images is employed to identify malarial infection. We compare the results of detection of malaria infected RBCs in terms of sensitivity and specificity obtained by three classifiers; 1) a support vector machine classifier trained on the statistical and textural features extracted from the suspected parasite locations (similar to Pinkaew et al. (2015); Savkare and Narote (2015); Preedanan et al.

(2016)), and CNN classifiers trained on 2) the 32×32 patches surrounding the suspected locations of the best focused image and 3) the focus stack of 32×32 patches surrounding the suspected locations. We have shown that the system operating on focus stack produces the best results both in terms of sensitivity and specificity and hence support the design of our custom built focus stack collecting portable microscope.

The main contributions of this work are 1) the proposal of cost effective malaria diagnosis using custom-built portable focus stack collecting slide scanner, 2) the method of identifying and processing only the suspected parasite locations instead of looking for entire slide thus saving valuable computing power, 3) the use of CNN rather than using the conventional classifier such as SVM to exploit the ability of the classifier to operate directly on the focus stack 4) the proposal of using focus stack of image patches for malaria detection instead of a single focused image and thereby reporting better accuracy,and 5) the automated cell counter making use of a proposed erythrocyte cell segmentation strategy.

This chapter is organized such that section 3.2 gives an overview of the framework, section 3.3 discusses the detection of infected locations, section 3.4 addresses the automated counting procedure. The quantitative analysis is provided in section 3.5 followed by the publications in section 3.6.

3.2 Overview of the Framework

This section discusses the prototype slide scanner used to collect the dataset and outline the proposed method for detecting and counting infected RBCs. The graphical abstract of the proposed system is shown in Fig. 3.1. As discussed in last section, a custom designed CNN is used to analyse focus stack of images for each field of view to identify the parasite locations. Rather than doing an exhaustive search by CNN at all locations in the image for parasite, we only look at suspected locations. Since parasite locations appear dark in Leishman stained image, the suspected parasite locations are identified as regional minima of intensities in the best focused image. A focus stack of patch around each suspected location is then fed to the CNN to confirm whether it is really a parasite location or is due to any

imaging artefact (improper stain or a dust on the sensor). A dust on the sensor appears almost same in images across the focus stack whereas a parasite will come into focus and then fades away. Thus a CNN operating on the focus stack can differentiate between these. The cells are segmented from the best focused image by a proposed cascaded segmentation strategy and it follows a parallel path (Fig. 3.1). With the knowledge of the parasite locations (from CNN) together with the exact boundary of the cells (from segmentation), we can determine the number of infected and healthy cells which in turn produce a quantitative result. Next subsection discusses the instrumentation details which is followed by an overview of the processing framework.

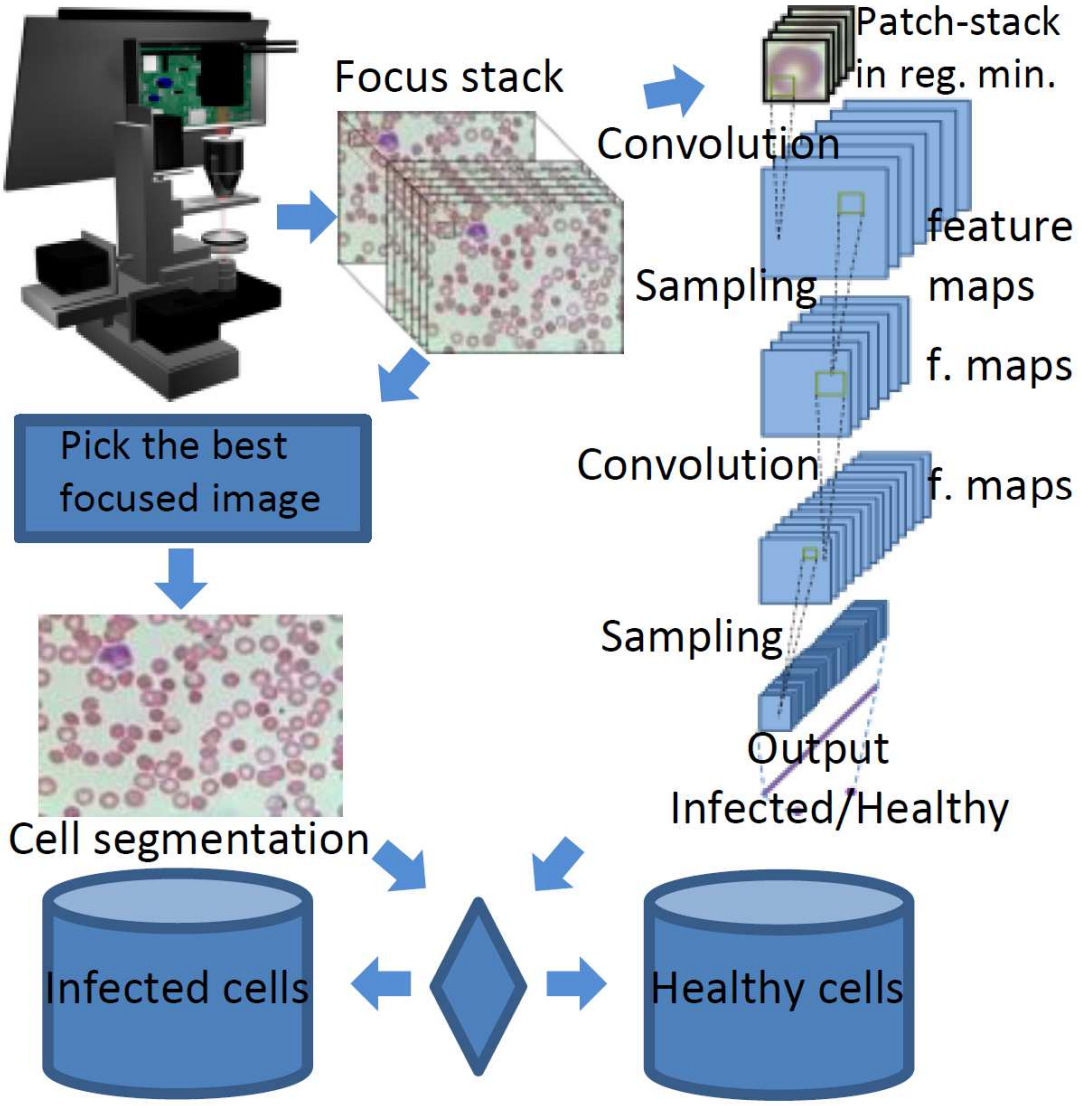

Figure 3.1: Graphical abstract of the framework

3.2.1 Experimental setup and Dataset generation

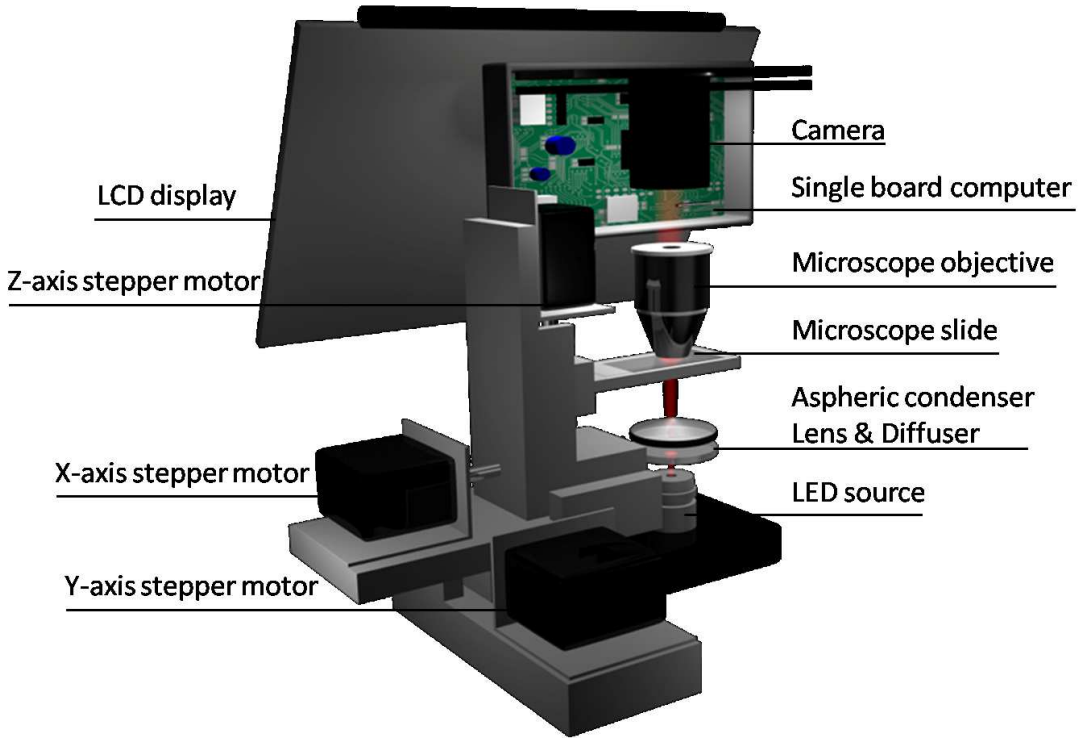

Figure 3.2: Schematic representation of the prototype slide scanner

The dataset used in this study are videos containing focus stack of multiple Field-of-Views (FoV) of Leishman stained slide images prepared using WBCs spiked cultured malarial samples. The *P. falciparum* malaria culture is maintained in 5% hematocrit with O^+ red blood cells containing Roswell Park Memorial Institute (RPMI) media. For preparation of blood mimicking the patient sample, the malaria culture is then spiked with very small amount of WBCs extracted by partial centrifugation of blood sample. About 10 μL of the resultant sample is pipetted onto a clean glass slides (base slides) and spread using a wedge slide, held at an angle of 45^0 (approximately) with the base slide. The wedge slide is moved horizontally over the base slide to result in a thin smear of the resultant sample. The prepared smears are exposed to 5% Leishman stain for 10 minutes for fixing and staining the cells. The stained slides are mounted with cover glass after DePeX addition for proper preservation. The slides are then sequentially imaged under the imaging setup.

A custom built focus-stack collecting bright-field transmission microscope setup built with inexpensive, off-the-shelf optical components and a camera unit is em-

ployed to capture the dataset for the present study. The objective behind realization of such a slide scanner is to devise an affordable and reliable cyto-diagnostic platform that can provide focus stack and function as an automated alternative to traditional microscopic tests for malaria diagnosis in resource limited settings. The optical setup of the prototype slide scanner unit consists of a white light LED source and necessary collimating lens arrangement for uniform illumination of the sample plane and subsequent acquisition of digital images of the slide. A low-cost 40X objective lens (Lawrence and Mayo) is employed to magnify the sample features. For a digital imaging system like ours, the characteristic of interest is the digital resolution which is interplay of the magnification of the objective lens used in the setup and the pixel size of the camera sensor used for imaging. For clear identification of sample features from the respective FoV images, the system should have good digital resolution. With the given setup, our objective is to cover a larger FoV without compromising the digital resolution of the setup. To meet these requirements, a digital colour camera (DFK22BUC03, Imaging source) with a very fine pixel size of 6 μm and a relay lens unit is employed to capture the videos/images of the slide. This has enabled us to get very good results in terms of the quality of recorded images even at a lower magnification of 40X which sufficed the diagnostic requirement. A custom-built 3-D printed motorized translational stage integral to the developed prototype is employed to navigate the slide across multiple FoVs. The motorized stage holds the sample slide in place and translates the slide independently along x, y and z directions. The stage was built with the assembly of three readily available low-cost linear actuator stepper motor units (Nema 11). Three separate digital motor driver units (DM422C) supply the required driving current to the respective motors. Each of the three axes of the motorized translational stage can be independently controlled, enabling the whole slide scanning. Appropriate electrical signals to enable the motors and to control the speed & direction of the motors are supplied by two off-the-shelf micro-controller boards (Arduino UNO). The imaging source camera was plugged into an on-board processor (Intel NUC) to record the videos. The on-board processor also communicates with the micro-controller boards through the Python Firmata serial communication link to facilitate the software control of the motors. An LCD display unit with touch-screen is attached to the prototype slide scanner for user

interaction and display purpose.

Figure 3.2 shows the schematic representation of the developed prototype slide scanner unit. All major electronic components of the prototype slide scanner including the on-board processor, display and motor drivers are powered using a regulated switching power supply with multiple voltage tappings (RT-125D, MeanWell). The micro-controller boards are powered from USB ports of the Intel NUC board. Motor control and simultaneous data acquisition are carried out by executing Python (version 2.9) code on the on-board processor. The custom-built prototype slide scanner employs z-stacking/scanning approach to acquire the whole slide images and employs passive focusing mechanism to determine the in-focus images from multiple FoVs. The Z stacks are recorded after translating the slide by appropriately actuating the lateral direction motors. In the process of z-stacking, the z-axis motor is translated in steps along the direction of optical axis, and the focus stack videos are recorded during the entire span. The z-stacking approach adopted in the prototype offers multiple advantages. Since a passive focusing mechanism is employed, the system relies on commonly used opto-mechanical components and does not require any high precision piezo positioning units, thus reducing the overall system cost to less than 1500 $ (US). Further additional information derived from the focus stacks can be used for examination of very fine sample features which may not be evident from the analysis of just the best focus image and for differentiating imaging artefact (small dark spots in images due to dust particles on camera sensor/relay lens) from similar sample features, thereby minimising the cases of misclassification. Further, focal stacking is essential while examining thicker sections of the slides.

We have used 765 FoVs containing 62015 cells of which 1191 cells are infected. The ground-truth was determined after consultation with experts working in the field after inspecting the focus stack of images which helped them to clearly differentiate the infected locations from other artefacts. The FoVs for our experiments are selected from 2 slides. For each FoV of a slide, the z-axis motor is used to translate in steps across the best focused frame, capturing multiple images. In this way, all valid FoVs of the slide are captured. We use variance as the focus measure to direct the z-axis motor and ensure that we always move across the best focused frame. In an ideal setup, once the best focused frame location is fixed using z-axis

motor, taking fixed number of frames on both direction about it and doing it for all FoVs will suffice. But in our setup, due to vibrations of the motors, there was small shift in the best focused location across the movement of different motors. We address this problem by readjusting (automatically) between successive focus scan by keeping track of the z-motor position that has produced the best focused frame in the last scan. We can also use the same focus measure to decide whether to image a FoV (i.e., whether it contains sufficient cells to image). For the present study, image variance computed from the gray scale equivalent image is employed as the focus measure to identify the best focus image in a given focus stack. The variance computation can be expressed as follows (Eq. (3.1)).

$$F = \frac{1}{M * N} \sum_{i=1}^{M} \sum_{j=1}^{N} (G(i,j) - \mu)^2 \tag{3.1}$$

Here M and N are the number of rows and columns of the frame ($M = 480$, $N = 720$), and μ is the average intensity of the gray level image G, defined by $\mu = \frac{1}{M*N} \sum_{i=1}^{M} \sum_{j=1}^{N} G(i,j)$.

Fig. 3.3 plots the focus measure computed from each of the 13 adjacent FoVs collected during a horizontal scan. As expected, for each FoV stack, the focus measure monotonically increases and attains maximum value for the frame in the best focus and then it decreases. In Fig. 3.3, the best focused image for each FoV is marked in red asterisks and the one marked in green corresponds to either the last frame of a FoV or the first frame of a new FoV.

Fig. 3.4 shows the best focused image from a FoV stack. It contains four infected RBCs (encircled in Red) and an imaging artefact on a RBC due to dust on the camera sensor (encircled in Black) and a WBC (encircled in Blue).

3.2.2 Framework

The flowchart for the detection of candidate parasite locations using a classifier can be found in Fig. 3.5. The regional minima locations are identified as the candidate locations and a 32×32 patch surrounding it is used to represent each such location. Depending on the classifier, the patches as such or features extracted

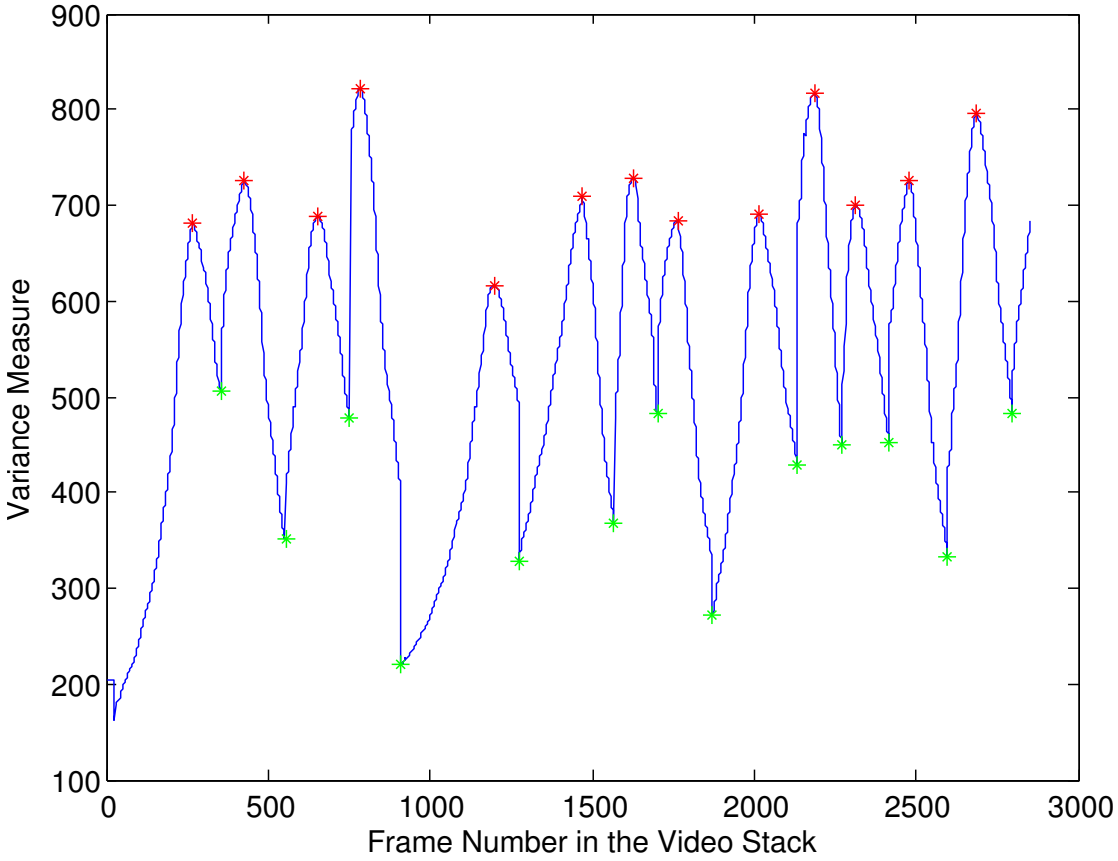

Figure 3.3: Plot showing image variance (Y Axis) across 13 focus stacks (the best focused image for each FoV is marked in red).

from the patches are used to take appropriate decision to classify the candidate location as infected or healthy. We have experimented with three classifiers: SVM classifier on hand engineered features, a CNN classifier operating only on the best focused patch and a CNN classifier operating on the focus stack of patches and have found that the CNN working on the focus stack gave the best performance. The detailed discussion can be found in section 3.3.

Section 3.4 discusses the problem of segmenting the cells towards taking the count of infected RBCs. In the segmentation procedure, we make use of typical size of an RBC to determine the number of cells in the slide particularly when there are clumps of cells. Since the size of a WBC is typically bigger than that of an RBC, we separate WBCs first, before deciding the number of RBCs by subsequent segmentation. Once the infected locations are identified and the cells are appropriately segmented out, the counting of infected RBCs is trivial. The flowchart of the procedure is shown in Fig. 3.6.

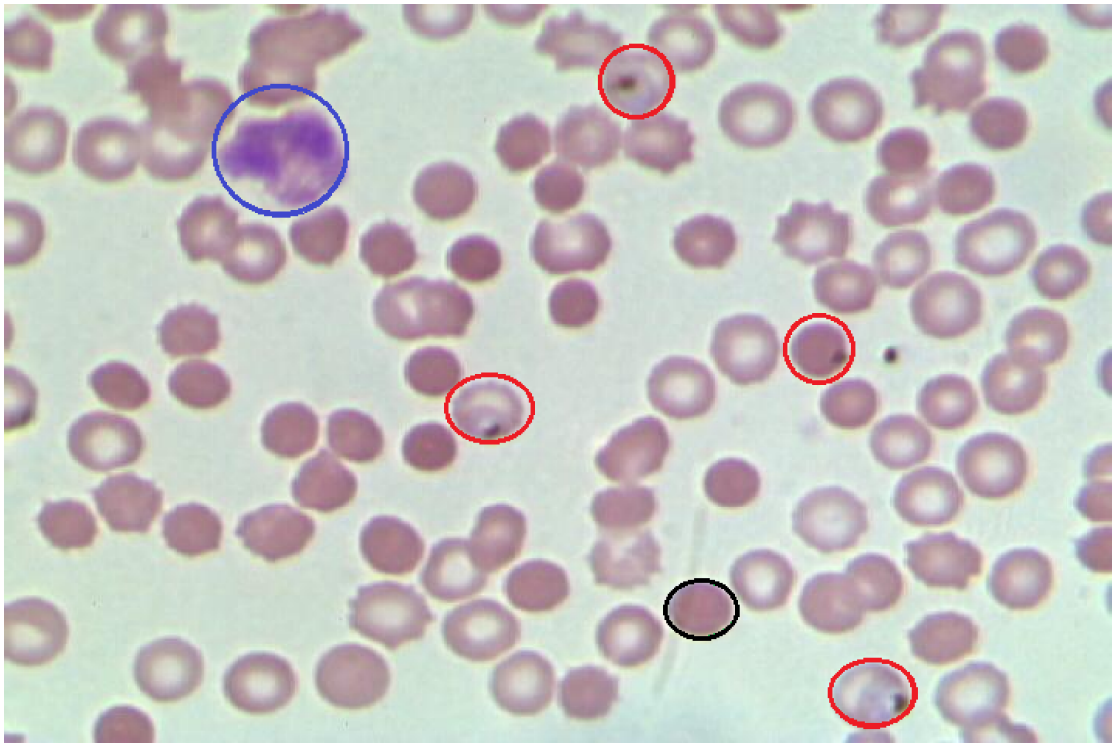

Figure 3.4: Best focused image from a stack. Infected RBCs are encircled in Red, an RBC with an artefact is encircled in Black and a WBC is encircled in Blue.

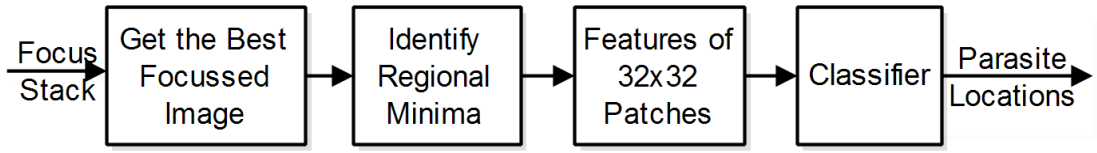

Figure 3.5: The flowchart depicting detection of parasite locations

3.3 Proposed Methodology : Detecting infected locations

In this section, we discuss the proposed method for detecting parasite locations. The idea is to classify identified candidate locations into either infected or healthy. We experiment with traditional way of extracting features and feeding it to SVM classifier. Also, we experiment the use of trending deep learning network: CNN. Two CNNs are designed for this; the first operates on patches from the best focused image of a FoV while the second operate on focus stack of patches.

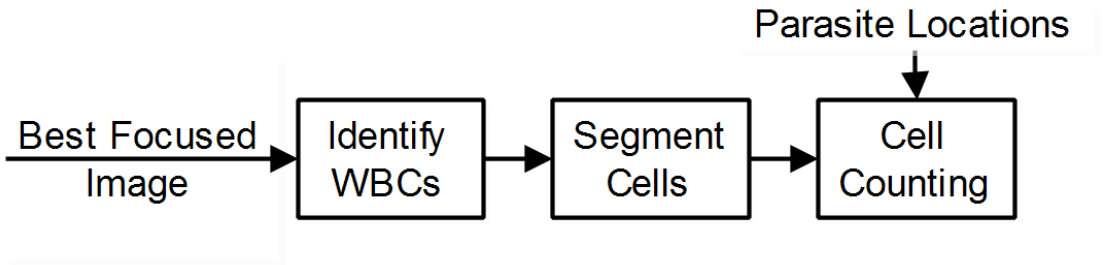

Figure 3.6: The flowchart depicting cell counting

3.3.1 Identifying candidate locations

The candidate locations are identified from the best focused image. The best focused image for each FoV is identified as the one with the highest focus measure (variance). The behaviour of focus measure across focus stack of images for 13 FoVs can be found in Fig. 3.3 and the best focused image for an arbitrary FoV is shown in Fig. 3.4. Once the best focused image is retrieved, the next step is to detect candidate locations. The parasites appear darker in slide images and hence suspected candidate locations of the parasite can be identified as the local minima intensity region of the image over a neighbourhood spanning the radius of typical RBC size. We can refine these candidate locations by excluding a few of them which are falling in the background region. The background image for this purpose can easily be found by Otsu's global thresholding (Otsu (1979)) since there is high contrast between cell region and background. Fig. 3.7 shows the suspected parasite locations identified (dilated version of centroids of the regional minima (Breen and Jones (1996); Soille (2003))) and are superimposed on the original image shown in Fig. 3.4. Though the darkness of parasite may slightly change across staining, we are looking only for the regional minima to identify the candidate locations (irrespective of the level of darkness). Thus every parasite location will be identified as a candidate location and in our experiments, the method has identified all such locations. When there is noise, its location also qualifies as candidate location for parasite, but will be taken care of by the classifier since noisy locations have different profile across the stack when compared to parasite locations.

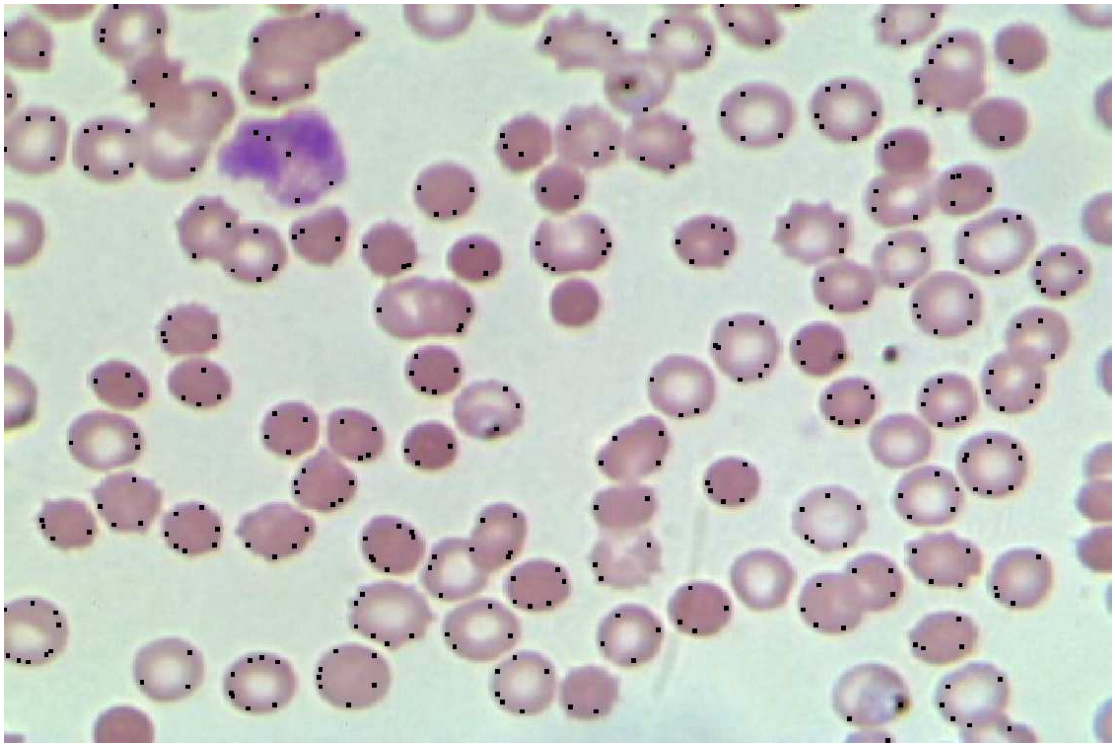

Figure 3.7: The centroid of the regional minima superimposed on the image shown in Fig. 3.4

3.3.2 Detecting infected locations

Once the candidate locations are determined, the next step is to identify the locations which are really infected by parasites. We treat this as a binary classification problem where we need to classify each of the candidate locations into classes either infected or healthy. We experiment both with traditional SVM classifier trained on hand engineered statistical as well as textural features and the trending CNN deep learning classifier. There were a total of 1400 positive patches which are the patches around ground truth parasite locations. Note that, a cell may have more than one parasite locations marked on it. There were 326934 negative patches which are the patches surrounding local intensity minima where there is no parasite. As the number of positive candidate patches is very less compared to the number of negative patches, we have rotated the patches at 90^0, 180^0 and 270^0 and increased the number of positive samples to 5600.

Classification by SVM

Each of the candidate location identified is represented by selecting 32×32 RGB patch surrounding it in the best focused image. Unlike using statistical features alone for the SVM based classification (Pinkaew et al. (2015); Savkare and Narote (2015); Preedanan et al. (2016)), we propose to use texture features as well. Altogether, fourteen features are considered. At parasite locations, there will be gradient difference, and the surrounding textural characteristics differ when compared to a healthy patch. Widely used features to characterize the texture of a patch are 'Contrast', 'Correlation', 'Energy' and 'Homogeneity'. These features are extracted from the gray level co-occurrence matrix (GLCM) of the region as defined in Gopakumar et al. (2016). The mean and standard deviation (Std) of these features computed for healthy and infected patches are shown in Table 3.1.

The parasite locations in a Lieshman's stained slide image appears to be darker and hence such locations can be characterized by identifying statistical features such as the minimum, maximum, mean and variance of intensities of the patch as well as the minimum and maximum gradient magnitude observed in 32×32 region. Table 3.2 shows the statics of these features in order.

In addition to global statistics, we have considered local statistics as parasite infections are often very local, especially in case of early stage of infection. Towards this, we have considered small (3×3) non-overlapping sub-regions for the 32×32 patch and have computed the minimum and maximum values of mean and variance observed for all sub-regions. The statistics of these features computed over all positive and negative patches are shown in Table 3.3.

Thus there are 4 texture features, 4 statistical features computed by taking all pixel intensities of the patch and 2 features computed from gradient of the patch, followed by 4 features computed by considering all 3×3 non-overlapping sub-regions. These constitute set of 14 features and their statistics are provided in Tables 3.1 through 3.3. The statistics reveal that the infected patches have low intensity, high variance and high gradient magnitude when compared to healthy patches.

The SVM classifier with radial basis function (RBF, $\sigma = 0.7$) kernel is then

Table 3.1: Mean (Std) of GLCM Features for 32×32 Patches

	Contrast	Correlation	Energy	Homogeneity
Healthy	0.5698 (0.2335)	0.9825 (0.0094)	0.1020 (0.0445)	0.8189 (0.0455)
Infected	0.8850 (0.3439)	0.9754 (0.0119)	0.0584 (0.0357)	0.7577 (0.0498)

Table 3.2: Mean (Std) of Global Statistical Features for 32×32 Patches

	Min_Int	Max_Int	Mean_Int	Var_Int	Min_GMag	Max_GMag
Healthy	0.5023 (0.0615)	0.8891 (0.0345)	0.6797 (0.0534)	0.0168 (0.0069)	0.0007 (0.0019)	0.6426 (0.1303)
Infected	0.2720 (0.1057)	0.8748 (0.0453)	0.6109 (0.0641)	0.0195 (0.0090)	0.0028 (0.0034)	0.8545 (0.2147)

trained by taking features of training patches. The kernel parameter σ is fixed after experimenting with a range of values between 0 and 1 in steps of 0.1.

Classification by CNN

In addition to SVM classifier on hand-engineered features, we have used custom-built CNN for detecting the infected locations. One of the designed CNNs, directly operates on the RGB candidate patches selected from the best focused image while the other operate on the focus stack of patches. We compare the advantage of using the focus stack over the best focused image for their capability in identifying the infected cells.

The CNN is a biologically inspired feed-forward multi-layer artificial neural network mapping an input vector X into an output vector Z. The connectivity pattern between its neurons is inspired by the organization of the animal visual cortex so as to respond to overlapping regions tiling the visual field. Internally it

Table 3.3: Mean (Std) of Local (3×3) Mean and Variance of Patches

	Min_Mean	Max_Mean	Min_Var	Max_Var
Healthy	0.5095 (0.0620)	0.8817 (0.0337)	$3.4520e^{-6}$ ($2.7481e^{-6}$)	0.0061 (0.0160)
Infected	0.2903 (0.1049)	0.8669 (0.0444)	$7.2392e^{-6}$ ($1.2817e^{-5}$)	0.0092 (0.0137)

can be thought of as a composition of functions each implementing simple convolutions on input feature map using learned kernels interleaved with non-linear and pooling operations followed by locally or fully connected layers (LeCun and Bengio (1998)). With the advent of high-end computing capability, CNN has recently become the de-facto standard for classification and has provided reliable classification in medical domain. CNNs are successfully used in detecting micro calcification on mammograms (Lo et al. (1995)), classifying interstitial lung diseases (Li et al. (2014)), detecting pathologic cases in chest Xray (Bar et al. (2015)) and detecting lung nodules in chest radiographs (Lo et al. (1995)) and for detecting mitosis in breast histology images (Cireşan et al. (2013)). Recently, Liang et al. (2016) have studied the capability of CNN (both transfer learning capability as well as stand alone classification capability) in deciding samples whether they are malaria infected or not. The dataset that they have used contains 27578 RBCs from Giemsa stained slide images. They have used a 16 layer architecture and reports that CNN as a stand alone classifier has produced better classification result (mean accuracy 97.37%) over the transfer learning based classification (91.99%). In our experiments, we are using a relatively smaller architecture (9 layers) and a bigger dataset.

The basic building blocks of a CNN are

Convolution block : Given a multi dimensional image X (D image instances), the convolution block finds Y (K instances) by performing the convolution between the image and a set of learned kernels f and by selecting only the valid part of the convolution. Equation 3.2 depicts this operation where $f^{\boldsymbol{T}}$ represents the flipped kernel.

$$X \in \mathbb{R}^{H \times W \times D}, f \in \mathbb{R}^{H^{'} \times W^{'} \times D \times K}, Y \in \mathbb{R}^{H^{''} \times W^{''} \times K} \tag{3.2}$$

$$y_{i''j''k} = b_k + \sum_{i^{'}=1}^{H^{'}} \sum_{j^{'}=1}^{W^{'}} \sum_{d^{'}=1}^{D} f_{i^{'}j^{'}d^{'}k}^{\boldsymbol{T}} \times x_{i''+i^{'}-1,j''+j^{'}-1,d^{'}}$$

Where $b_k \in \mathbb{R}$ are biases for the nodes in the layer.

Note that the weights learned in convolutional block of the CNN forms the kernel and biases. Intuitively, the operation on the convolution block using the learned

kernel on an input image can be thought of extracting very local background and foreground features such as (but not exactly) low pass and high pass filters, edges (just like operating a Sobel kernel on the image (Sobel (2014))) and corners. Thus we are actually learning the local feature extractors (not the actual features) just like contrast quantifiers, edge, line, and corner detectors which when operated on the test images extracts the needed features.

ReLU activation function : Given an input y_{ijk}, the ReLU suppresses the value if it is negative. Thus the operation performs like a non-linear activation function and it follows the convolution operation. Without ReLU, the non-linear activation function, the whole network would have been reduced to a simple linear transformation. Also, unlike other non-linear activation functions such as sigmoid, ReLU offers better resistance to slow learning (due to vanishing of gradients (Rohan (2016))) especially at lower layers of deep networks.

$$y^1_{ijk} = \max(0, y_{ijk}) \tag{3.3}$$

Max pooling : The max pooling typically follows the ReLU operation and it produces the maximum response of each feature channel in a $H^{'} \times W^{'}$ neighbourhood patch. Thus, this layer introduces small shift invariance as well as scale invariance to the features. This is because the subsequent convolutions operate on a scaled down version of the input.

$$y^2_{ijk} = \underset{1 \leqslant i^{'} \leqslant H^{'}, 1 \leqslant j^{'} \leqslant W^{'}}{max} \left(y^1_{i+i^{'}-1, j+j^{'}-1, k} \right) \tag{3.4}$$

3.3.3 Detecting infected locations

CNN opens up a good space for analysing the focus stack particularly in malaria diagnosis, as it has the capability to directly deal with multi-channel images. The basic design of the CNN used in our experiment is shown in Fig. 3.8. Here, input is a D channel image patch of size 32×32. The C, R, P in the block represents Convolution, ReLU, and Pooling operations respectively. The size of the

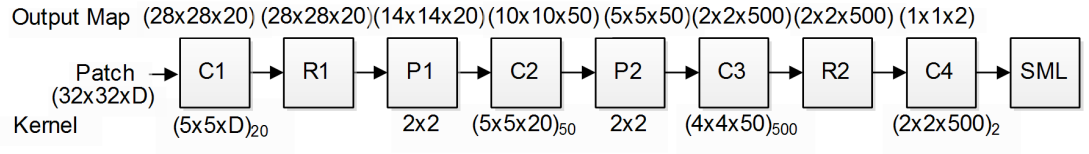

Figure 3.8: The CNN designed for malaria detection

kernels used for the convolution operation is also shown under each block where the subscript shows the number of kernels used. The size of the output map computed is provided above the respective block. All the pooling blocks do max pooling in 2×2 area and use a stride of 2. We have chosen this architecture based on the following observations. As we are dealing with patches around suspected parasite locations, and being the typical cell size 41×41, the input to the CNN is decided as 32×32 since it is decent enough to hold the neighbourhood in making the decision. We have used the standard CNN building blocks: convolution, ReLU and sub-sampling. However, sub-sampling by max pooling is preferred over average pooling. This is because, in the case of early infection, the parasite may confine to very small area probably looking like a dark spot, and we don't want to average out the details. Towards deciding size of feature detectors, we have considered the size of kernels used in many of the well-established CNN models like imageNet pre-trained model (Deng et al. (2009)), VGGNet (Simonyan and Zisserman (2015)), resNet (He et al. (2016)), googleNet (Szegedy et al. (2015)) and are found to be from the set $\{3, 5, 7, 11\}$. A lower size is preferable in order to have good sensitivity for the detector even for parasite in its early stage of infection, where parasitic infections often appear just like a black spot particularly when observed using 40X objective (used in our experiments). At the same time, there can be isolated pixel noises in slide images masquerading as infection. Thus we decide the kernel size not too small and not too large and is decided as 5×5. Being a binary classification, the desired number of output neurons was set to 2. The output layer producing the feature on which the decision is made is set as the convolution layer and is motivated from one of the most successful CNN ImageNet classification models (discussed in section 6.4). Then we have selected the intermediate layers and the parameters. Thus the architecture contains 4 feature extraction (convolution) layers. These make us to avoid pooling after R_2 and C_4 as they are already at the

lowest possible dimension on the feature level to have the final 2 element decision vector. However, the non-linearity of the feature from layer C_2 is ensured through max pooling.

As noted, for every input patch, the CNN is designed to produces a binary vector at the output (C4 in Fig. 3.8). Towards this goal, the CNN is trained using back-propagation algorithm, considering the log loss (Eq. 3.5) error function. The log loss J_i for an i^{th} input patch is defined as

$$J_i = -\sum_{k=1}^{C} L_{ik} \log P_{ik} \tag{3.5}$$

In Eq. 3.5, C is the number of output classes (number of output neurons) which is 2 in our case and L_i is the boolean vector with value 1 only at the true class location for the input patch; i.e., output label vector is $[1, 0]$ for an infected patch and $[0, 1]$ for a healthy patch. P_{ik} is the model probability of assigning label k to the i^{th} input instance. A perfect classifier should have a log loss of precisely zero.

Eq. 3.5 can also be written as

$$J_i = -\log P_{iT} \tag{3.6}$$

Where P_{iT} is the model probability of assigning label T (the true class label) to the i^{th} input instance. We have considered T as 1 for infected patches and 2 for healthy patches. The model probability P_{iT} can be calculated using soft-max function and the final function that we want to minimise turns out to be the soft-max log loss (Eq. 3.7).

$$\begin{aligned} J_i &= -\log\left(\frac{\exp^{Z_{iT}}}{\sum_{t=1}^{C}\exp^{Z_{it}}}\right) \\ &= -Z_{iT} + \log\sum_{t=1}^{C}\exp^{Z_{it}} \end{aligned} \tag{3.7}$$

Note that for each of the class there is one neuron in the output layer and Z_{iT}

in Eq. 3.7 is the response produced at the final layer neuron corresponding to the true class T for the specific input at the input layer. We want to minimise this function as much as possible. However, there is an instability in evaluating this expression for any input. It is due to the difficulty in evaluating the second term, the log sum of exponentials. If any of the value Z_{it} becomes sufficiently large, its exponential becomes very large and the sum can over flow to positive infinity. Similar is the case, if any of the Z_{it} value becomes sufficiently smaller, the log sum of exponentials can underflow. However, we can get rid of this problem with little algebraic manipulation, if we take $\Psi = \max(Z_{it})_{t=1}^{C}$.

$$\begin{aligned}
\log \sum_{t=1}^{C} \exp^{Z_{it}} &= \log \left(\sum_{t=1}^{C} \frac{\exp^{\Psi}}{\exp^{\Psi}} \exp^{Z_{it}} \right) \\
&= \log \left(\exp^{\Psi} \sum_{t=1}^{C} \exp^{Z_{it}-\Psi} \right) \\
&= \Psi + \log \sum_{t=1}^{C} \exp^{Z_{it}-\Psi} \qquad (3.8)
\end{aligned}$$

Thus Eq. 3.7 and Eq. 3.8 together defines soft-max log-loss (SML) function as

$$J_i = -Z_{iT} + \Psi + \log \sum_{t=1}^{C} \exp^{Z_{it}-\Psi} \qquad (3.9)$$

Eq. 3.9 ensures that the largest value passed to the exponential function is 0. If there are really tiny values after subtracting Ψ, they will become 0 and will be dropped out as they should be with limited precision arithmetic. This soft-max log loss has a simple derivative . From Eq. 3.7

$$\frac{dJ_i}{dZ_{it}} = -\left(\delta_{t=T} - \frac{\exp^{Z_{iT}}}{\sum_{t=1}^{C} \exp^{Z_{it}}} \right) \qquad (3.10)$$

Here $\delta_{t=T}$ is a vector which has value 1 only at the true class location Z_{iT} and everywhere else it is 0. As noted the CNN is learned by backpropagation algorithm, in which the error at the final layer is computed for each input image considering its target class. Then this error is propagated down the layers. Whenever it crosses ReLU layer, the derivative of ReLU (i.e., for positive values 1, else 0) is

multiplied, and whenever it crosses the pooling layers, the error matrix gets up-sampled, and across the convolutional layers it is proportionately multiplied by the corresponding weight contributions from the kernel. Then the parameters to be learned (the kernel weights and biases) are then updated by gradient descent, for which the gradient is computed by multiplying the error and the input to the connection for which the parameter is computed. The explicit equations and the derivations for the weight update can be found in Vedaldi and Lenc (2014). Also refer Appendix A for a general discussion on CNN learning.

During testing, SML layer is excluded and the label of the class that yields maximum response is assigned to the test sample. We have used the CNN building blocks developed for Matlab (Vedaldi and Lenc (2014)) to design our network.

CNN on the Best Focused Patches

For each of the selected candidate parasite locations for training, 32×32 RGB patch is extracted from the best focused frame surrounding the suspected location. These are then used to train the CNN classifier shown in Fig. 3.8. Note that the dimension of input (D) is 3 since we are using only one RGB patch for each candidate location.

CNN on 32×32 focus stack patches

In this section, we will start with the motivation behind using focus stack in recognizing parasite locations. Fig. 3.9 shows 4 focus stacks (one in each row), where the 5^{th} image is from the best focused frame. The remaining 4 images each on both sides are the images after skipping 4 frames away from the best focused frame on both sides of the stack. These are actually tiled cell images (each of size 51×51) cropped selectively from frames and are then scaled to fit width of the manuscript. The cell in first row is healthy while the others are infected by parasite. Note that, in the infected cases, the parasites are coming into focus and then fades away unlike the dust on healthy cell which is having almost constant appearance across the stack. Though there is a variation in case of parasite infection across focus stack, the change is minimal especially in case of

early stage of infection. Our intention is to capture these changes with minimum number of images (to restrict computational complexity) and we chose a patch from the best focused frame, and two patches far away from the best focused, one on both sides. We have chosen 32×32 RGB patches for training, since this is decent enough to capture the neighbourhood to decide the infection, being 41×41 the typical cell size. Thus, each focus stack taken to train the CNN will now contain 3 RGB patches ($32 \times 32 \times 9$). A few of the focus stacks (3 in each row) of such patches used for training CNN are provided in Fig. 3.10. Cells shown in first row are really infected while in second and third rows are healthy. Note that, the cells in second row have artefact due to dust and have almost the same appearance unlike the parasite in first row which has change in appearance across the stack. Thus, the sets that we have selected capture the change profile of the parasite/artefact across the stack and value the main motive of the experiment: checking whether focus stack improves the detection accuracy in malarial cases. Thus, in the third experiment, a CNN is trained using these focus stack of patches, in the same way using the same candidate locations used to train the CNN on the best focused patches.

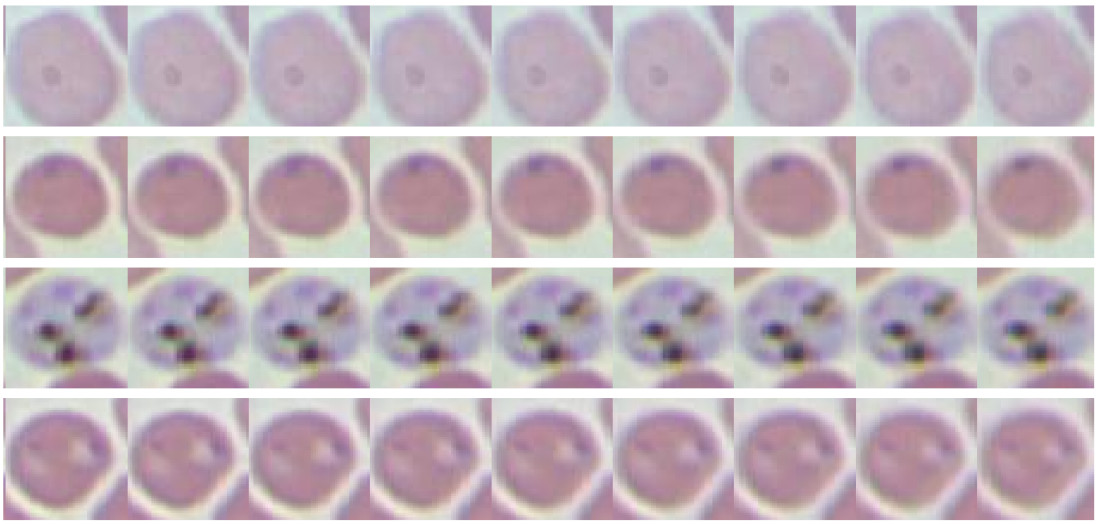

Figure 3.9: Focus stacks of 4 cells each containing 9 images (along columns): Last three stacks are that of infected cells while the first row represents a healthy cell with a dust on it

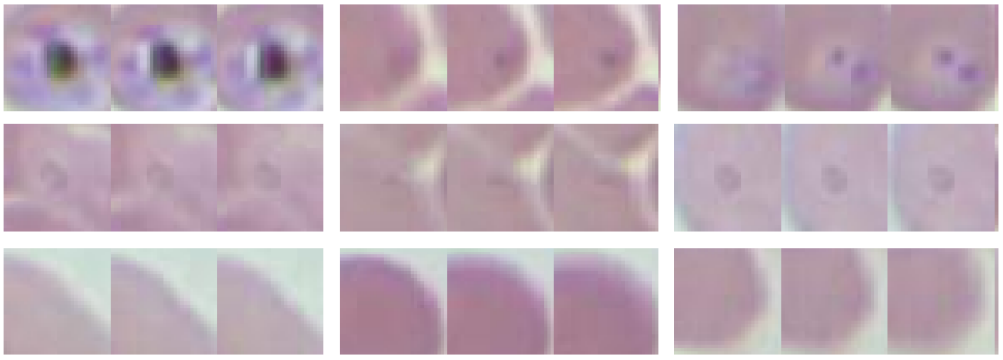

Figure 3.10: Sample focus stacks (3 in each row) used in training CNN: Cells in first row are infected while in second and third rows are healthy. Note that cells in second row has artefact due to dust and have almost the same appearance across the stack.

3.4 Automatic segmentation and counting of infected RBCs

The detection of malaria infected RBCs is a sub-problem to the more general problem of determining the parasitemia level. In order to count number of infected RBCs, each cell has to be separately identified. We address this problem by automatic segmentation of cells and is going to be discussed in this section.

3.4.1 Segmentation procedure

The proposed segmentation is a cascaded two step procedure : segmenting the cells which are more or less separated and segmenting the cells from clumps. We employ an adaptive thresholding strategy to segment cells from the background (Fig. 3.11) followed by marker based watershed segmentation for cells that forms clump (Fig. 3.12).

Segmenting the cells from the background

In order to segment the more or less separated cells, a combination of adaptive as well as global threshold based strategy is proposed. Once the clear cases are

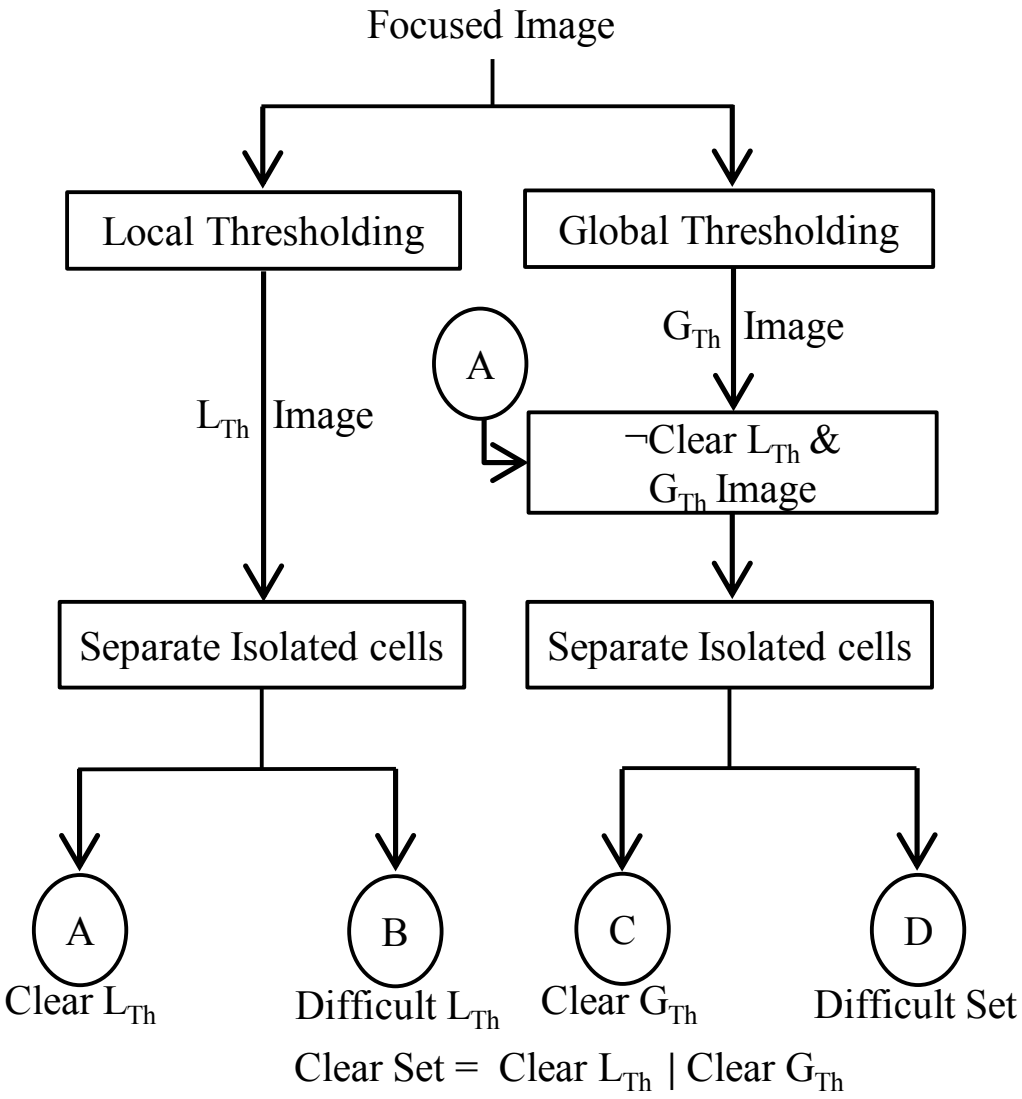

Figure 3.11: Procedure to segment more or less separated cells

segmented from background, the next level of processing is to separate the cells from clumps, if any. The main steps involved are shown as a flowchart in Fig. 3.11. The local thresholding schemes segment cells based on local statistics and hence better separability can be achieved especially at slowly varying region between cells; with global threshold, this would have formed a cell clump. However, local threshold can produce many holes/breaks in cells especially if the cell area is of constant intensity. This would not happen in case of segmentation by global thresholding as there is high inter class variability between cell region and background. We make use of this complementary nature to have a good segmentation. This is accomplished by first identifying isolated cells in local thresholded image

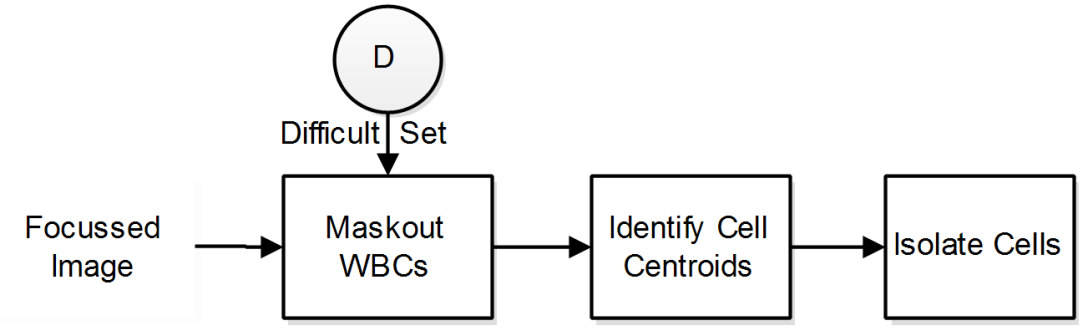

Figure 3.12: Procedure to segment cells from clumps

(Clear L_{Th}) and then masking them in the global thresholded image (¬Clear L_{Th} & G_{Th} Image) and finally looking for good cells (Clear G_{Th}) in the remaining segments. Once isolated cells are identified from local and global thresholded images, the objects left in global thresholded image (the Difficult Set in Fig. 3.11) are recognized to contain multiple cells (as cell clumps) and are processed separately (Fig. 3.12). Following subsections explain these procedures in detail.

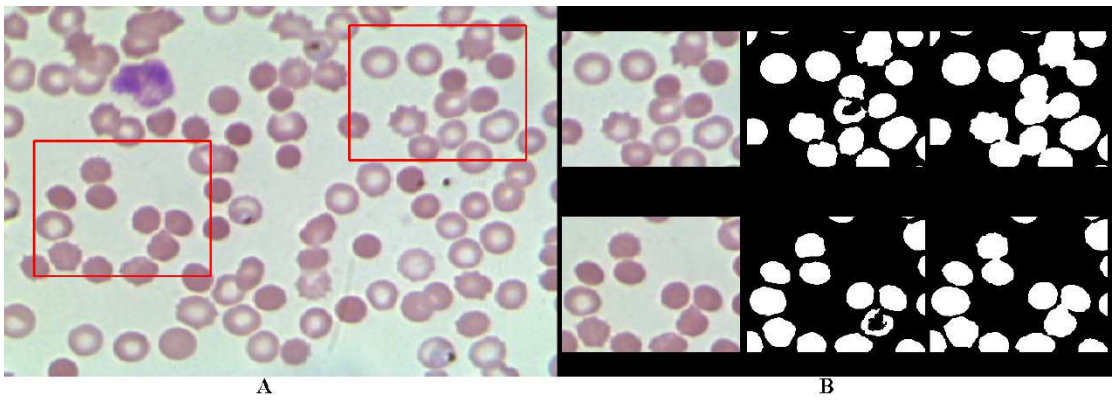

Figure 3.13: Comparing local adaptive and global thresholding: A) the best focused image B) the region of interest, results for local adaptive and global thresholding respectively.

Local Adaptive Thresholding A Leishman stained slide image appears in good contrast and cell regions appear totally different in colour when compared to background. Rather than using a single global threshold to segment all cells, we find an adaptive threshold at each pixel position considering the pixel's neighbourhood (21×21) intensity values. Then we assign a pixel to the cell area if its intensity is still lower than this threshold. We define a threshold little (ϵ) lower than the Gaussian weighted neighbourhood pixel intensities so that smooth regions

inside the cell are rightly assigned to the cell area and the background region as background. It also helps to produce a good dark threshold for the pixels on cell wall and around the cell wall. We could better separate the cells by local adaptive threshold when compared to the segmentation obtained by using a single global threshold. This is highlighted in Fig. 3.13 **b** by taking two sub-regions of interest. The segmentation result for the full image is provided in top row of Fig. 3.14 and are respectively, the local and global thresholded images. The size of averaging kernel for picking local threshold is set to 21×21 which provided the best result, when experimented with a range of sizes in between 3 and 27 and for the same reason ϵ is set to 0.005.

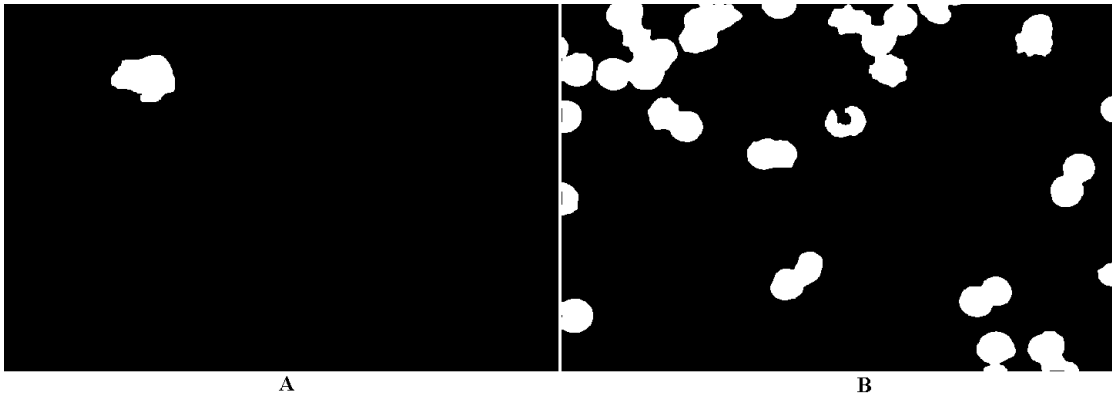

Figure 3.14: Segmentation by local adaptive as well as global thresholds: A) local adaptive thresholding B) global (Otsu's) thresholding C) the well separated cells segmented out D) cells to be segmented from clumps

Global Thresholding The Otsu's (Otsu (1979)) method is used for global thresholding. However, in order to reduce the effect of variation in brightness (in different regions of FoV) on the threshold, Otsu's threshold is computed for all quarter regions and the thresholds are applied separately. The effect of global threshold on the image in Fig. 3.4 is shown in Fig. 3.14 **b**. As noted, we could isolate more cells by local thresholding (Fig. 3.14 **a**) when compared to the results by global thresholding (Fig. 3.14.(b)). However, note that the cell regions are better segmented from background with less breaks/holes when compared to the result of local thresholding. We will make use of this to isolate more number of cells and is going to be discussed in next subsection.

Separating out the segmented cells and clumps The *area* and *solidity* Gopakumar et al. (2016) are computed for each object segmented out by local thresholding and are checked against measures for typical RBCs. The area is computed by counting the number of pixels in the segment and solidity is measured as the ratio of this area to the area of convex hull holding the segment. We have set lower bound for cell area as 1400 and upper bound as 2400 pixel area. If the segment whose area is within these bounds and if solidity is greater than 0.85, they are qualified as single cell. The bounds on cell area are set from the fact that typical cell diameter for an RBC is in between 6 to 8 micrometers (μm) and by considering that 15 pixels corresponds to 2.19 micrometers in our imaging setup. Thus $3\mu m$ radius corresponds to ˜20.6 pixels and $4\mu m$ radius corresponds to ˜27.4 pixels and the corresponding cell region must therefore contain 1400 pixels and 2400 pixels ($\pi \times r^2$) respectively. However, the threshold for solidity (0.85) is set empirically. Once the qualified cells are identified from global thresholded image, these are then masked out. By this masking, a few cells may get isolated in global thresholded image which were earlier part of a clump. This can be seen by analysing segmented cells in Fig. 3.14 **c** in the regions of interest marked in Fig. 3.13. The newly qualified cells are decided by computing 'area' and 'solidity' measures for the objects remaining in global thresholded image after the masking operation. The cells identified by this procedure are shown in Fig. 3.14 **c**. All other segments need to be processed separately and are shown in Fig. 3.14 **d**.

Segmenting the cells from clumps

The flowchart for the procedure to segment cells from clump is shown in Fig. 3.12. The number of cells contained in each segment is determined based on the size of a typical RBC. However, being bigger, WBCs can masquerade as a clump of RBCs. So the first step is to mask out all WBCs, if present in difficult set produced after segmenting out the well separated RBCs (Fig. 3.11). The next step is to decide approximate centroids of cells. Finally, we apply marker based watershed to segment cells from clumps, where cell centroids identified are used as cell markers. The following subsections explain this procedure in detail.

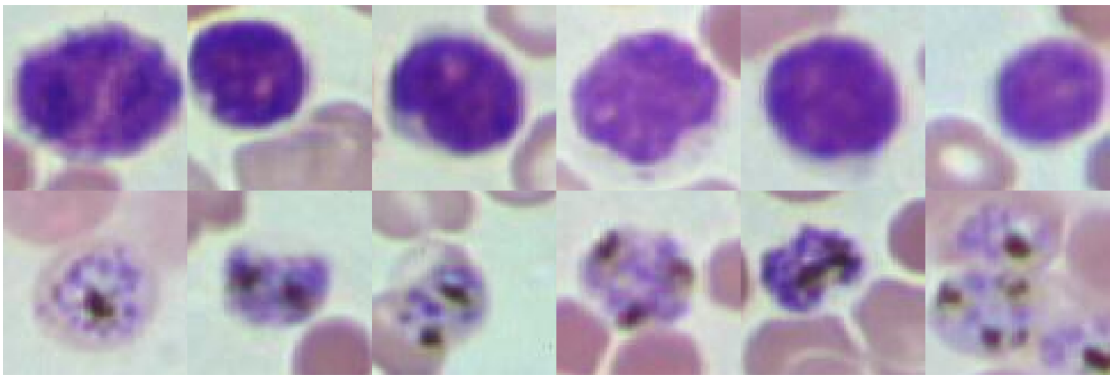

Figure 3.15: Top row : WBCs; Bottom row : infected RBCs

Mask out WBCs by Bayesian classification The Leishman stains nuclear chromatin structures in dark blue (Fig. 3.15). Being nucleated, all WBCs are stained in dark blue unlike the non-nucleated RBCs. Except lymphocytes, all other WBCs are much larger cells than RBCs. Thus, we use colour and cell size information to identify WBCs from background and RBCs. We address this as a classification problem. We had collected 1000 pixel samples each from RBCs, WBCs and background region. The colour components at these pixel location in LAB colour space (A & B) (Hunter (1958)) are used to build likelihood models for RBCs, WBCs and background. We have used Gaussian to model the distribution since it was observed that the data spread more or less in Gaussian distribution. These models are then used to decide whether a pixel belongs to RBC, WBC, or background. For any test pixel, if the likelihood of WBC is greater than the likelihood of RBCs and background, it is identified as a candidate pixel from a WBC cell. For any segment thus identified, if its area is less than the maximum area considered for an RBC, it can be either a heavily infected RBC or a lymphocyte. If the RBCs are healthy, then we can differentiate between lymphocyte and RBCs just using the colour. However, when there is heavy infection, the infected RBCs also stain in blue due to chromatin structures of parasites. Luckily, in such cases, there will be dark parasite spots in infected RBCs as shown in Fig. 3.15. In order to identify such spots, we have considered pixels whose intensity in all channels is less than 0.4 (on a scale of 1). If sufficient such pixels (10 connected pixels, in our experiments) are found, the cell is identified as an infected RBC else a lymphocyte. The threshold for identifying the black spot is decided by considering 50 heavily infected cells having dark parasite areas. With this procedure, we could correctly

locate WBCs (47 WBCs) without any false positives from the 765 slide images used in this study. Finally the WBCs are masked out and are excluded from the rest of the procedure to segment cells from the clump. The WBC identified for the slide image in Fig. 3.4 is shown in Fig. 3.16 **a**. The masked out image which is to be further processed for segmenting the cells is shown in Fig. 3.16 **b**.

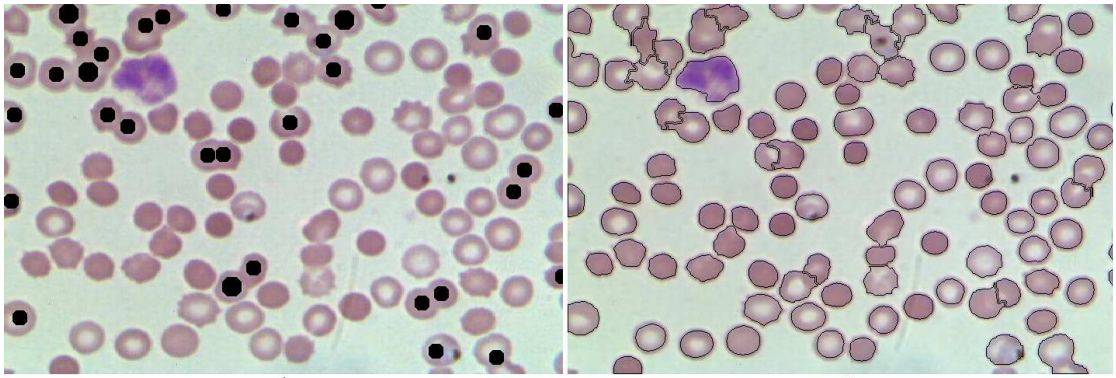

Figure 3.16: Masking out WBCs: A) WBCs identified B) WBCs excluded from clumps processing

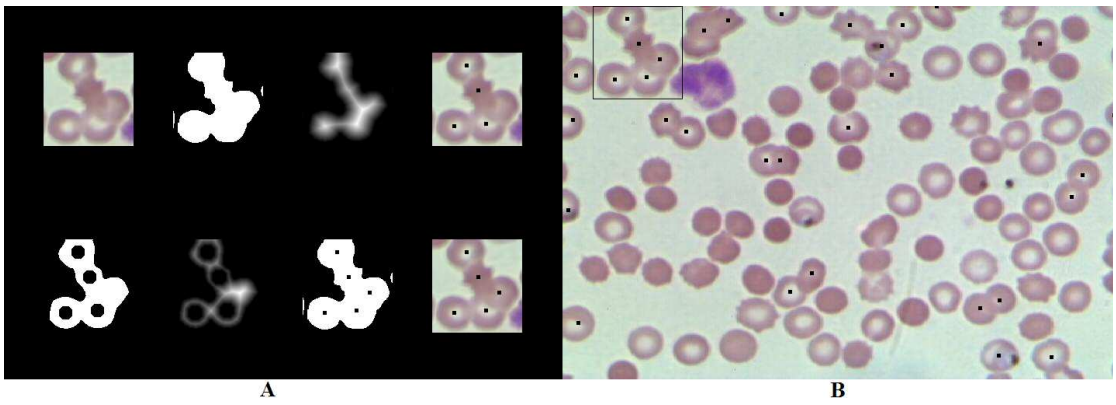

Figure 3.17: Cell centroids for clumps: A) finding cell centroids for the segment marked using bounding box in B B) cell centroids identified for the clumps

Get the centres of the cells The centroid is the farthest lying point from cell boundary. Thus, if we measure the distance for each point on the cell from background, the cell center will be the farthest from background. This means that the points on the cell which are lying on the periphery are all having zero distance while a pixel just inside the boundary will be having unit distance and so on with the highest value at the center. If there is only one cell, then the overall

maximum in the distance transform occurs at centroid. However, for a cell clump, the distance transform is supposed to have as many local maxima as the number of cells contained in it. These local maxima are due to high distance transform value for each of the cell which has boundary with respect to background for majority of the portion though there is a portion which clumps them with other cells. There are efficient algorithms that compute this distance metric (Fabbri et al. (2008)). Therefore, we compute the distance transform of the segments and regional maxima points are identified as initial centroids of cells in the segment. Being the typical cell diameter in pixels is 41, the region for determining the local maxima is fixed as a disk of diameter 41. If the number of centroids identified for each segment does not match with the number of cells expecting from the clump (segment area/typical cell area), the computed centroid regions are masked out from the segment and the remaining number of centroids is picked out using the distance transform of the masked out segment. Towards this, the centroids of the regional maxima points from the distance transform of masked out segment are computed but this time reducing the radius by 25% (empirically set) of the previous value and the top qualifying regional maxima points are picked out as centroids. The top qualifying local maxima points are identified based on how big the distance value is, in the distance transform corresponding to the points of interest. This process is repeated until the desired number of centroids is found out. From our experience, the method worked quite well, once we could correctly identify number of cells in clumps. However, if there are heavily overlapping cases, the number of cells in the clump could not be accurately determined as the method that we have proposed is only based on the cell area. This can be seen in case of the clump shown immediately right to the bounding box (just above the WBC) shown in Fig. 3.17. In this case, only two centroids are identified, even though there are three cells.

The cell centroids identified for the clumps of cells in Fig. 3.4 is shown in Fig. 3.17. Fig. 3.17 **a** shows different steps for finding the cell centroids for a segment marked in Fig.3.17 **b**. The first row of Fig. 3.17 **a** shows, respectively, a clump in the original image selected for processing, its binary image, distance transform, cell centroids, and cell centroids superimposed on the clump. Since the expected number of cell centroids for this segment is five but found only four

as explained earlier the required centroids are determined by masking out the computed centroids and taking maxima on the distance transform of the masked image. This is shown in the second row of Fig. 3.17 **a**. The images are the centroid region masked out binary image, its distance transform, all the centroids identified and the centroids super imposed image. Note that all clumps in Fig. 3.4 but the one shown in Fig. 3.17 **a** produced the correct number of centroids as desired right from the first level of processing (the processing shown in the first row of Fig. 3.17. **a**).

Split the cells from clumps Once the centroids of cells in clumps are determined, the cells are segmented using watershed (Vincent and Soille (1991)). The centroids found out are used as foreground marker, and the background image generated by global (Otsu's) threshold is used as the background marker. We use gradient magnitude as segmentation function. That is, we impose minimum at the marker locations and apply watershed algorithm on the gradient magnitude image. The output of the watershed based segmentation is shown in Fig. 3.18.

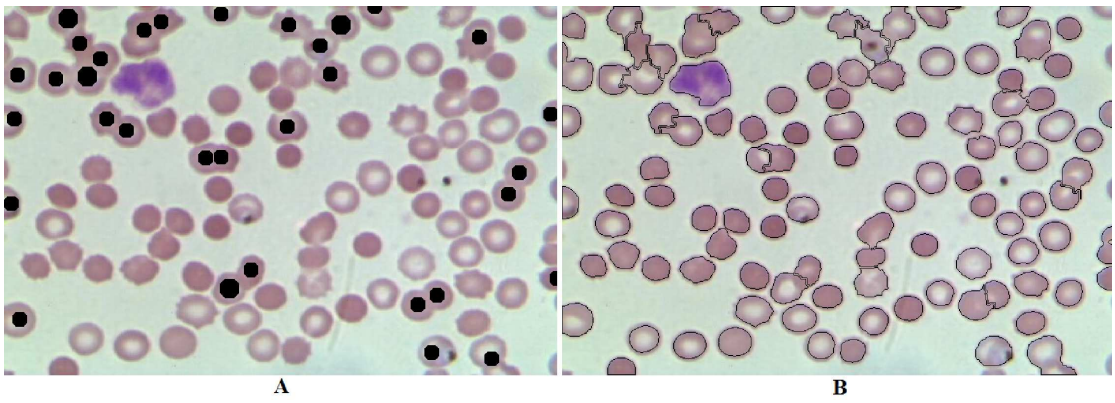

Figure 3.18: Marker Based Watershed Segmentation: A) The foreground cell markers embedded on cells B) super-imposed cell boundaries on slide image shown in Fig. 3.4

3.5 Results and Discussion

In this section, we provide and analyse the results of parasite detection and segmentation procedure presented in last sections. As noted earlier, the pathologist working in the field have marked infected locations on slide images. The number

of infected and healthy cells is then correctly identified manually by verifying the segmentation produced by the automated counting procedure. Thus the infected parasite locations marked by experts on slide images along with the number of infected and healthy cells provide ground truth for the present study. We perform the quantitative analysis of parasite detection in terms of false positives and false negatives. We perform the quantitative analysis of segmentation by seeing how close the cell count identified by the automated counting procedure is when compared to the manually verified count. We also assess the visual quality of the proposed segmentation.

As per the WHO manual on Malaria diagnosis (W.H.O. (2010)), malaria detection from blood smear requires the examination of 800 high power (100X) FoVs. To cover the prescribed physical slide area, the developed system requires imaging of only 128 FoVs (with 40X magnification). The focus stack acquisition from the required number of FoVs takes about 11 minutes. This is followed by analysis of the focus stack for identifying the infected and healthy RBCs. This analysis is conducted in Matlab 2014a installed for Windows 7, 64 bit operating system running on an Intel i3 machine @ 3.10 GHz with 4 GB RAM. The processing of each FoV (i.e., slide image of dimension $480 \times 720 \times 3$) on average took around 6.5 seconds; ˜5 seconds for segmentation and ˜1.5 seconds for parasite detection.

In order to anlalyse the effectiveness of using focus stack in accurately detecting parasite locations, we have performed a 10 fold cross validation experiment on the classifiers: FFN (Feed Forward Neural Network) & SVM on hand engineered features, CNN on the best focused patches and CNN on the focus stack of patches. Since the dimensions of feature is only 14 and since we are addressing a binary classification task, the FFN is designed with a single hidden layer having 8 neurons. For the cross validation experiment, the 5600 positive patches along with 5600 negative patches, selected at random from the available 326934 negative patches, constitute the training dataset. The entire 11200 training patches are now divided into 10 sets, by selecting samples at random but ensuring 560 positive and 560 negative samples in each set. Now, 9 sets are used to train the classifiers as discussed in previous section, and used the remaining set for validation. Such 10 run has been made, where in each run, each set is used for validation of the classifiers trained with the remaining 9 sets. We measure the effectiveness of parasite

detection in terms of sensitivity, specificity and Matthews Correlation Coefficient (MCC). All these measures are computed from the number of true positives (TP), false negatives (FN), false positives (FP) and true negatives (TN). The infected cells which are correctly classified as infected fall in TP, and that are misclassified as healthy fall in FN. Similarly, the healthy cells that are correctly classified fall in TN while the misclassified cells fall in FP. The sensitivity (TP/(TP+FN)) measures the ability of a classifier to correctly identify an infected cell while specificity (TN/(FP+TN)) measures the ability of a classifier to correctly identify healthy cells. For any classification system, there is a trade-off between these two quantities. MCC takes this into consideration and it turns out to be a better measure than the simple accuracy especially in cases where the number of positives and negatives are quite unbalanced. The MCC is defined by

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(FP + TP)(TP + FN)(FN + TN)(TN + FP)}} \tag{3.11}$$

As per the definition in Eq. (3.11), MCC value can range between -1 and 1. Value 1 corresponds to perfect classification, 0 corresponds to not better than a random guess while -1 corresponds to the worst classifier. The average sensitivity, specificity and MCC measures computed for each of the classifier across 10 folds along with their standard deviation (std) is shown in Table 3.4. The results show that the CNN working on the focus stack has an advantage in reducing false positives, and false negatives as indicated by the highest values for all measures in Table 3.4. The Receiver Operating Characteristics (ROC) for the first cross validation fold is shown in Fig. 3.19, where we plot true positive rate (sensitivity) against false positive rate (1 - specificity). The area under the curve (AUC) is a measure of the goodness of classification, where an ideal classifier should give unit area and a classifier that does a random guess should give 0.5. The mean area under the curve for 10 fold cross validation experiment that we have conducted turned out to be 0.9992 for the CNN on focus stack, 0.9987 for CNN on the best focused image, 0.9910 for SVM on features and 0.9813 for FFN (Feed Forward Neural Network) on features. The standard deviation reported are respectively $7.5764e^{-4}$, $9.9871e^{-4}$, 0.0037 and 0.0044. The high value for the mean AUC and low value of standard deviation reveals that the CNN working on the focus stack

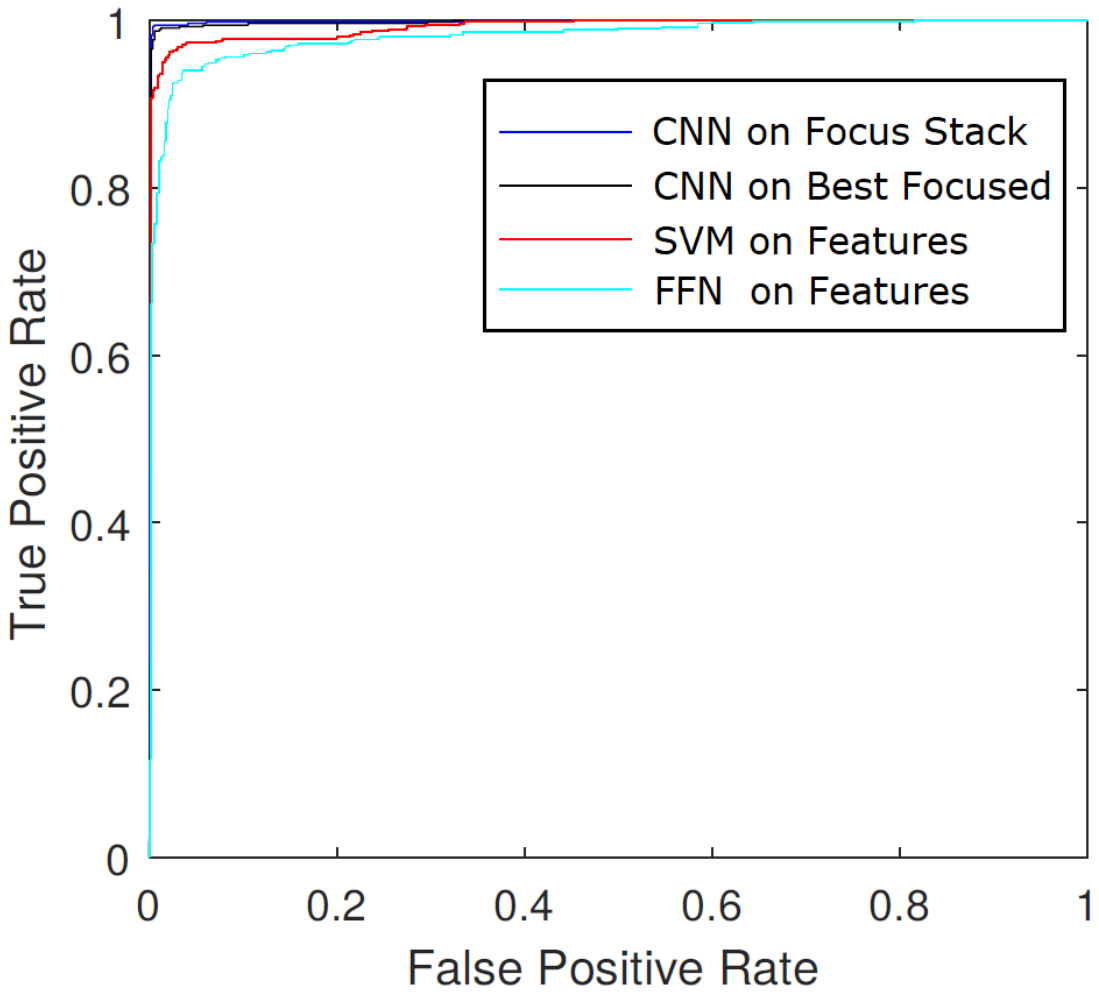

Figure 3.19: The ROC for the proposed classifiers : CNN on focus stack, CNN on the best focused image, SVM on features and FFN on Features

offers superior performance. From this point onwards, we experiment with only the top 3 performing classifiers.

In order to perform a detailed analysis on the capability of CNN operating on the focus stack, in our second experiment, we have trained the classifiers using relatively few samples. The classifiers are trained by taking at random, 60% positive samples and an equal number of negative samples. A separate 20% positive samples (+ve: samples which are really infected) and an equal number of negative samples (-ve: samples which are really healthy) are used to validate the network during training. The trained system is then tested for all patches in all slide images. The number of training, testing and validation patches are explicitly provided in Table 3.5. All classifiers are trained with exactly the same set of

Table 3.4: Average Sensitivity, Specificity and MCC along with their standard deviation in 10 fold cross validation: O) FFN on Features A) SVM on Features and CNN on B) Patches C) Focus Stack

Metric	Method - O	Method - A	Method - B	Method - C
Sensitivity	92.44% (0.84)	96.38% (0.88)	98.91% (0.36)	99.14% (0.37)
Specificity	97.36% (0.73)	95.43% (0.85)	99.39% (0.31)	99.62% (0.18)
MCC	0.8991 (0.0140)	0.9181 (0.0150)	0.9831 (0.0039)	0.9877 (0.0032)

Table 3.5: Number of Samples (#) used in Training and Validation

Patches	# Un-rotated	# Rotated	# Train	# Validation	# Test
+ve	1400	1400×4 (5600)	60% of 5600 (3360)	20% of 5600 (1120)	1400
-ve	326934	-	3360	1120	326934

candidate parasite locations in order to facilitate a fair comparison.

Note that only 3360 negative samples out of the total 333352 negative patches available are used for training since we have only very limited number of positive patches. The learning behaviour of CNNs can be found in Fig. 3.20, where the left side plot corresponds to the CNN learning only from the best focused patches while the right side plot corresponds to the CNN learned to operate on the focus stack of patches. We have used a set of independent samples to assess the accuracy of the network at each epoch during training. The accuracy of the network on this validation set can be seen in the plot (blue) shown in Fig. 3.20. In order to avoid over-fitting on the training data, we chose the network for testing as the one that gives minimum error on the validation set and not on the training set. These turned out to be the trained network at iteration 36 from the set of classifiers trained to work on the best focused patches and the network at iteration 85 from the set of classifiers trained to work on the focus stack of patches.

For a test image, the trained classifiers can be applied at candidate locations and the parasite locations can be marked. As noted earlier in subsection 3.3.1, the candidate locations are found from regional minima of the intensities. Depending on the classifier selected, the 32×32 RGB patch or the focus stack of patches or the

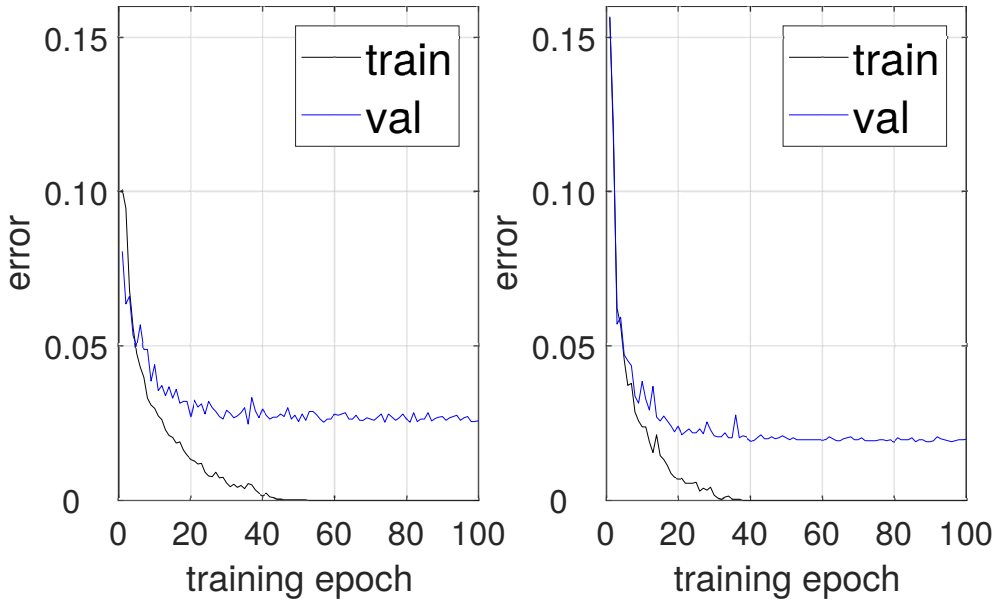

Figure 3.20: The behaviour of CNN training A.) on patches from the best focused image and B.) on focus stack of patches

features are extracted and are used for the classification. The parasite locations identified by different classifiers for the test image provided in Fig. 3.4 are shown in Fig. 3.21. The ground truth parasite locations are provided in Fig. 3.21 **a** and Fig. 3.21 **b** provides the locations detected by the SVM classifier. Fig. 3.21 **c** and **d** are respectively the locations identified by the CNN trained on the best focused patches and CNN trained on the focus stack. Note that the SVM trained on the features unnecessarily identify a healthy cell as infected while the CNN working on the best focused patch misses out one infected cell.

The confusion matrix generated for each of the classifier is shown in Table 3.6. It can be seen from Table 3.6 (c) that the CNN trained on the focus stack produced superior performance with minimum false positives and false negatives. Fig. 3.22 shows the focus stacks of 8 cell images which are resized to 40×40. The middle row holds the best focused image, the first and last row hold the images which are respectively the 16^{th} image after skipping 15 images on either side of the focus stack as discussed in subsection 3.3.3. The first 3 cell images (a)-(c) are the true positives (infected) only identified by the CNN operating on the focus stack and so as the case with the next three true negatives (d)-(f). The cells shown in Fig. 3.22 **d-f** are really healthy cells and the marks are due to dust on the sensor and are

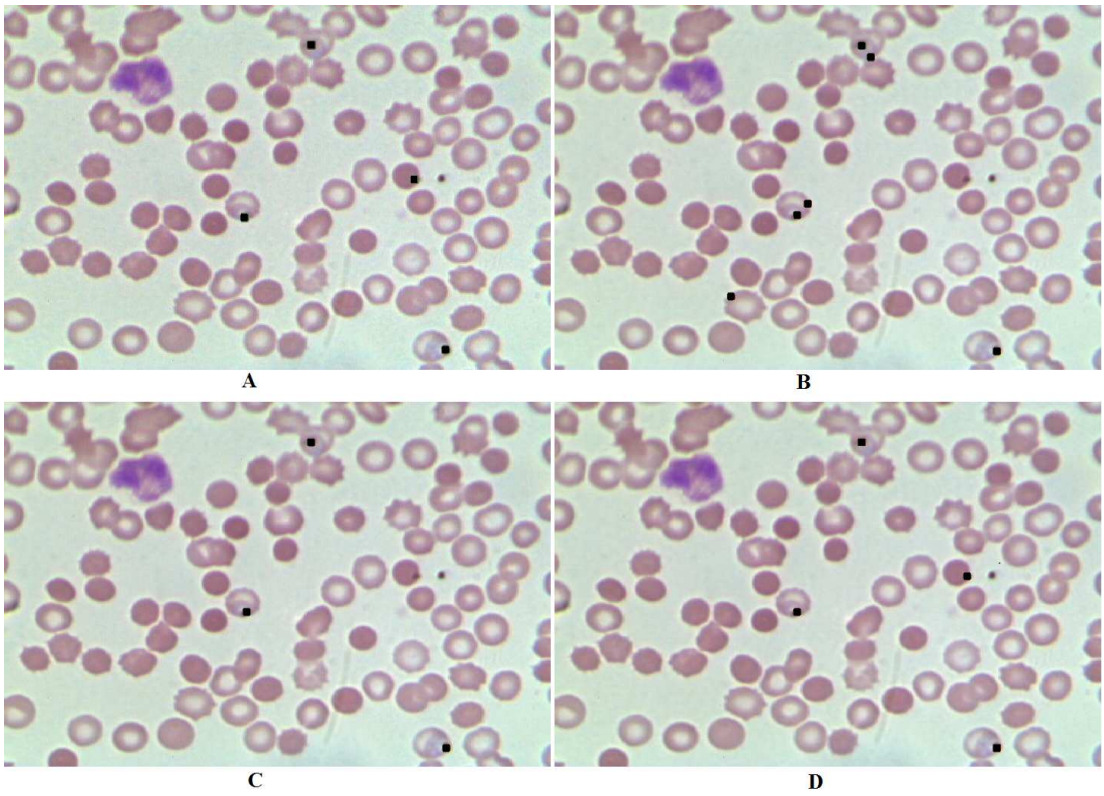

Figure 3.21: A) The ground-truth parasite locations in the slide image shown in Fig. 3.4. The parasite locations identified by B) SVM trained on features C) CNN trained on best focused image and D) CNN trained on focus stack.

correctly identified as healthy only by the CNN operating on the focus stack. It can be clearly seen that the dust area does not change considerably across the focus stack unlike the change around the parasite locations. The last two images (g)-(h) are the infected cells which are wrongly marked as healthy by all three classifiers. It can be understood from the shown cell images that it is hard to go for a decision just by looking at the cells in the best focused slide image (shown in the middle row) and as reflected by the confusion matrix in Table 3.6, CNN operating on focus stack gets an upper hand in taking the correct decision compared to the classifiers working on the single best focused image. This can be observed by the highest MCC measure in Table 3.7, for the CNN running on focus stack. The table also reveals that the proposed system could correctly classify 1156 cells out of the total 1191 infected cells yielding a sensitivity of 97.06% and it could correctly classify 59912 cells out of the total 60824 healthy cells yielding a specificity of 98.50%. The sensitivity and specificity of the system using the SVM classifier

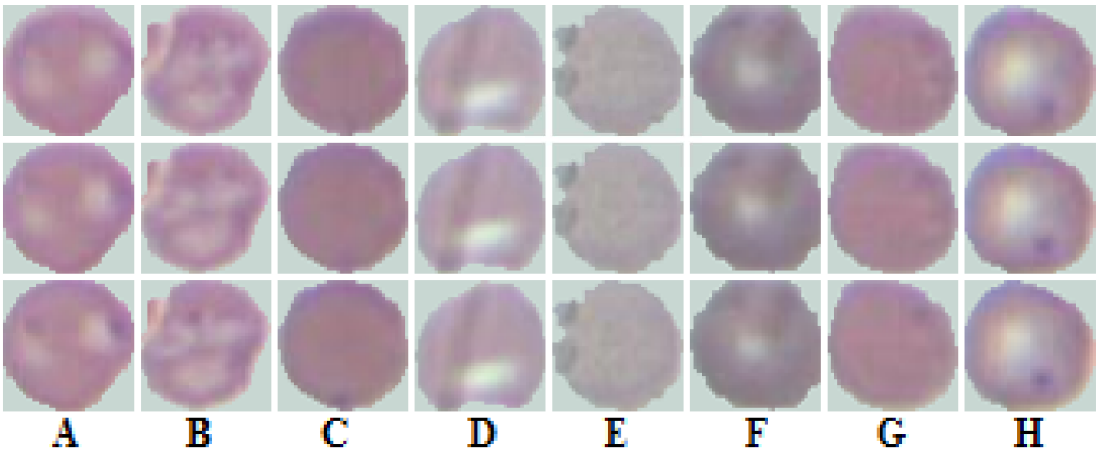

Figure 3.22: A - C) are the three true positives (focus stacks) only identified by CNN on focus stack D - F) are three true negatives only identified by CNN on focus stack G - H) are two infected cells missed out by all the classifiers

Table 3.6: Confusion Matrices : A) SVM on Features B) CNN on Patches C) CNN on Focus Stack

	Infected	Healthy
Infected	1107	84
Healthy	3756	57068

A

	Infected	Healthy
Infected	1151	40
Healthy	1053	59771

B

	Infected	Healthy
Infected	1156	35
Healthy	912	59912

C

trained on features of the patches are 92.95% and 93.82% respectively and that of the CNN trained on the best focused image are 96.64% and 98.27%. Note that, a considerably infected cell can easily be identified from the best focused image itself while the difficulty is in the case of a cell at early stage of infection as can be seen in Fig. 3.9. In such cases of early infection, though the variation across the stack is minimal, the CNN working on the focus stack could differentiate parasites from artefacts like dust where the changes across the focus stack is insignificant. This further results in the improvement of accuracy in terms of sensitivity, specificity and MCC as observed in Table 3.7.

In order to evaluate the effectiveness of the automated procedure for counting, the confusion matrix generated by the classification discussed in section 3.3.3 is shown in Table 3.8. The cells are counted by the automated procedure developed

Table 3.7: Sensitivity, Specificity and MCC: A) SVM on Features and CNN on B) Patches C) Focus Stack

Metric	Method - A	Method - B	Method - C
Sensitivity	92.95%	96.64%	97.06%
Specificity	93.82%	98.27%	98.50%
MCC	0.4430	0.7036	0.7305

Table 3.8: Confusion Matrices (Automated): A) SVM on Features B) CNN on Patches C) CNN on Focus Stack

	Infected	Healthy
Infected	1107	85
Healthy	3741	56978

A

	Infected	Healthy
Infected	1151	41
Healthy	1053	59666

B

	Infected	Healthy
Infected	1156	36
Healthy	912	59807

C

in section 3.4 unlike the results produced in Table 3.6 where the cells are counted manually. The corresponding sensitivity and specificity obtained by the completely automated system are also provided in Table 3.9. It can be easily verified that the results provided in Table 3.6 and Table 3.8 are comparable. Now, the parasitemia level reported by the automated system can be computed. It is defined as the ratio of total number of infected RBCs to the total number of RBCs considered and is typically expressed in percentage. The actual parasitemia level is defined by the ground truth and is 1.92% (1191/62015). The statistics in Table 3.8 shows that the CNN on the focus stack has produced the closest prediction 3.34% (2068/61911) compared to the CNN on the best focused patch (3.57%) as well as SVM on hand engineered features (7.83%).

In order to assess separately the parasite detection and automated counting procedure on the FoVs collected from the two slides that we have used in this study, the confusion matrices are provided in Table 3.10. There were 392 FoVs from Slide-1 and 373 from Slide-2. Table 3.10 (a) provides the statistics for first slide while Table 3.10 (b) provides the statistics for second slide. The automated

Table 3.9: Sensitivity, Specificity and MCC (Automated): A) SVM on Features and CNN on B) Patches C) Focus Stack

Metric	Method - A	Method - B	Method - C
Sensitivity	92.87%	96.56%	96.98%
Specificity	93.84%	98.27%	98.50%
MCC	0.4435	0.7033	0.7302

Table 3.10: Confusion Matrices for CNN on Focus stack for A) Slide-1 and B) Slide-2

	Infected	Healthy
Infected	566 (566)	21 (22)
Healthy	506 (506)	32352 (32293)

A

Infected	Healthy
590 (590)	14 (14)
406 (406)	27560 (27514)

B

count given by the proposed system for TP, FN, FP and TN is also provided in Table 3.10 and are provided within brackets. The corresponding sensitivity and specificity metrics computed for the detection procedure for both slides are comparable and are respectively 96.42% & 97.68% and 98.46% & 98.55%.

We have seen that the number of cells identified by the automated counting procedure comes very close to the manually verified count. Now, we will analyse the visual quality of segmentation. The slide images used in this study had so much variation in cell overlap, focus and in staining. Fig. 3.23 shows this variability where Fig. 3.23 **a** and **b** show different amount of staining and Fig. 3.23 **c** shows that a few of the cell images are not in focus. The corresponding segmentation result is provided in second row. Fig. 3.23 **d** shows a clump of two cells wrongly identified as a single cell due to the large overlapping region between the cells and Fig. 3.23 **e** shows the case where a single cell get segmented as if there are two cells.

The result of segmentation procedure proposed is compared with the result of

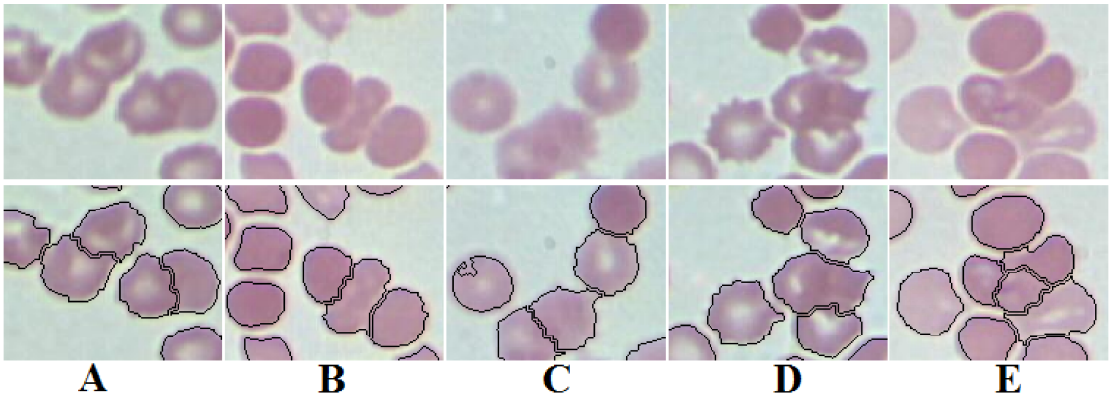

Figure 3.23: Segmentation results : A), B), C) Images at different staining level and focus D) Wrong segmentation of a clump containing two cells E) Wrong segmentation of a single cell into two cells

the method in Dimopoulos et al. (2014) where the authors used the membrane pattern for identifying the cell boundaries. We have used the implementation provided in the link associated to Dimopoulos et al. (2014). The parameters to define the membrane pattern are set as explained in the manuscript Dimopoulos et al. (2014). For this purpose, 60% of FoVs is selected at random and from each selected FoV, one cell is used to define the parameters. The result of segmentation on the slide image in Fig. 3.4 is shown in Fig. 3.24 **a**. Fig. 3.24 **c** and (d) show respectively the results obtained on the slide image shown in Fig.3.24 **b** by the method described in Dimopoulos et al. (2014) and our method. We have also shown in Fig. 3.25, the segmentation results on twelve more sub-images for a comparison. The results are ordered such that the top row shows the sub-images, second row holds the segmentation by following the membrane pattern (Dimopoulos et al. (2014)) and third row holds the result of segmentation by the proposed approach. It can be seen that the method (Dimopoulos et al. (2014)) missed out quite a few number of cells. This is because when there is more than one type of cells in the slide image under study, a unique membrane pattern is often difficult to find. In our dataset though most of the RBCs are nearly circular and are having almost the same size, some of them take very different shape and size. Also heavily infected RBC takes completely a different membrane profile and so does WBCs. From the result of segmentation obtained on our dataset, we infer that our proposed method of segmentation performed better than the method in (Dimopoulos et al. (2014)).

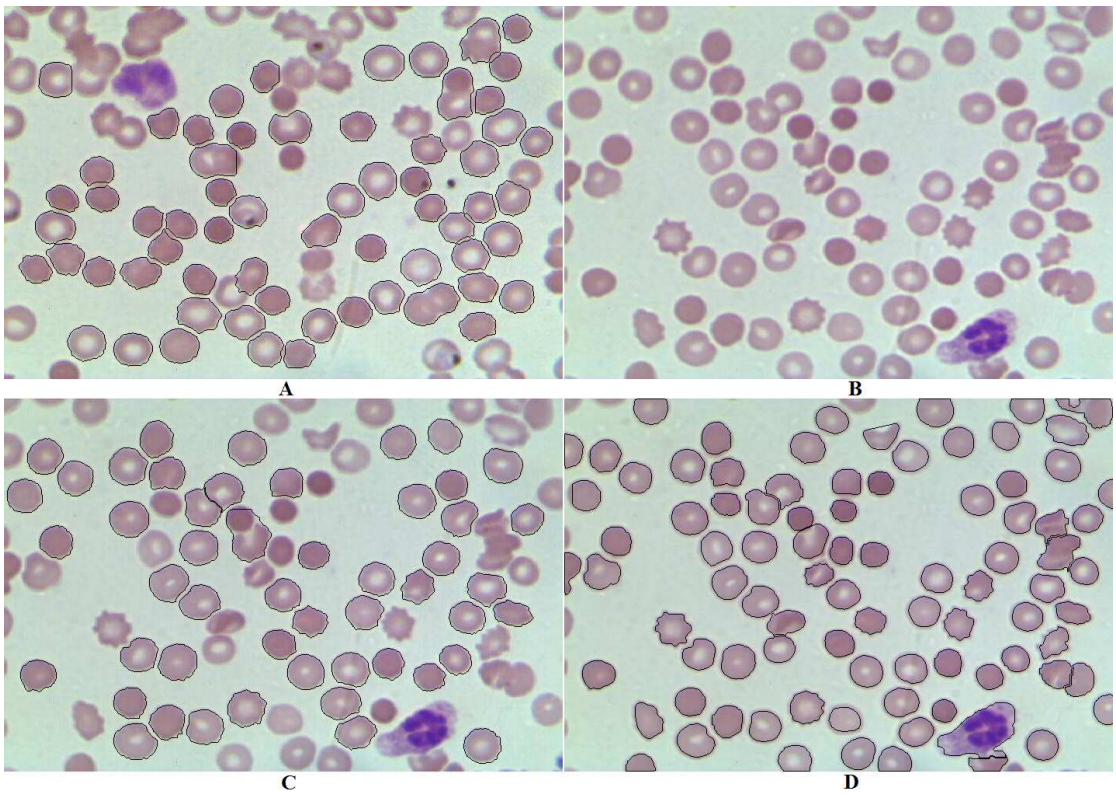

Figure 3.24: Comparing segmentation results : A) the result of segmentation by CellX method (Dimopoulos et al. (2014)) for the image in Fig. 3.4 B) Another focused image, the cells segmented using C) CellX method (Dimopoulos et al. (2014)) and D) by the proposed method.

Further, by inspecting the confusion matrices in Table 3.6 and Table 3.8, it can be seen that the number of false positives for the classification by CNN remained the same. This means that no cells marked with more than one false positive location is wrongly segmented into two cells such that the marked locations fall into multiple number of segments. Also, no two adjacent cells marked as false positive come into the same segmented region. The count difference of 15 in the number of false positives by SVM is due to the fact that adjacent cells are identified false positive but are counted as single cells since they fall in the same segment. These are heavily overlapped cases and one such case is shown in Fig. 3.23 **d**. The count difference of 105 in healthy cells between the manual and the automated cell counting is due to the same reason. Also, there is a count difference of one cell in the number of false negatives. In this case, one cell marked positive was wrongly split into two segments. One segment was identified as infected by all the three

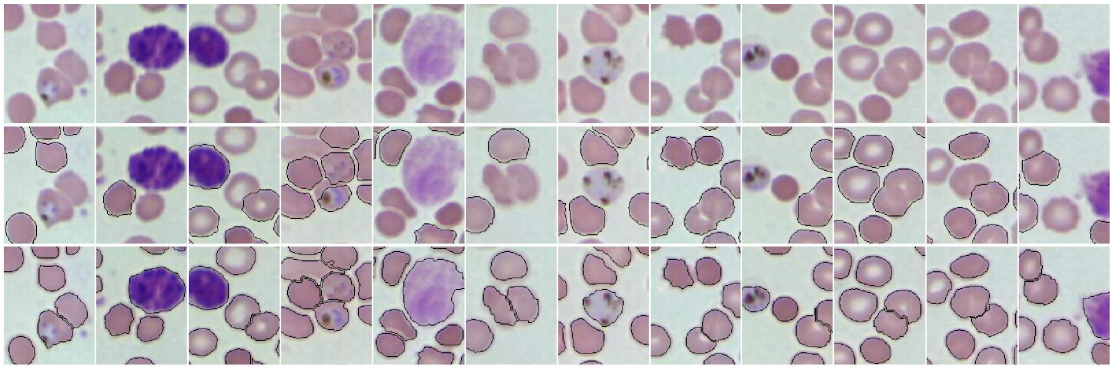

Figure 3.25: Comparing segmentation results on 12 sub-images: Original image in first row, segmentation by CellX (Dimopoulos et al. (2014)) in second row and the result of the proposed method in third row.

classifiers, but none of them identified the second as infected which contained the marked ground truth location. This case is shown in Fig. 3.23 **e**. However the comparable results in confusion matrices shown in Tables 3.6 and 3.8, reveal that overall the segmentation was good and produced comparable cell count with the manual method.

Note that, we have used the results in Table 1 to understand how good the training is across different cross validation folds (10 fold) run by considering all infected patches. The large value of the mean accuracy and low standard deviation reveal that the system trains effectively well across different set of training set used in the cross-validation experiment. Note that, we have used 90% of infected cells for each cross validation experiment reported in Table 1 while we have used only 60% infected patches for the results reported in Tables 3 to 6. Thus, the result in Table 1 provides how good the training is (mean accuracy and standard deviation) across the folds in 10 fold cross validation experiment, while the tables 3 to 6 help us in establishing the generality of approach, its robustness to classification even with 60% data and validates segmentation & classification procedures altogether.

We have also compared the detection accuracy of our method with the accuracy on a patch based malarial detection method reported in Linder et al. (2014). Though the stain for the slide images used in Linder et al. (2014) was Giemsa, we adopted the approach of using the LBP/VAR (Ojala et al. (2002)) and SIFT features (Lowe (2004)) as reported in the manuscript (Linder et al. (2014)). The

Table 3.11: Performance of the Method in Linder et al. (2014)

	Infected	Healthy
Infected	984	208
Healthy	4091	56628

	Method - Linder et al. (2014)
Sensitivity	82.55
Specificity	93.26

LBP/VAR feature with the given specification is computed with the source code available at LBP (2016). The number of false positives and false negatives obtained by this method on our dataset is shown in Table 3.11. Comparing the sensitivity and specificity of the method with those reported in Table 3.9, it can be understood that the CNN trained on the focus stack has produced the best performance. Note that, the dataset used in our research experiment itself is prepared by us using custom developed imaging setup and we have compared the proposed method with state-of-the-art method (Linder et al. (2014)) on this dataset.

3.6 Publications

1. **G. Gopakumar**, M. Swetha, S.S. Gorthi, G.R.K.S. Subrahmanyam. Automatic Detection of Malaria Infected RBCs from a Focus Stack of Bright Field Microscope Slide Images. Tenth Indian conference on computer vision, graphics and image processing (ICVGIP'16), *ACM pp.16:1–16:7*, 2016.

2. **G. Gopakumar**, M. Swetha, S.S. Gorthi and G.R.K.S. Subrahmanyam. "CNN based malaria diagnosis from focus-stack of blood smear images acquired using custom-built slide scanner", *J. Biophotonics*, 2017, DOI: **jbio.201700003**.

3.7 Summary

Malaria is a deadly infectious disease affecting a few million individuals around the globe. High degree of sensitivity and specificity are desired for any malaria diagnosis system. We have proposed a completely automated, custom-built, portable, cost-effective prototype system with necessary instrumentation as well as image

analysis/classification algorithms that can be used for quantitative malaria detection. The results produced suggest to use CNN based focus stack of image analysis in automated malaria diagnosis. We have shown better result with focus stack of images and by using CNN directly operating on the stack. We strongly believe that this work makes a significant step forward in the malaria eradication programme run by many countries and organisation worldwide.

CHAPTER 4

Automated Microfluidic Cytology for Cancer Screening

This chapter introduces cost-effective microfluidic based imaging flow cytometry and proposes a signature based framework for cancerous cell identification. By automating the processes of image acquisition and cell identification, the approach enables higher-system throughputs as well as lower overall system cost. We discuss essential instrumentation layout for high-throughput image acquisition of cells and demonstrate a signature based classification on leukaemia cell-lines.

4.1 Introduction

As discussed in chapter 2, there is huge interest in cytopathology automation. However, the research efforts to facilitate wider deployment of cytopathology have mostly been constrained to automation with very little focus on cost optimization. The automated slide based image analysis platform such as PathScope (PathScope (2016)), though improved the throughput and reduced the effort from clinician, used extensive amount of robotic handling for slide preparation making the system both bulky and costly. Similarly the flow-based image analysis platforms such as Amnis ImageStream (Amn (2016)), though improved the throughput further, used expensive fluid handling schemes and sophisticated sample acquisition systems making them costlier than the costly automated microscopy system. These facts had limited the operability and affordability of such systems in resource-poor clinics especially in low income group countries.

An elegant and cost-effective approach to automation would be to employ fluid based sample handling coupled with microfluidics. The use of microfluidic microscopy (cost-effective automated image acquisition) in conjunction with automated image processing (malignant cell recognition) help to achieve low-cost

disease diagnostic/screening platforms. The use of microfluidics as opposed to conventional robotic sample handling would ensure low system cost, miniaturization/portability while at the same time enable high-throughput image acquisition (Basiji et al. (2007); Schonbrun et al. (2013); Gorthi et al. (2013)). An added advantage of microfluidic sample handling is the controllable spatial separation between individual cells. Appropriate spatial separation would enable individual cell recognition with simpler image processing algorithms; as opposed to the slide/smear based approaches, wherein segmentation of overlapping cells demands the use of computationally intensive algorithms.

In this chapter, first we introduce a custom-built, portable, cost-effective microfluidics based imaging flow cytometry (mIFC) or microfluidic based microscopy (Mf-Ms) in section 4.2. Then we address in section 4.3, the important problem of setting up low-cost screening or diagnostic platforms for cancer. We propose a cell signature based cancer screening in section 4.4 and the publication came thereof in section 4.5. The summary of the chapter is provided in section 4.6.

4.2 Microfluidic Microscopy for High-throughput Cellular Imaging

Initially proposed in the year of 1979 (Kay et al. (1979); Kachel et al. (1979)), the concept of imaging cells in flow has evolved significantly in the recent past. Some of the currently known techniques enable imaging throughputs from few to several thousand cells per second (Basiji et al. (2007); Goda et al. (2012); Schonbrun et al. (2012)). Enhanced imaging throughput is only one of the several advantages that microfluidic microscopy offers. Different synergistic opto-fluidic arrangements have been shown to enable quantitative phase imaging (Gorthi and Schonbrun (2012)), 3D imaging (Wu et al. (2013); Sung et al. (2014)), cell height (Schonbrun et al. (2013)) and fast fluorescence imaging (Gorthi et al. (2013)) to name a few. Further, some techniques enable assessment of chemical composition of cells from images (Schonbrun et al. (2014); Di Caprio et al. (2015)). To summarize, several different forms of microfluidic microscopy which can enable acquisition of rich morphological information of huge number of cells have been developed.

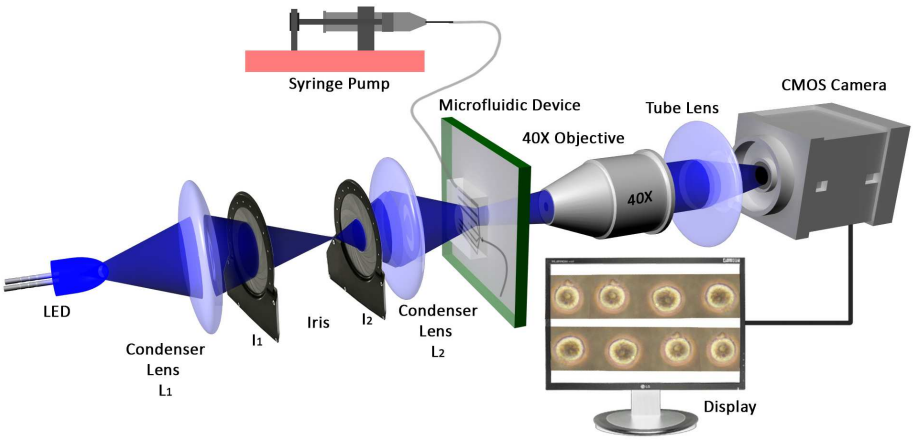

Figure 4.1: Schematic of a generic microfluidic microscopy system for high-throughput imaging.

In this section, we outline the proposed diagnostic framework and instrumentation layout of a generic microfluidics based high-throughput imaging system. The schematic of the Mf-Ms system is shown in Fig. 4.1. As the imaging is carried out on objects flowing at high flow velocities, it is essential to use very low exposure times to obtain blur-free images. This makes it necessary to have a highly bright and uniform illumination in the sample plane. In order to efficiently illuminate the sample, the standard köhler illumination configuration has been used. A high power LED (3W) is used to uniformly illuminate the sample plane. Using an aspheric condenser lens, the image of the LED is focused onto an iris. A second condenser lens is placed at a focal length distance from this iris, so as to obtain a uniform and bright illumination at the sample plane. While the lenses L_1 (f = 20 mm, diameter = 25 mm), L_2 (f = 20 mm, diameter = 25 mm) serve the purpose of collector and condenser lenses of a typical microscope, the two irises (I_1, I_2) serve the purpose of field and condenser diaphragms respectively. A microfluidic device fabricated using conventional soft-lithography (Xia and Whitesides (1998)) is placed in the sample plane. It consists of straight channels with appropriate width and depth so as to nearly match the sizes of the specimen being investigated. This helps to restrict the motion of cells within the depth of field of the chosen imaging system. If the sample contains cells of varying sizes, 3D flow focusing devices have to be employed. For the case of leukaemia cell lines (cells used in this study), channels with a width of 25 μm and depth of 15 μm are used. In order to avoid clogging of the channels with cells, a spatial filter was incorporated at the

inlet of the microfluidic device. This helps to filter out entities larger than the dimensions of the channels. For these channel dimensions, when operated at flow rates within the range 100-300 $\mu l h^{-1}$, the Reynolds number is very low ($R_e << 1$). For such low Reynolds numbers, the flow profile falls well within the laminar flow regime, wherein cells do not migrate across different stream lines and tend to flow in nearly straight lines. Such a flow stream would help individual cell identification with great ease.

The suspension of cells is flown across this microfluidic device using a syringe pump. A microscope objective (40X, NA = 0.75) has been used to image the sample plane onto the CMOS camera (Mikrotron MC1362) through a tube lens. While the cells flow across the microfluidic channels, the video of the flow stream is captured using the high-speed camera. As it can be noticed, the system consists of only one expensive component: microscope objective. The rest of the system is built using inexpensive off-the-shelf components making the system significantly cost-effective as opposed to conventional microscopes and automated slide analysers.

4.3 Case Study: Classification of Leukaemia Cells

Cancer is one of the leading causes of death around the world and about 7-8 million deaths occur due to cancer every year. About two-thirds of cancer-related deaths are in low and middle-income group countries (Stewart and Wild (2014)). Early detection of cancer is known to significantly improve the chances of successful treatment. In general, cancer detection is carried out in multiple steps using different diagnostic modalities like MRI, CT scanning etc, of which cytopathological testing is quite critical. Cytopathology is the study and diagnosis of diseases at the cellular level (Nayar (2014)). In cytopathology, samples are extracted from the suspected location of tumour using a minimally invasive technique known as fine needle aspiration (FNA). Using the extracted samples, slides are then prepared for diagnostic assessment via microscopic observation. Due to the increased nuclear activity, cancerous cells tend to exhibit abnormal morphological features like a non-uniform nucleus or differently coloured nucleus (Dey (2010)). These

morphological signatures help in identifying the cancerous cells. However, cytopathological testing is a skilful and considerably involved process and tends to suffer from limitations such as high cost and low throughput. In this chapter, we propose a cell signature based cancerous cell identification system for a low-cost, custom-built, high-throughput microfluidic microscopy.

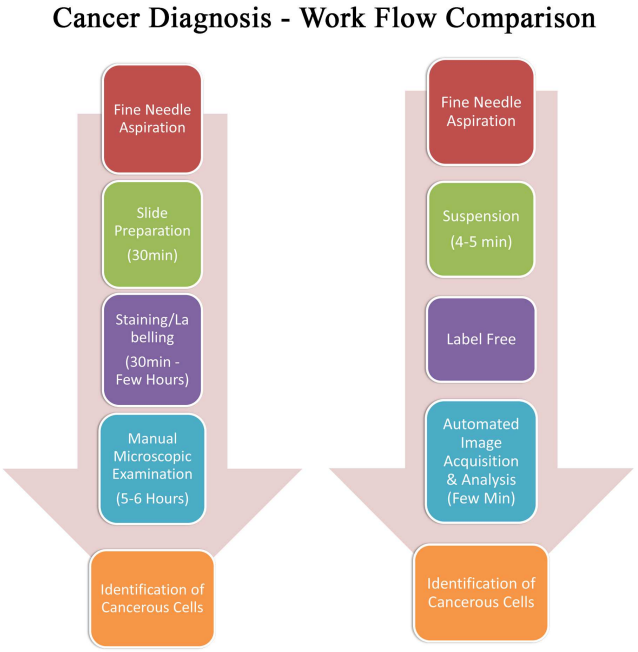

Figure 4.2: Comparison of experimental work-flow for conventional cancer cytopathology and the proposed technique

The overall comparative work-flow of the diagnostic analysis using conventional cancer cytopathology and the microfluidics-based high-throughput imaging system is shown in Fig. 4.2. In conventional methods, a smear of the collected sample has to be prepared on slide following the FNA/biopsy. The process of slide preparation requires skill and typically takes about 30 minutes (Bancroft (2008)). In the case of the proposed technique, the cells have to be re-suspended in solution which is far simpler and can be performed even by untrained personnel. Also, it takes much lesser time when compared to conventional slide preparation techniques. For re-suspension, the known protocols (Terstappen et al. (1990)) in conventional flow cytometric analysis of bone marrow aspirate (Terstappen et al. (1992)) can be adopted. Further, adhesive cells from tissues can be extracted for analysis, using well established flow cytometry protocols. After the slides have

been prepared, the cells have to be stained appropriately. In the conventional approach, the process of staining may take about 30 minutes to a few hours (Bancroft (2008)) depending upon the tissue/cells (under investigation) and the stain being used. In contrast, the proposed technique is a label and stain-free method. Even if the nucleus of cell has to be stained, it can be done by simple mixing of the stain solution at appropriate concentration with the suspension of cells. Unlike conventional slide preparation which involves numerous essential steps (like fixation, several wash steps, staining), in-suspension staining of cells is significantly simpler. In some cases, the clinician may choose to repeat the test or assess additional quantity of the biopsy sample. Such a scenario would demand the preparation of additional slides, further increasing the burden on the clinician/technical personnel. Whereas, in the current framework, one would have to simply flow an added amount of sample (through the microfluidic device) to assess further sample quantities and acquire images of more cells. This, greatly alleviates clinician's burden, while performing a more rigorous assessment of the sample. Further, manual slide scanning requires several hours; whereas a microfluidics based image acquisition is automated and takes comparatively much less time.

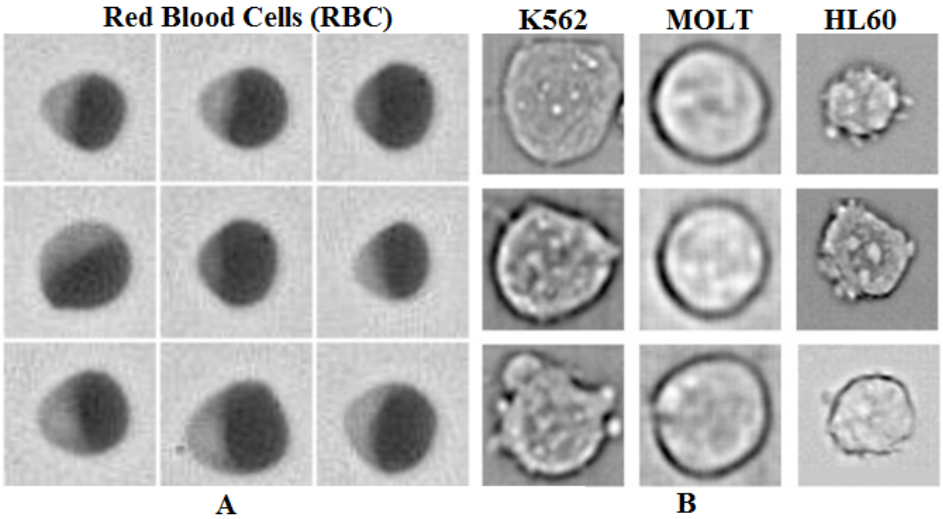

Figure 4.3: a) Red Blood Cells; b) Leukaemia cells in Mf-Ms channels

In this work, we investigate for the presence of principal component analysis (PCA) based unique signatures of cells that are neither stained nor labeled. Moreover, PCA based identification methods become more robust with the increase in size of training data sets. As it enables fast acquisition of large image data sets, automation enabled by microfluidic sample handling compliments the

chosen training and classification method. We demonstrate the signature based cell classification framework by addressing the leukaemia cell-line classification. The cell lines of HL60, K562 and MOLT are obtained from ATCC (American Type Culture Collection) and are separately cultured in the lab. The rationale behind demonstrating the usefulness of the presented medical imaging approach with the cultured cell-lines rather than the clinical blood samples is twofold: (1) To have ground-truth and thereby in a position to evaluate the performance of the classification (2) Classification of RBCs from WBCs/cancer cells is fairly easy, and not of much interest in the current work while the former would contribute to unnecessary bloating of the dataset. For the chosen wavelength of illumination (LED having 405 nm central wavelength), Haemoglobin in RBCs absorb quite a lot of light and appears to be darker at the front part of cells (Fig. 4.3). Thus RBCs can easily be distinguished from that of WBCs/cancer cells based on size (RBCs are 6-8 microns, while cancer cells are about 20 microns) and absorption properties. A total of 618 cells are imaged; 388 HL60, 124 K562 and 106 MOLT. Though at first it seems as a small data set, it is having significant size. Note that there is only about $4-11\times10^3$ WBCs in one microliter of blood, while the number of RBCs is $4-5\times10^6$. Thus even a high throughput system having the capability to image say 10,000 cells/sec would still end-up in imaging only 10 WBCs in a second, while rest of them would be RBCs. As we have noted, due to their simple morphological structure differentiating RBCs from WBCs is a fairly simple task (Fig. 4.3), whereas the identification of abnormal or cancerous cells amongst the few hundreds of WBCs is a challenging problem; the solution of which is essential for diagnosis of the disease. In case of screening for leukaemia from blood samples, we must rely on the WBCs having abnormal features/morphology, and thus the total number of cells in the data-set with which one needs to perform identification/classification is still only a few hundreds of cells.

In the following section, we discuss our signature based approach for cancer cell identification.

4.4 Signature Based Cancerous Cell Identification

We propose a signature based cell classification approach for the leukaemia cell-lines captured using a custom-built, portable, low-cost microfluidics based imaging flow cytometry device. The schematic of the framework is shown in Fig. 4.4. In conventional microscopy classification systems, specific morphological features which are meaningful to expert's perception are extracted from cell images subsequent to dedicated segmentation process. Whereas, the proposed approach employs cell images directly as they are and encodes 1D PCA or 2D PCA features similar to well known face recognition systems and apply support vector machine (SVM). As in the case of face recognition systems, the demonstrated approach is capable of analysing and identifying cells from a large image dataset. Evaluation of a large image set by human experts would be significantly cumbersome, time-consuming and possibly error-prone. Further, the presented approach allows for generation of class (cell type) specific computational signatures, which may not be identifiable by human experts. We demonstrate the framework by addressing the classification of leukaemia cell-lines : K562, MOLT and HL60.

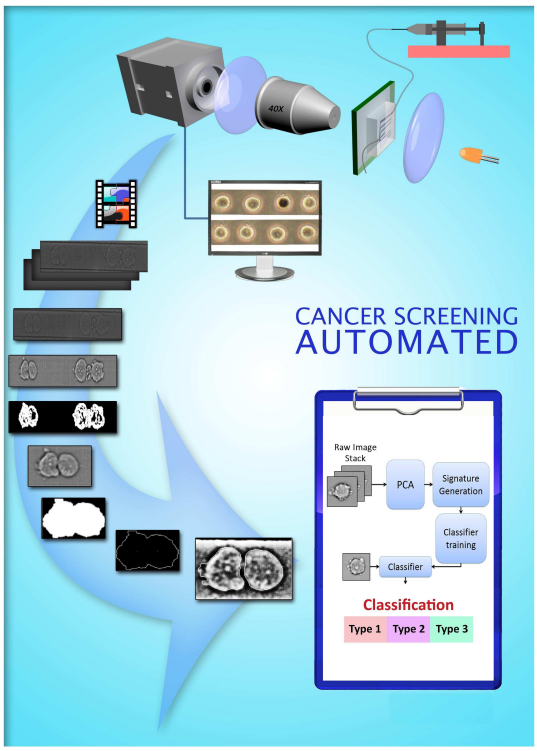

Figure 4.4: Overview of the proposed screening framework for cancerous cell identification

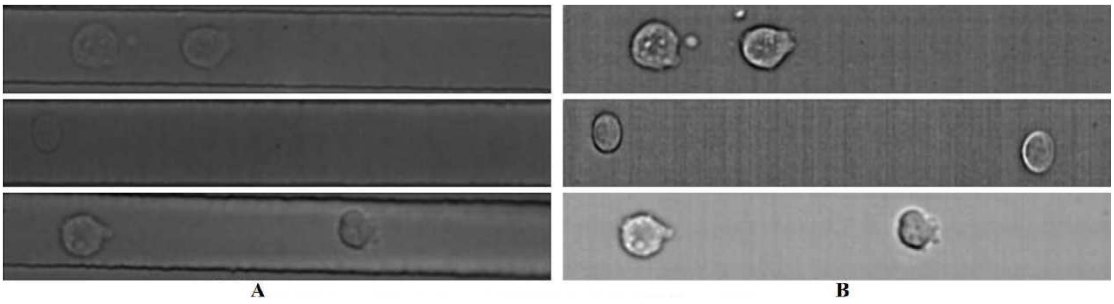

Figure 4.5: a) Frames in each row contain respectively K562, MOLT and HL60 cells; b) the corresponding background subtracted enhanced frames.

As mentioned earlier, the images acquired from Mf-Ms are mined for possible signatures, which are then used to classify cells. Class specific signatures are generated from the training samples. Following which, the classification is performed based on the proximity of the test sample to these signatures. Raw videos of flow streams containing different leukaemia cell lines K562, MOLT and HL60 are separately acquired for this research using the setup provided in the last section. Each frame from the acquired video is pre-processed in order to enhance cell features and reduce the noise. The pre-processing step involves the filtering using 5×5 averaging mask in order to reduce the noise due to cell debris followed by background subtraction to enhance cellular features. The background frame for subtraction can easily be obtained by capturing the channel along with sheath fluid before pumping the cells. The result of enhancement achieved through noise reduction and background subtraction can be observed in Fig. 4.5. The rough location of the cells is then identified via thresholding using the higher gradient (in intensity) at the cell location. The cells from the frame are roughly localised by identifying the rectangular bounding box enclosing the morphologically filled binary cell image (Fig. 4.6 **b**). The cells are then segmented out using this bounding box and are resized to $m \times n$ (In our experiment, $m = 26$, $n = 24$). The detailed procedure for the localisation of the cells is discussed in chapter 5.

For the classification system, we need to assign a general signature describing the samples in each class. In this study, the class specific signatures are generated based on principal component analysis. In PCA based methods, we look for a few directions that can capture the variability of cell population with minimum

Figure 4.6: a) Roughly localised cells from the K562 frame in Fig. 4.5; b) bounding box containing left most object of **'a'** and left most cells from the background subtracted frames in Fig. 4.5

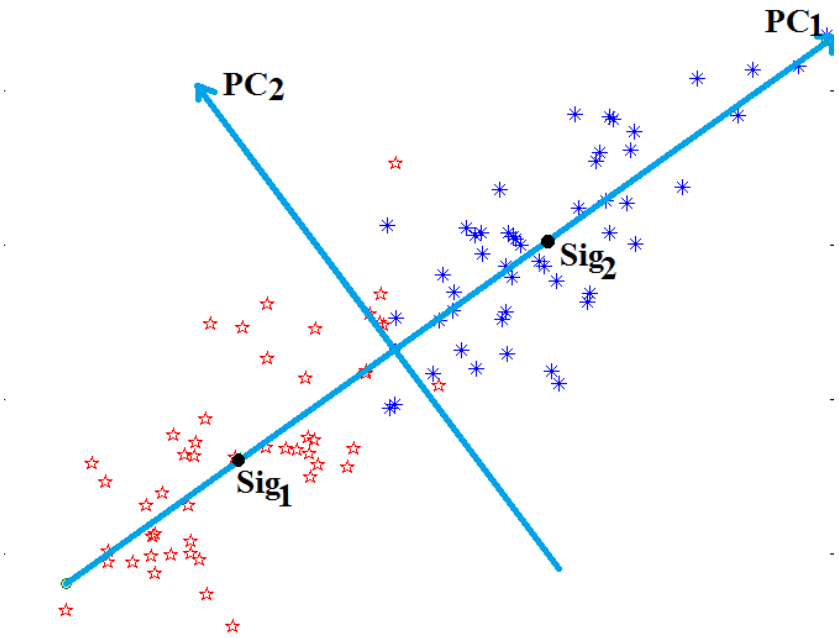

Figure 4.7: Depicting principal directions (PC_1 and PC_2) for a sample two class 2D dataset. The signatures (Sig_1 and Sig_2) found by projecting the samples to PC_1 are also shown.

loss/distortion. If we could select sufficient such directions, the cells could be effectively represented using a signature in low dimension thereby reducing the effect of noise in the signature. We experiment with both 1D (Kirby and Sirovich (1990); Turk and Pentland (1991)) and 2D PCA (Yang et al. (2004)) cell signatures.

4.4.1 1D PCA based Cell Signature

In 1D PCA, the cell images are reshaped as 1D vectors by stacking the columns, and we want to find the basis directions that captures the variability of the data with minimum distortion. Let $\{x_i\}_1^N$ be the set of vectorized N images each of

having size $mn \times 1$, where $m \times n$ is the dimension of the cell images. We are looking for a direction e such that the variability of the mean (μ) subtracted vectorised cell images is best captured. Thus, we look for a unit vector e that minimises the following error function J across N images.

$$J = \sum_{k=1}^{N} \|a_k e - (x_k - \mu)\|^2 \tag{4.1}$$

Where

$$\mu = \frac{1}{N} \sum_{k=1}^{N} x_k \tag{4.2}$$

Looking for a_k, the best representation of the image for its feature vector x_k turns out to be $a_k = e^t(x_k - \mu)$. This can be found by differentiating Eq. 4.1 with respect to a_k and equating to 0, as the best a_k minimises the error. To find the direction e, we can substitute the value of a_k back in Eq. 4.1. This results in

$$J = -e^t(x_k - \mu)(x_k - \mu)^t e = -e^t S e \tag{4.3}$$

Here in Eq. 4.3, J has to be minimised with respect to the constraint that e is a unit vector. i.e., $e^t e = 1$. Also note that it is the same e that maximises $e^t S e$, where S is called the scatter matrix of the vectorised cell images. The unconstrained form of this maximisation is due to Joseph-Louis Lagrange and is

$$J^{'} = e^t S e - \lambda(e^t e - 1) \tag{4.4}$$

When differentiated with respect to e and equated to 0, to find the e that maximises $J^{'}$, we will get $Se = \lambda e$. Thus the best direction that we are seeking is nothing but the Eigenvector of S. It is particularly the Eigenvector corresponding to the largest Eigenvalue since $e^t S e = \lambda e^t e = \lambda$ being the function that we try to maximize.

Fig. 4.7 explains the 1D PCA based signature generation taking a random 2D two class dataset. The class-1 samples are represented using star symbol and class-2 samples are represented using asterisk. The directions along which the maximum variability in data is observed along PC_1. The two directions PC_1 and its perpendicular direction PC_2 are the principal directions for the given data. In

order to find the 1D PCA signature (for 1 dimension), the class specific samples are projected onto PC_1 and the mean is computed. The respective signatures are shown as Sig_1 and Sig_2. In general, to generate signature based on 1D PCA, the samples are projected onto the vectors representing 'd' principal directions, along which the samples have greater variance (Fig. 4.7). As we have seen, these principal directions turns out to be the eigenvectors corresponding to the top 'd' eigenvalues of the scatter matrix computed from the training vectors. Each of such projection by a training sample yield a 'd' element vector, and the PCA signature of each class is then identified as the mean of the projected vectors of the class specific training samples.

4.4.2 2D PCA based Cell Signature

In 2D PCA, we directly deal with the 2D cell images $\{A_i\}_{i=1}^{N}$ each of dimension $m \times n$. We look for a n dimensional unit vector P on to which the cell images can be projected such that the projected vectors best captures the variability of the cell images. Yang et al. (2004) has identified that the total scatter of the projected samples can be used to measure the discriminatory power of the projection vectors. This is reflected by the trace of the scatter matrix S_Y of the projected output vectors, and has to be maximised. Thus the cost function that we want to maximise is $J = trace(S_Y)$.

$$S_Y = \mathbb{E}[(Y - \mathbb{E}Y)(Y - \mathbb{E}Y)^t] \tag{4.5}$$

Substituting $Y = AP$ in Eq. 4.5 will produce

$$S_Y = \mathbb{E}[\{(A - \mathbb{E}A)P\}\{(A - \mathbb{E}A)P\}^t] \tag{4.6}$$

where $\mathbb{E}$ is the expectation operator.

$$trace(S_Y) = P^t\{\mathbb{E}(A - \mathbb{E}A)^t(A - \mathbb{E}A)\}P = P^tGP \tag{4.7}$$

Thus the projection vector that best capture the variability of our cell images can be found by finding P that maximises the cost function defined in Eq. 4.7.

Table 4.1: NN – Average Accuracy % (std) : Label of Nearest Signature

Classifier	K562	MOLT	HL60
1D PCA + NN	89.23 (3.21)	78.70 (5.14)	84.66 (2.13)
2D PCA + NN	88.34 (4.08)	80.79 (4.43)	85.78 (2.38)

For the reason discussed in 1D PCA, this vector turns out to be the Eigenvector corresponding to the largest Eigenvalue of the image scatter matrix, G.

4.4.3 Matching Test Signature with Stored Signatures

We have chosen 40 principal directions for 1D PCA signature generation and have used 20 principal directions for 2D PCA signature generations. The Eigenvalues of these principal directions contributed around 99% to the sum of all Eigenvalues. When a new test sample comes, it is projected onto the same projection directions used to generate the class signatures. The class of the test sample is decided based on the proximity of the test signatures to the class specific signatures. Note that, all the results discussed in this chapter are generated by taking 50% of the available samples for training (to create signature) and the rest of the samples for testing. Table 4.1 shows the result of assigning the label of proximal signature.

In our second experiment, the difficult cases are separated out and are given the label 'Ambiguous'. The cytopathologists need to examine only the ambiguous cases rather than spending time assessing the clearly normal and abnormal cases. This would make for efficient utilization of the expert's time. The results are provided in Table 4.2, where the label 'Ambiguous' is assigned if the two proximal signatures are not well separated. In our implementation, if the distance to the closest signature (d_1) is within 85 % of the distance to the second closest signature (d_2), the test sample is considered as ambiguous (i.e, when $d_1/d_2 < 0.85$). The classification accuracy for each cell type is computed without considering the samples assigned the 'Ambiguous' label. The mean accuracy as well as standard deviation for 100 independent runs is shown in Table 4.2 and 4.3. For example, the first row in Table 4.2 shows that among the classified K562 samples, 94.11% are correctly classified. Similarly 81.54% MOLT and 86.16% HL60 are correctly

Table 4.2: NN – Average Accuracy % (std) : Difficult cases as Ambiguous

Classifier	K562	MOLT	HL60	Ambiguous
1D PCA + NN	94.11 (2.81)	81.54 (4.90)	86.16 (2.48)	6.64 (0.98)
2D PCA + NN	94.37 (2.44)	86.82 (4.26)	87.35 (2.66)	8.87 (1.27)

classified. The overall accuracy observed among all the classified samples using 1D PCA is 86.99% while 2D PCA is 88.61%. The classification accuracy has been increased among the classified samples, but at the cost of keeping a few as 'Ambiguous'.

We have also explored the possibility of using SVM for reducing the false positives by classifying difficult cases as 'Ambiguous'. Note that, SVM finds the best separating hyperplane that maximizes the margin while minimizing some measure of loss on the training data. The classification of the test samples has been done based on the side a given case falls, with respect to the hyperplane. Fig. 4.8 shows the typical SVM for noisy class samples. It shows the separating hyperplane for 2 classes (Star and Asterisk), the support vectors identified, as well as the margin. The ξ is the SVM parameter used in the design to quantify the error of the training samples in terms of the distance from the margin boundaries. The separating hyperplane is decided as a compromise between maximizing the margin and minimizing the error term. In our experiment, the projected training samples onto the principal directions are used to train SVMs. Being a three class classification problem, three SVMs are trained by taking samples from each class as positive and keeping all other samples as negative. A test sample is classified based on the following strategy. If there is only one claim by the classifier, the test sample is assigned the label of the positive class used to train the particular classifier, provided the distance of the test sample is at least 0.8 unit (margin is at 1 units) away from the separating hyperplane. In all other cases, the label 'Ambiguous' is assigned. The results are shown in Table 4.3. The overall accuracy observed among all the classified samples using 1D PCA is 98.01% while 2D PCA is 97.71%.

In essence, the individual methods/modules employed to demonstrate the approach are bright-field in-line transmission microscopy, PCA based feature extrac-

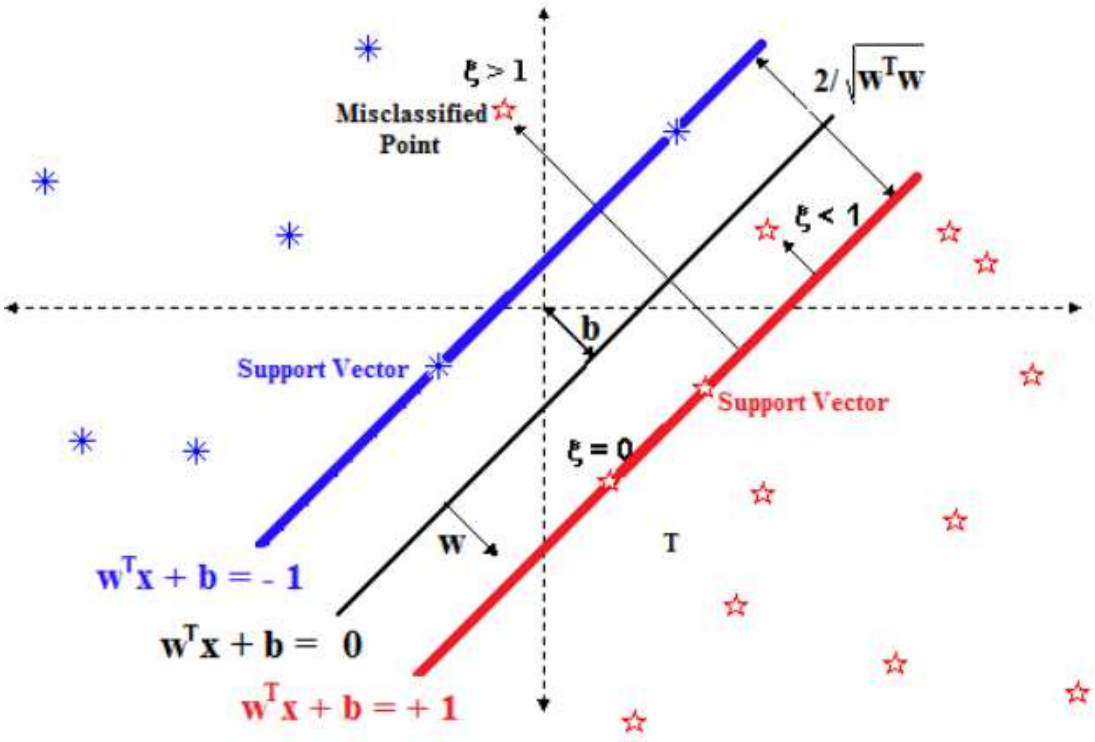

Figure 4.8: Shows the data from 2 classes (Asterisk and Star), support vectors, margin ($1/\sqrt{W^T W}$) and the separating hyperplane.

Table 4.3: SVM – Average Accuracy % (std) : Difficult cases as Ambiguous

Classifier	K562	MOLT	HL60	Ambiguous
1D PCA + SVM	93.46 (3.93)	93.74 (3.71)	99.94 (0.22)	15.17 (2.35)
2D PCA + SVM	89.30 (2.44)	93.97 (4.26)	99.99 (0.05)	14.26 (1.90)

tion, NN/SVM based classification. While the individual modules/methods are quite well known, the purpose is to demonstrate the realization of an exemplary "Malignant Cell Surveillance System" (similar to surveillance cameras powered by face recognition algorithms for the identification of criminals) suitable for cancer screening, with the powerful combination of microfluidic microscopy and digital cytology. Further, we have extended the standard NN and SVM classification procedures to separate out the difficult cases leading to more accurate classification.

4.5 Publications

1. V.K. Jagannadh, **G. Gopakumar**, G.R.K.S. Subrahmanyam, S.S. Gorthi. "Microfluidic microscopy-assisted label-free approach for cancer screening: automated microfluidic cytology for cancer screening", *Med. Biol. Eng. Comput.*, 1–8, 2016.

4.6 Summary

To summarize, we have presented a novel approach for automated high-throughput label-free cancer screening. We have presented the clinical diagnostic work flow and the associated instrumentation framework to perform image acquisition of cells. We envisage that the presented approach leverage and bridge independent advances in the fields of modern microscopy, automated high-throughput imaging and sample preparation/handling with Lab-on-a-Chip and image classification. This would in turn give impetus to the development of affordable and automated triaging systems for the early detection of deadly diseases such as cancer.

With the use of presented approach, an exhaustive database of various types of cancerous cell signatures can be very easily generated for a given type of cancer. Analogous to a criminal database (for face recognition), this database would help in instantaneously and automatically identifying a cancerous cell as soon as it is imaged by microfluidic microscopy system. This high-throughput cancerous cell identification system can also detect the presence of rare cells (like circulating tumour cells(CTCs)). CTCs form a minuscule percentage of total cells present

in a given volume of blood. Detection of CTCs in a given blood sample can be an invaluable tool for early detection as well as prognosis monitoring of the disease (Cohen et al. (2008)). In the event that this approach fails to generate required signatures for certain types of malignant cells (necessary for unambiguous identification), it would act as an efficient and automated approach for triaging(primary screening). Triaging is more relevant for resource-limited settings and when augmented with other diagnostic modalities, enables early detection of cancer (Solomon (2003)). The approach presented here can be easily used to carry out triaging, wherein the cells can be classified into different categories like 'normal', 'malignant/abnormal' and 'suspected-to-be-abnormal'. Cost-effective automation of cancer diagnosis would potentially facilitate screening camps in the low-income group countries and thereby help prevent a significant number of deaths due to cancer. Further development of this technique would transform cancer screening test into a routinely performed laboratory investigation.

CHAPTER 5

Framework for Morphometric Classification of Cells in Imaging Flow Cytometry

Imaging flow cytometry (IFC) is an emerging technology that combines the statistical power of flow cytometry with spatial and quantitative morphology of digital microscopy. We had provided a low-cost IFC instrumentation as well as signature based cancerous cell identification in chapter 4. However, to have a clinically usable system, the accuracy has to be improved further and in this chapter we propose a general framework for the processing of cells imaged using the microfluidics microscopy (Mf-Ms) system presented in chapter 4. The framework includes feasible preprocessing, segmentation, feature extraction and classification stages. Each cell is localised by finding an accurate cell contour. Then features reflecting cell size, circularity, and complexity are extracted for the classification using support vector machine. Unlike the conventional iterative, semi-automatic segmentation algorithms such as active contour, we propose a non-iterative, fully automatic graph based cell localisation. The framework is then tested for the same leukaemia cell-line dataset (K562, MOLT and HL60) presented in chapter 4. The proposed system is a significant development in the direction of building a cost-effective cell analysis platform which would facilitate affordable mass screening camps looking cellular morphology for disease diagnosis.

5.1 Introduction

In microfluidics based IFC, the cells in suspension are allowed to flow across the microfluidics channel and the video of the flow stream is captured using a high speed microscopy system. The raw data (videos) generated by microfluidics based IFC system has to be processed so as to extract and subsequently analyse images of cells. A highly efficient classification system, which employs different algorithms to extract quantitative morphological features of cells, is an essential requirement

to develop a practically usable IFC system. Computerized methods have been evolved to assist pathologists by automating the segmentation of cells (and its components such as nuclei (Di Cataldo et al. (2008)) and cytoplasm), feature extraction and classification (Thiran and Macq (1996)). A good review on techniques used for segmentation, feature extraction, and classification of cells from histopathology imagery can be found in Irshad et al. (2014). Image processing algorithms for histopathology slides are thus a sufficiently explored subject whereas IFC is relatively new imaging modality and is yet to take the form of a diagnostic tool.

We have introduced a low-cost microfluidic microscopy instrumentation along with signature based framework for classifying leukaemia cell-lines in chapter 4. However, in order for IFC to be practically usable as a diagnostic tool, the accuracy has to be improved further. This requires development of a framework to assess the raw data and extract relevant information. The framework design would involve the development of intelligent image processing algorithms for the pre-processing and segmentation of cells, reliable feature extraction techniques to capture the important and discriminant features of cells, and finally the development of high performance classification system/model. Once the classification model is developed with a set of training images (cells), subsequent classification and counting can be done in nearly real time, provided the pre-processing, segmentation and feature extraction are made to be less complex in computation.

This chapter proposes a general framework for the processing of cells in a microfluidics based imaging flow cytometer. As shown in Fig. 5.2, the framework incorporates several steps: beginning from pre-processing of the raw video frame and culminating in extraction of different quantitative morphological parameters and subsequent classification of cells. The first step in the proposed framework is a rapid pre-processing stage to identify the presence of cells within the frame and subsequently enhance the cell images. In IFC, the cells are imaged one at a time and in most cases appear as isolated cells, with a few exceptions where the cells may appear as a clump. The proposed framework also takes into account these few exceptions and includes a clump identification step. Following the clump identification, a novel, accurate, fully automatic, non-iterative graph based algorithm has been implemented to segment out individual cells from the clump.

Once the cells are localised by identifying its contour, we present the relevant image processing algorithms which can be used to extract morphological features reflecting size, shape and complexity of cells. The extracted features are taken for cell classification using SVM. The effectiveness of the proposed framework is tested by performing the classification of label-free unstained leukaemia cell lines K562, MOLT and HL60.

In conventional microscopy/slide based examination, the expert has to analyse a large population of cells to identify abnormal cells. In most cases, abnormal cells form a very small minority of the overall population. For example, in the case of a typical malarial infection, only about 0.2% of the total population of red blood cells is infected and in the case of cancer, there are only about 5-10 cancerous cells (circulating tumour cells) per millilitre of blood (which contains more than a few billion normal cells). Further, it is difficult to assess some cells at first look and have to be re-examined before taking a clinical decision. These cells represent the suspected-to-be abnormal cases. Examining clearly normal cells or clearly abnormal cells put an unnecessary workload on the expert. In the current work, we have developed the classification system with a realistic clinical scenario in mind. The system enables classification of cells as normal and abnormal. The difficult cases are labeled as 'Ambiguous', which would then be closely examined by the expert.

In a nutshell, the main contributions in this chapter are 1) the proposal of a general framework for processing cells in microfluidics based IFC, 2) the proposal of a novel, fully automatic, non-iterative algorithm for cell segmentation, 3) the extraction of relevant features to perform efficient classification of leukaemia cell-lines K562, MOLT and HL60 and 4) exploring the possibility of reducing false positives by identifying the suspected-to-be abnormal cases. The content of this chapter is organized such that section 5.2 introduces the proposed framework, section 5.3 details the proposed cell localisation algorithm and section 5.4 explains the feature extraction. The classification result for the leukaemia cell-lines is provided in section 5.5. The publications based on the work are provided in section 5.6 followed by the conclusion in section 5.7.

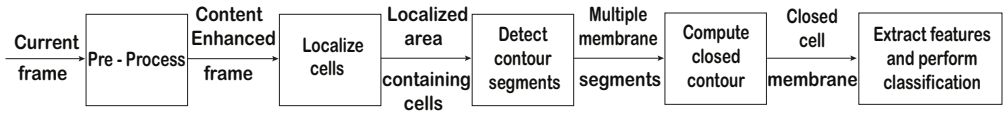

Figure 5.1: Overview of the framework for processing cells in IFC

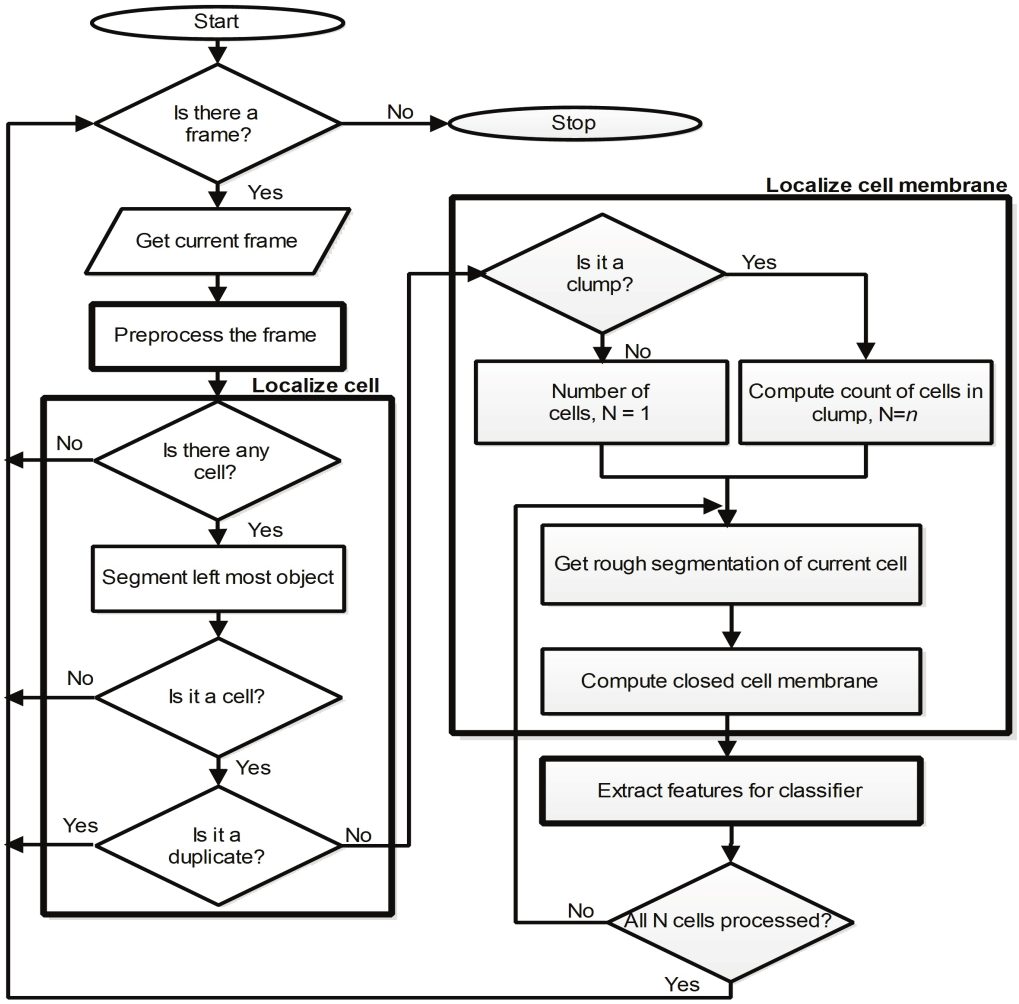

Figure 5.2: Block diagram for processing cells in generic IFC system.

5.2 The Framework

The overview of the proposed system for automated image processing is shown as a block diagram in Fig. 5.1 and is detailed in the flowchart in Fig. 5.2. The important blocks are pre-processing, object localisation, cell contour localisation, extraction of features and subsequent classification. Note that the entire procedure, though general, is going to be explained by taking leukaemia cell-line K562, MOLT and HL60. The video stream of these cell-lines are captured, while they are in flow, using the mIFC technique as detailed in chapter 4. The instrument setup and imaging details can be found in section 4.2.

5.2.1 Pre-processing

The captured frames in the recorded raw video may contain noise, which may affect the subsequent image processing. Background subtraction has been used to filter out this noise. Background images can be found by imaging the channels with sheath fluid but before pumping the cells. In order to reduce the effect of noise in the image due to cell debris, a 5×5 averaging mask is applied to both foreground and background images (Fig. 5.3) prior to background subtraction. The effect of background subtraction is shown in Fig. 5.4 **a**.

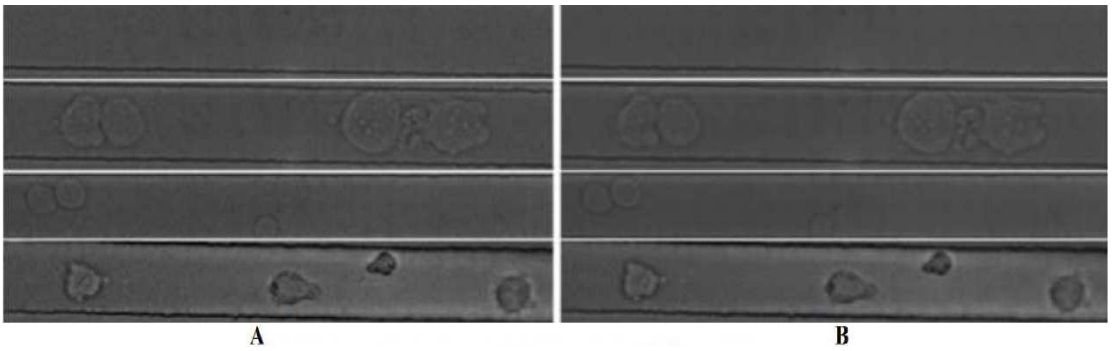

Figure 5.3: a) Frames in each row contain respectively the background, K562, MOLT and HL60; b) frames after smoothing.

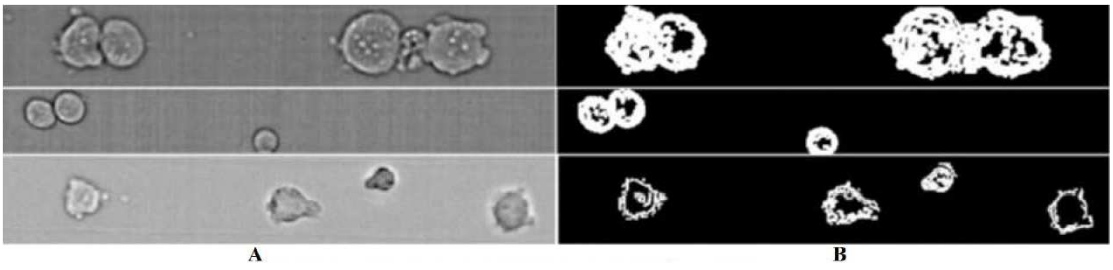

Figure 5.4: a) Background subtracted frames; b) density images.

The next step is to localise the cells from the current frame. All the frames are then processed to check for the presence of cells. Frames not containing any cells are filtered out at this step. The detection of frames devoid of cells is done using a pre-computed threshold, T_{es}. In order to find this threshold, K background subtracted frames not containing any cells are chosen and their Sobel edge sensitivity thresholds (Th^i_{back}) are determined (Sobel (2014)). Similarly K background subtracted frames each containing only one cell are chosen and their edge sensitivity thresholds (Th^i_{fore}) are also found. The final threshold, T_{es} is then defined as the average of the maximum from the set Th^i_{back} and the minimum from the set Th^i_{fore}.

In our implementation, we have chosen K as 10. Any frame with edge threshold less than T_{es} does not contain any cell and are excluded from further processing. Note that, the brightness of the light source would have only minimal effect on the threshold, as background subtracted images have been used in deciding the threshold.

$$T_{es} = \frac{1}{2}\left(\max(Th^i_{back}) + \min(Th^i_{fore})\right); \forall i = 1 : K \tag{5.1}$$

Where

$$Th^i_C = \sqrt{4\left[\frac{1}{PQ}\sum_{m=1}^{P}\sum_{n=1}^{Q}\left\{G^2_X(m,n) + G^2_Y(m,n)\right\}\right]}$$

In Eq. 5.1, G_X and G_Y are the X directional and Y directional gradient components for the i^{th} image using corresponding normalised Sobel operators. Th^i_C represents the threshold computed for the i^{th} image in the C category. i.e., background or foreground category.

The set of background as well as single cell foreground frames for the computation of the threshold T_{es} can be taken manually. It can also be identified automatically by using first few frames. For the automatic identification, initially each frame is assumed to have cells and is processed through section 5.2.2 and 5.3.1. If there are no cells in a frame, due to small amount of noise present throughout the frames, the morphological closing followed by the filling on the Sobel edge image result in an almost completely filled frame. Such a frame is taken as a candidate for background frame. On the other hand, due to the higher gradient magnitude around the cell (and hence large Sobel sensitivity threshold compared to the frame without containing any cells), a frame with single cell is identified (in section 5.3.1) as a frame containing only one object and with single centroid. The set of background and foreground frames thus identified are then used to compute the threshold T_{es} and is used in subsequent processing in order to avoid unnecessary computations on a frame not containing any cell.

5.2.2 Identification of regions of interest

In a frame containing a cell, the regions of interest (in which the cell is present) have to be identified. The background subtracted frame is first averaged with a 3×3 mask and is then thresholded (Otsu (1979)) to obtain the density image (Fig. 5.4 **b**). Whenever a cell enters into a frame, it has to be identified as an object of interest and is roughly localised by identifying the rectangular region covering the left most object in the current density frame. This region is then morphologically (Najman and Ronse (2005)) closed and filled to get the object of interest (Fig. 5.5 **a**). Objects identified in this step are further processed only if their size is greater than the minimum possible size of the cells under consideration. Note that, the left most object is selected as a matter of convenience. Picking the left most object ensures that all cells are considered, as whenever a new cell enters, it is marked as the left most object; this being dependent on the fluid flow direction through the micro-channels, and is left to right in these experiments.

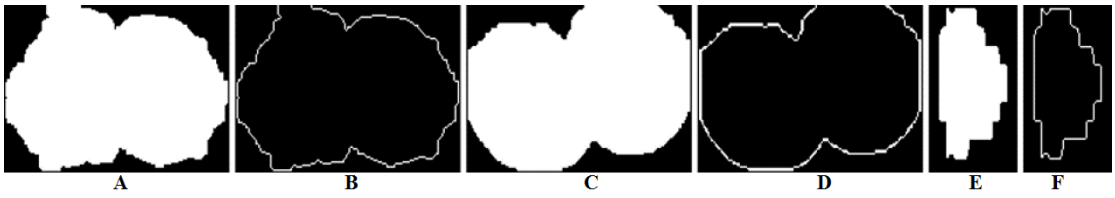

Figure 5.5: The object of interest and their outline by filling holes on the morphologically closed left most objects shown in Fig. 5.4.

Depending upon the velocity of the cell and the frame-rate (of acquisition), a single cell may be imaged multiple times. Assuming cells move from left side to right side of the frame, a cell present in the left half of the current frame may appear in the right half of the next frame. In order to avoid unnecessary computation on the same cell (appearing in different frames), it is essential to check if the cell has been imaged in the previous frame. A duplicate object is identified by comparing the region properties (area, perimeter, centroid and the area of bounding box) of the binary object of interest (Fig. 5.5) in the current frame with that of the object of interest in the previous frame. The centroid is measured with respect to the bounding box of the binary object. Being the fluid flow laminar, the cells move in nearly straight and never migrate across different stream lines. This ensures that the region properties computed for the same cell

but in adjacent frame to remain very close. Ideally, the Euclidean distance has to be zero for the duplicates, and in our implementation, we have used a threshold of 0.1. If the position of the object of interest in the frame falls left to the position of the object of interest in the previous frame, it is directly identified as a new object without checking aforementioned region properties; as the same cell cannot move in the opposite direction to that of fluid flow. The non-duplicate objects identified in this step are then further processed. The threshold 0.1 was experimentally found out by considering the region properties of 40 pairs of duplicate as well as non-duplicate cells. The problem of duplicate imaging of a cell can be mitigated with the use of flow focusing, so as to ensure the velocity of cells remains fairly consistent. The FoV for image acquisition is appropriately adjusted so that image of each cell is acquired only once, as they flow at a given velocity.

5.3 Proposed Cell Localization Algorithm

The cell localisation step involves the identification of the location of each non-duplicate cell. It is a very important step as the subsequent feature extraction and classification (accuracy) highly depend on the exact localisation/segmentation of the cells. In IFC, though it is rare, a few cells can form a clump. Hence, it is essential to detect clumps of multiple cells and localise every cells in the clump by identifying cell contours. Following subsections explain clump detection, rough segmentation of cells, and exact localisation of the cell by identifying its contour.

5.3.1 Clump identification and single cell localisation

The first step in clump identification is the computation of the distance transform (Fabbri et al. (2008)) for the binary object of interest. The distance transform compute the shortest possible distance of each object pixel from the background. The next step is non-maxima suppression on the output of the distance transform to produce a binary image M with all maxima points as 1 (ON) and other points as 0 (OFF). If the corresponding distance of an ON point (obtained from the distance transformed image) is greater than or equal to the cell radius T_{DT} of the smallest possible cell under consideration, then the given ON point is identified as a

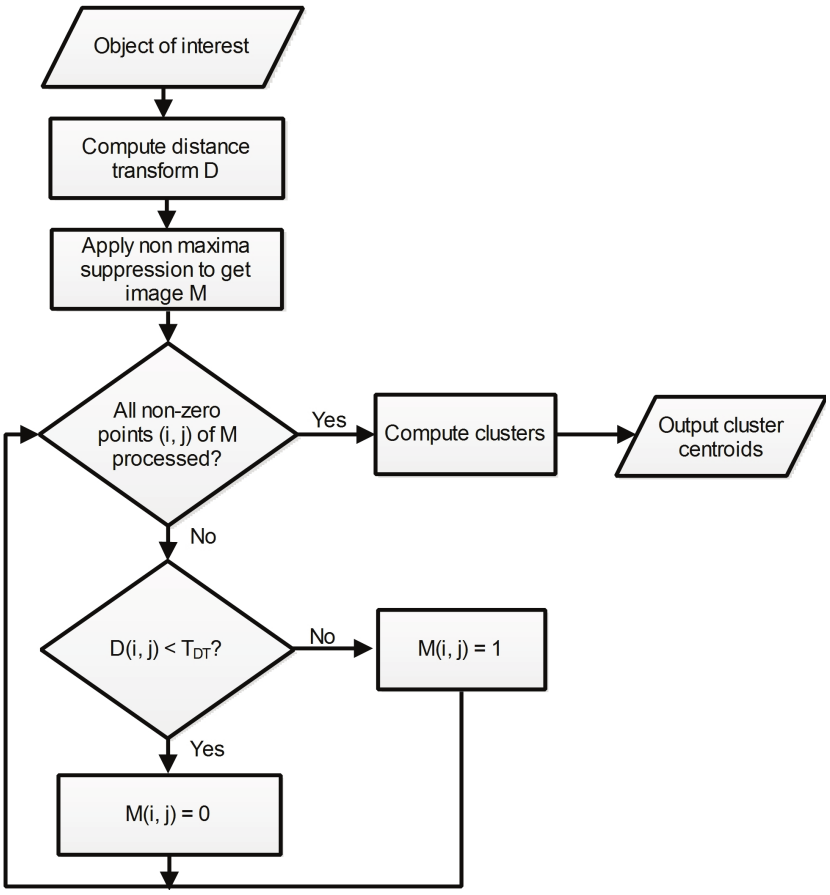

Figure 5.6: Flowchart showing the procedure to find the centroids of the cells in the clump.

probable centroid. These probable centroids will be close to the actual centroids, because of the near circular (in fact, spherical) shape of the cells. The spatial clustering of these points will give the actual cell centroids in the clump. This is done by an adaptive version of the K-means clustering (Hartigan (1975)) (which determines number of clusters automatically) to give the final centroid for each cells in the clump. The entire procedure is shown as flowcharts in Fig. 5.6 and Fig. 5.7. The flowchart in Fig. 5.6 takes the object of interest as input and finds an intermediate image 'M' with suspected centroids as ON points. Note that after the non-maxima suppression, the distance from cell boundary of each maxima point 'D' is compared with a threshold T_{DT} to decide whether it is really a good candidate for the cell centroid. The *'Compute clusters'* block takes these suspected qualified centroids as input, and clusters spatially adjacent centroids to output only valid centroids. This clustering block is expanded and shown as flowchart in Fig. 5.7.

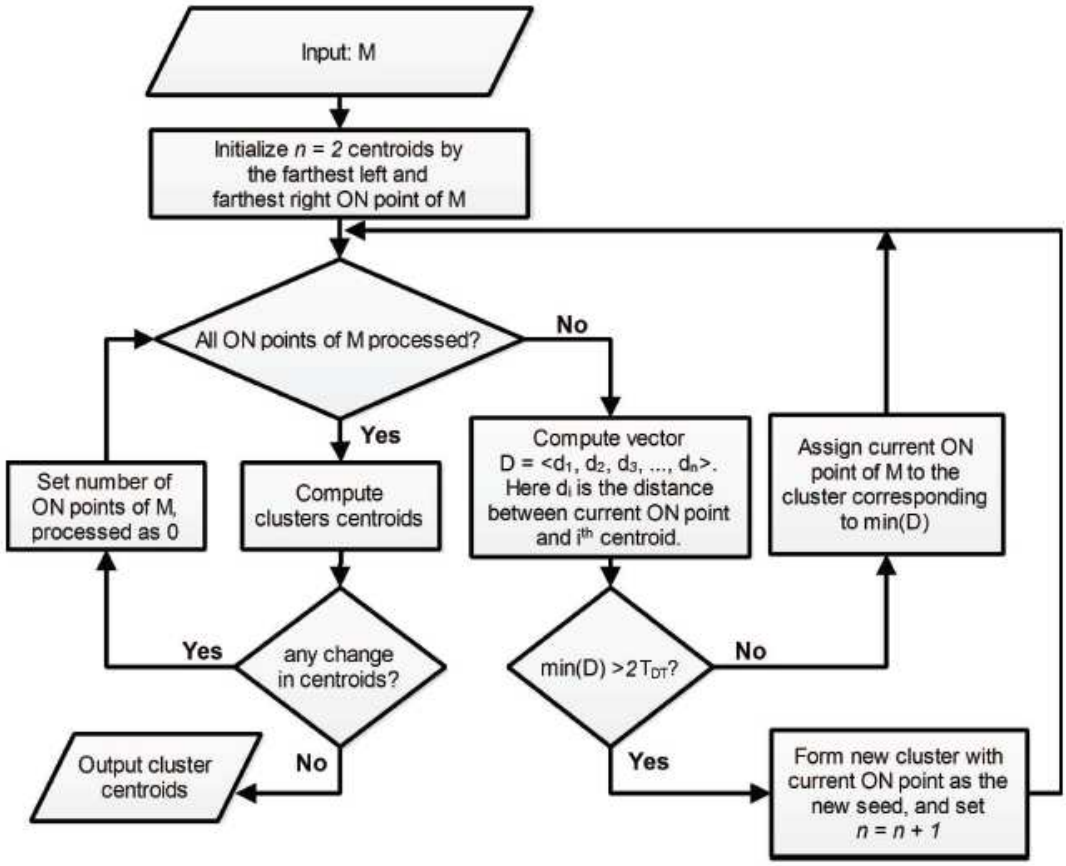

Figure 5.7: The block named **'Compute clusters'** in Fig. 5.6 is expanded for clustering the centroids.

For the spatial clustering, we assume that there are only two clusters initially whose centroids are taken as the farthermost two probable centroids. All other centroids are assigned to one of these clusters, provided the distance between the representative cluster centroid and the probable centroid is less than the diameter of the smallest possible cell ($2 \times T_{DT}$). If the spatial distance from a probable centroid to every other cluster centroids is greater than $2T_{DT}$, a new cluster is formed with the corresponding probable centroid as the cluster centroid. At the end of each iteration (i.e., one pass through all the probable centroids), the centroids of the clusters are updated as the average of the candidates in each cluster. The clustering procedure is repeated until no new cluster is added. Note that the smallest possible cell size may vary depending upon the type of the cells and based on the imaging set up. In our implementation, for the leukaemia cell lines, we have set $T_{DT} = 20$ pixels. However, since the centroid calculation is determined by the non-maxima suppression on the distance transformed density image, the probable

centroids identified will be closer to true centroids (Fig. 5.8 **c**) giving us greater freedom in selecting this threshold.

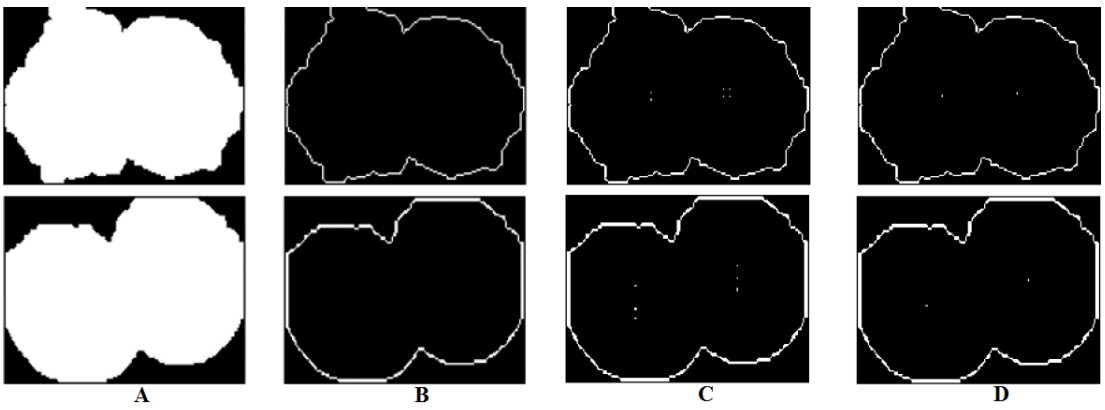

Figure 5.8: a) Cell clump K562 (top row), and MOLT (bottom row); b) their outline; c) clumps with suspected centroids; d) clump with final cell centroids after clustering.

The result of the probable centroid identification is shown in Fig. 5.8 **c**. Here, 2 probable centroids are identified for the left cell and 4 probable centroids for the right cell of the K562 clump. Similarly 3 probable centroids are identified for both left and right cells of the MOLT cell clump.

5.3.2 Rough segmentation of cells

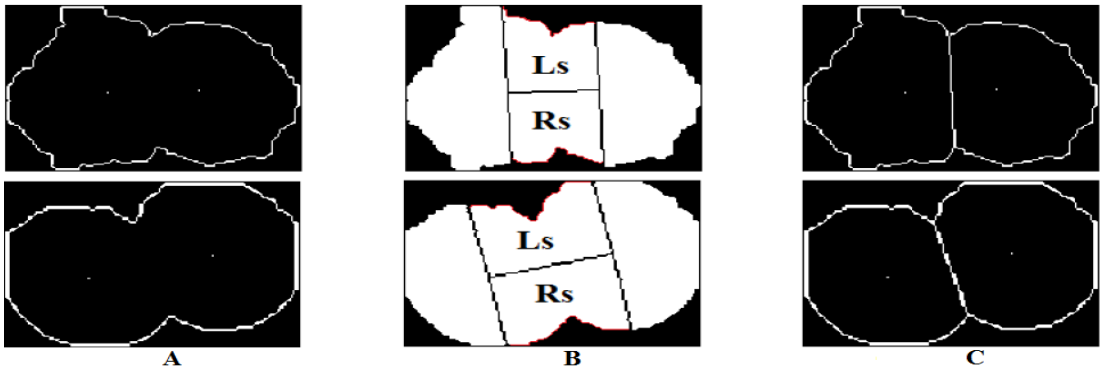

Figure 5.9: a) Cell clumps along with centroids; b) restricted Ls and Rs areas (on both sides of the line connecting the centroids); c) rough segmentation of the cells.

A rough boundary (line) which appropriately segments the clump into different cells has to be identified. Two lines perpendicular to the line joining the centroids

are used to divide the clump in to two segments (Left segment (Ls), Right segment (Rs) as shown in Fig. 5.9 **b**). The outline points lying on these two segments form the search space (shown as red pixels in Fig. 5.9 **b**) to identify the end points of the rough boundary. In fact, the shortest line connecting the Ls and Rs segments forms the rough segmentation boundary. Since the segmentation strategy is first to find the correct centroids and then find rough boundary between every two adjacent centroids, the method works even for clumps containing more than 2 cells.

5.3.3 Exact localisation of cells

Following the rough segmentation of the cell, a contour enclosing the cell has to be identified. The Canny (Canny (1986)) edges detect the cell contour but include edges corresponding to internal organelles as well (Fig. 5.13 **b**). Sometimes the cell contour may be broken at multiple locations and hence a good cell segmentation algorithm is necessary to localise the accurate contour. It is also necessary to extract correct cellular features, since the subsequent feature extraction greatly depends on the contour. A simple method, such as morphologically closing the Canny edge of the cell followed by finding the boundary of the filled cell image often results in bad segmentation (Fig. 5.10 **a**, **c**). Snake based segmentation is also not a feasible option primarily because of the throughput due to the iterative nature of the algorithm. Also, the accuracy as well as the convergence time of the snake based algorithms greatly depends on the initial contour approximation. We are proposing a contour detection algorithm similar to Kubota (2010, 2012). However the proposed algorithm differs in two aspects to its original formulation. The new algorithm is fully automatic, ruling out any manual intervention and it uses only a reduced set of vertices in the graph formulation. The automatic identification of the cell contour is achieved by proposing a way to identify valid starting and finishing points on the contour and then following a shortest weighted path surrounding the cell centroid. The result of segmentation by the proposed algorithm on a HL60 cell and on a K562 cell are shown in Fig. 5.10. For simplicity, we are going to explain the algorithm in detail by finding the contour of an isolated K562 cell. However the method works equally well on the roughly

localised cells from the clump. The result of cell localisation by the same method for the clumps that we have considered earlier is shown in Fig. 5.12. In order to get the correct closed cell contour, we define a directed, weighted graph from the Canny edge image (Fig. 5.13 **b**) such that the closed cell contour lies along the shortest weighted path from any starting point on the original curve segment to an immediate point. Rather than making use of all edge points, only a subset of points is used to define the graph thereby saving computing power. The steps involved in the localisation of the cell are shown as a block diagram (Fig. 5.11) and are detailed in the following subsections.

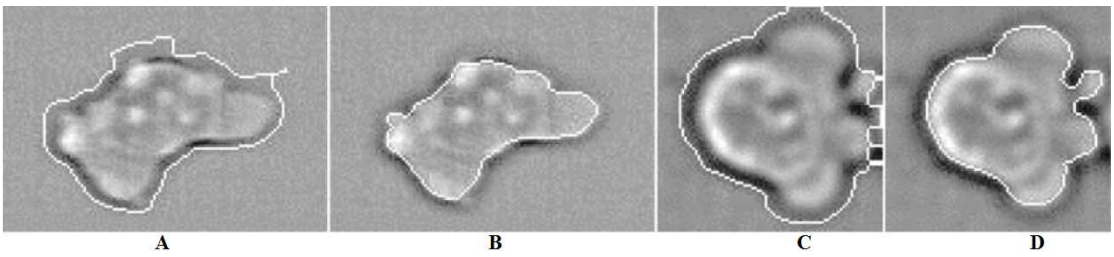

Figure 5.10: a), c) cell contours (Hl60 & K562) identified by finding the boundary of filled binary cell image; b), d) cell contour identified by the proposed algorithm

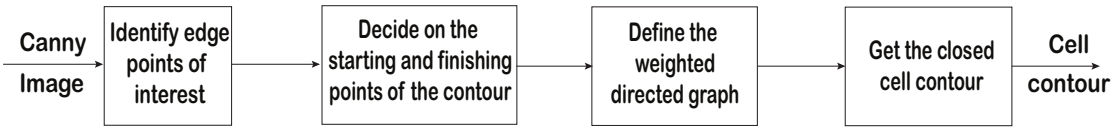

Figure 5.11: Steps involved in the localisation of cell contour

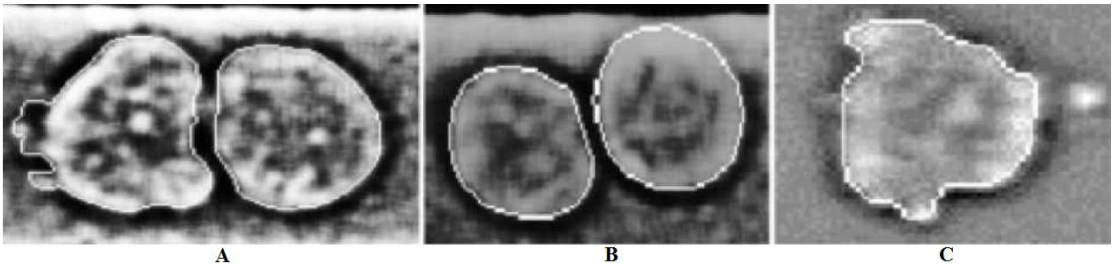

Figure 5.12: The final segmented out cells from the objects of interest in Fig. 5.4: a) K562 b) MOLT and c) HL60

Identifying the edge points of interest

The search space for the actual curve segments of the contour can be limited by selectively masking out Canny edges of the roughly segmented background subtracted image (Fig. 5.13 **a**). This image is dilated, filled and then eroded with disk structuring element of radius 3 to get an external mask. This mask is going to be a binary image with true value at cell region and false value outside (i.e, the image in Fig. 5.13 **c** with its central hole filled). An internal mask is derived by eroding the external mask with the same disk structuring element to generate the edge filtration mask shown in Fig. 5.13 **c**. The edge filtration mask is generated by subtracting the internal mask from the external mask ($ExtMask$ **&** $\neg IntMask$). Here '**&**' represents logical AND operation and '$\neg$' represents logical negation. The edges corresponding to internal organelles are then eliminated by doing logical AND operation between the edge image shown in Fig. 5.13 **b** and the edge filtration mask shown in Fig. 5.13 **c**. The result is shown in Fig. 5.13 **d** which now contains the edges of interest for the contour.

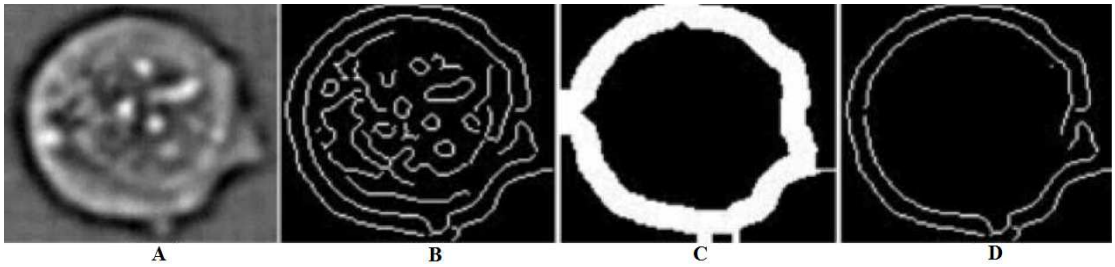

Figure 5.13: a) Background subtracted image of a K562 cell; b) Canny edges; c) mask computed to discard unnecessary edge points; d) candidate edges after the masking operating

Decide on the starting and finishing points of the contour

In order to get accurate cell contour, as noted, a weighted directed graph is defined such that the closed cell contour lies along the shortest weighted path from any good point on the curve to an immediate point along the curve. Clearly, the correctness of the contour detected will largely depend on whether the starting point is selected from the right curve segment or not. The imaging modality employed in our case is bright-field transmission microscopy. The regions surrounding the

edges appear darker in this modality due to the finite numerical aperture of the objective, and consequently the scattered light from the object edges escapes out of the collection-cone of the imaging system. Therefore, it was observed that the region just outside the cell wall appears darker in a background subtracted image. Hence an edge segment in the Canny image that has sufficiently large number of dark pixels lying just outside the edge pixels (with respect to the centroid) is a good candidate for selecting a starting point. In this implementation, segments with number of pixels greater than 75% of the longest segment are selected as probable candidates. For each pixel on these probable candidate segments, two closest pixels that are lying but after the pixel in the direction of the line connecting the centroid of the cell to the pixel on the segment are considered as radially outward pixels. The average intensity (T_{In}) of all these radially outward pixels for each segment is then determined. The segment with minimum T_{In} is selected and any point on this segment whose outward pixel intensity less than the segment's T_{In} is identified as the starting point (S^p) of the actual contour. The radially outward vector from the centroid of the cell to the starting point is identified as the start vector. A neighbouring point from the same segment with the corresponding vector forming almost 360 degrees with respect to the start vector in the anticlockwise direction is identified as the finishing point (F^p) (Fig. 5.13 **b**). The F^p is selected by deliberately making a pixel break between F^p and S^p so that they now lie in two different segments. Pseudo code in 5.2 explains how to find radially outward pixel (X, Y) for a point $(x(i), y(i))$ on a cell segment, given the centroid of the cell (x_c, y_c).

$Slope = \frac{y(i)-y_c}{x(i)-x_c}$

If $|Slope| < 1$

$$Y = y(i) + Slope * (X - x(i))$$

Where

$$X = \begin{cases} x(i) - 1 & \text{;if } x(i) < x_c \\ x(i) + 1 & \text{;if } x(i) \geq x_c \end{cases} \tag{5.2}$$

else

$$X = x(i) + \frac{1}{Slope} * (Y - y(i))$$

Where

$$Y = \begin{cases} y(i) - 1 & ;\text{if } y(i) < y_c \\ y(i) + 1 & ;\text{if } y(i) \geq y_c \end{cases}$$

Define the weighted directed graph

All the endpoints and the branch points of the edge image (Fig. 5.13 **d**) are identified as the vertices of the graph. An endpoint is a point with less than 2 neighbours while a branch point is the one with more than 2 neighbours. We will also include those points on different segments as potential vertices if they lie too close (within three pixels) to the already considered vertices so that we are not loosing the contour due to spurious projections. The vertices are placed in the increasing order of angle that the corresponding position vector of each vertex subtends with the position vector of the starting point. Note that the position vectors are defined by taking the centroid of the cell as the origin. The finish point is deliberately made at 360 degrees to keep it as the last node. Edges are defined between every pair of vertices, if they lie within an angular closure of θ_1 (100^0) degrees. If two such vertices are connected, the edge between them is formed by the curve segment connecting them. Otherwise the line connecting them defines the edge. In order to avoid selecting a wrong path, short edges are preferred over long edges. Hence no edges are defined if the vertices are separated by an angular closure more than θ_1^0. This is done by assigning an ∞ weight to those edges during the formulation of the graph. In order to make sure that there is at least one edge with in this θ_1 enclosure, single point breaks are introduced at every θ_2^0 ($\theta_2 < \theta_1$). In our implementation, we have selected θ_1 as 100^0 and θ_2 as 90^0. A weight for each edge is then defined, which is zero if the pixels are connected else the squared Euclidean distance between the vertices. Note that, the starting and finishing points as well as the end points created by introducing the break at every θ_2 degrees are also considered as vertices in the definition of the graph.

If $i < j$, and $\theta < \theta_1$; (if angular enclosure between $v(i)$ & S^p at centroid is less

than that made by $v(j)$ and if the enclosure between $v(i)$ & $v(j)$ is within θ_1)

$$W\left[v(i), v(j)\right] = \begin{cases} 0 & ;\text{if } L\left[v(i)\right] = L\left[v(j)\right] \\ \left\|v(i), v(j)\right\|^2 & ;\text{else} \end{cases} \tag{5.3}$$

else $W[v(i), v(j)] = \infty$

Where

$$\theta = cos^{-1}\left(\frac{\langle v_1, v_2\rangle}{\|v_1\|\,\|v_2\|}\right);$$
$$v_1 = \begin{bmatrix} x(i) - x_c \\ y(i) - y_c \end{bmatrix};\ \text{and}\ v_2 = \begin{bmatrix} x(j) - x_c \\ y(j) - y_c \end{bmatrix};$$
$$v(i) = (x(i), y(i))\ \text{and}\ v(j) = (x(j), y(j))$$

In Eq. 5.3, $W[v(i), v(j)]$ is the weight for the edge connecting i^{th} node $v(i)$ and j^{th} node $v(j)$. $L[v(i)]$ is the label assigned to the critical point $v(i)$ such that two critical points share the same label if and only if they are part of the same segment.

Get the closed cell contour

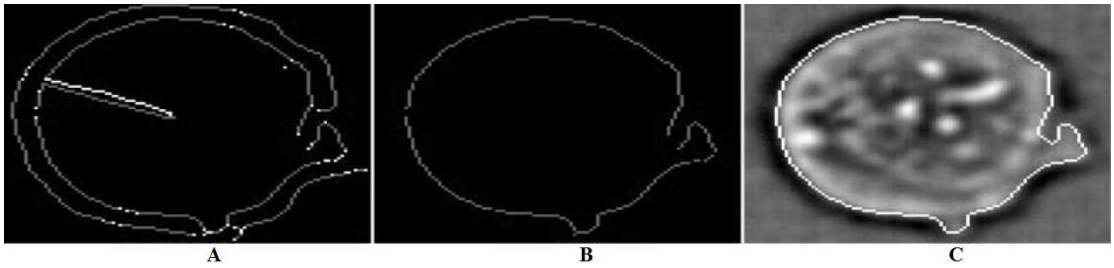

Figure 5.14: a) contours of interest with critical points (White), start vector (Gray) and finish vector (White); b) shortest weighted path; c) closed contour embedded on the cell image.

Once the graph is defined, the curve segments of the contour of interest lies along the longest shortest weighted path from the starting point to the finishing point. Dijkstra's (Dijkstra (1959)) shortest path algorithm is used to find the shortest weighted path. But there can be unconnected curve segments (Fig. 5.14

b) in the path, which need to be connected using the points along the line connecting them. The unconnected curve segments are characterised by an edge between the identified curve segments having non-zero finite weights. The area inside the connected contour is filled, eroded and then dilated with disk structuring element of size 3 to get an intermediate image whose boundary pixels will give the final contour. The closed contour embedded on the cell image is shown in Fig. 5.14 **c**.

5.4 Feature Extraction

In order to characterise the cells, we extract features that capture the size, shape, and complexity. The features such as area, perimeter, and diameter are extracted to reflect the size of cells. In order to quantify the deviation of the shape of cells from circle (or ellipse), eccentricity is used as a measure. In addition to the earlier mentioned size measures, major and minor axes lengths are also used. Solidity is used as a measure of compactness of cells in the convex polygon. The texture features and the number of objects inside the cell are used to infer cell complexity.

5.4.1 Area, Perimeter and Equivalent diameter

The area (AA) is measured by counting total number of pixels on the contour as well as the pixels enclosed by the contour. The perimeter (PM) is determined by counting the number of pixels on the contour. The equivalent diameter (ED) is measured as the diameter of the circle having area AA.

$$ED = \sqrt{\frac{4 \times Area}{\pi}} \tag{5.4}$$

5.4.2 Major axis length, Minor axis length and Eccentricity

Major (MJ) and minor (MN) axes lengths are measured as the axes lengths of the ellipse that has the same normalized second central moments as that of the cell region (Haralick and Shapiro (1992)). Eccentricity (EC) measures how much the cell deviates from being circular. Equations 5.5, 5.6 and 5.7 compute these

measures where (x_i, y_i) are the pixel positions in the cell region.

$$MJ = 2\sqrt{2}\sqrt{U_{xx} + U_{yy} + \sqrt{(U_{xx} - U_{yy})^2 + 4U_{xy}^2}} \tag{5.5}$$

$$MN = 2\sqrt{2}\sqrt{U_{xx} + U_{yy} - \sqrt{(U_{xx} - U_{yy})^2 + 4U_{xy}^2}} \tag{5.6}$$

$$EC = \sqrt{1 - \left(\frac{MN}{MJ}\right)^2} \tag{5.7}$$

Where

$$U_{xx} = \frac{\sum_{i=1}^{N} X_i^2}{N}; U_{yy} = \frac{\sum_{i=1}^{N} Y_i^2}{N}; U_{xy} = \frac{\sum_{i=1}^{N} X_i Y_i}{N};$$

$$X_i = x_i - \bar{X};\ \text{and}\ Y_i = y_i - \bar{Y}; \forall i = 1 : N$$

$$\bar{X} = \frac{1}{N}\sum_{i=1}^{N} x_i;\ \text{and}\ \bar{Y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

5.4.3 Solidity and Convex area

Solidity (SY) is a measure of the proportion of pixels in the convex hull to that of the cell area. It is measured by taking the ratio of cell area to convex area. The convex area is measured by counting the number of pixels in the convex hull.

$$SY = \frac{Area}{ConvexArea} \tag{5.8}$$

5.4.4 Number of objects inside the cell

The number of objects bounded by the contour is determined from the equalized background subtracted image after applying the Otsu's threshold (Fig. 5.15 **b**). Objects that do not share pixels with the outline (Fig. 5.15 **c**) and whose area falls in between a low and high threshold, and those having nearly a circular shape are counted as organelles (Fig. 5.15 **d**). Based on analysing typical sizes of cellular components, we have taken 10 as low area threshold and 80 as high area threshold. The ratio between the major axis length and minor axis length is used as a measure of circularity.

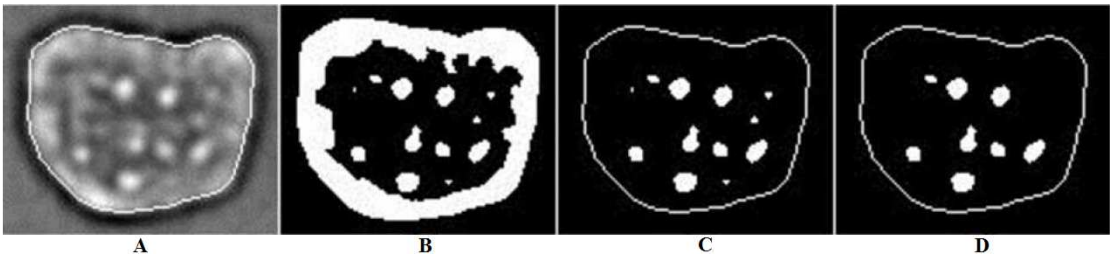

Figure 5.15: a) cell contour superimposed image of a K562 cell; b) Otsu's thresholded image; c) objects that are not sharing pixels with the cell contour; d) final objects of interest

5.4.5 Texture features

Texture features are extracted from the gray level co-occurrence matrix (GLCM) (Haralick et al. (1973)) computed from the pixels in the cell area of the background subtracted image. Each element at position (i, j) in the GLCM specifies how often a pixel with a gray level value j occurs horizontally right adjacent to a pixel with value i. Rather than considering 255 gray levels, we have considered a smaller number of gray levels ($N = 32$). The values in the cell area locations of the background subtracted image is scaled such that each value is 0 to $N-1$, and the $N \times N$ gray level co-occurrence matrix is computed. The computed GLCM is normalized to have sum of its elements as 1. Then, each element at position (i, j) in the normalized gray level co-occurrence matrix (NGLCM) can be thought of as $p(i, j)$, the joint probability occurrence of pixel pairs with a defined spatial relationship having gray level values i, and j in the cell area. The texture properties such as contrast, correlation, energy and homogeneity are then extracted from the NGLCM. The contrast (CT) measures the intensity contrast between a pixel and its neighbours over the cell area. The correlation (CR) measure gives the amount of correlation of a pixel to its neighbours over the cell area. Energy (EG) measures the sum of squared elements in the NGLCM. Homogeneity (HG) is a measure of the closeness of the distribution of elements in the NGLCM and

its diagonal.

$$CT = \sum_{i=1}^{N}\sum_{j=1}^{N} |i-j|^2 p(i,j) \tag{5.9}$$

$$CR = \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{(i-\mu_i)(j-\mu_j)}{\sigma_i \sigma_j} p(i,j) \tag{5.10}$$

$$EG = \sum_{i=1}^{N}\sum_{j=1}^{N} p(i,j)^2 \tag{5.11}$$

$$HG = \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{p(i,j)}{1+|i-j|} \tag{5.12}$$

Where

$$\mu_i = \sum_{i=1}^{N}\sum_{j=1}^{N} ip(i,j)\,;\ \text{and}\ \mu_j = \sum_{i=1}^{N}\sum_{j=1}^{N} jp(i,j)\,;$$
$$\sigma_i = \sum_{i=1}^{N}\sum_{j=1}^{N} (i-\mu_i)^2 p(i,j)\,;\ \sigma_j = \sum_{i=1}^{N}\sum_{j=1}^{N} (j-\mu_j)^2 p(i,j)$$

Note that, the GLCM matrices can be computed by taking any direction, typically horizontal, vertical, along 45 degrees and 135 degrees. Also we can have two possibilities in defining the matrix entries, one that count (1, 2) and (2, 1) differently and the other one contributing a count of 2 to GLCM (1, 2) and GLCM (2, 1) positions. The latter definition leads to a symmetric GLCM. We have computed GLCM considering only the horizontal direction and right adjacency (i.e, just by looking what is the pixel value on right side in the region of interest, thereby considering the non-symmetric GLCM construction). This is because the GLCM features computed using other direction did not bring any extra advantage in the classification task. However, the equations provided for the texture features are general and are applicable to both symmetric and non-symmetric GLCM.

5.5 Case Study: Classification of Leukaemia Cell-lines K562, MOLT and HL60

The effectiveness of the framework is tested using the same leukaemia cell-line dataset that we have considered in chapter 4. The cell lines of HL60, K562 and MOLT were obtained from ATCC (American Type Culture Collection) and are separately cultured in the lab. Thus, each culture contains cells from its own category alone and hence provide the ground truth. Each cell is localised by finding a closed cell contour and the morphological features discussed in the previous section are extracted. Mostly, morphological features, reflecting the size and shape of the cells have been used to classify cancerous cells (Thiran and Macq (1996)). Once the required features are extracted, SVM is used for the classification. Note that, for multi class classification, one against all strategy is followed. i.e., separate SVM classifiers are trained by taking each cell line as positive class and all other samples together as negative class. When tested, if only one of the classifiers claim a sample as positive, the label of the corresponding positive class is assigned to the test sample. If more than one classifier claim a sample as positive, then one of the following two strategies can be adopted. The first strategy is to assign the label based on the distance between the sample and separating hyper plane. The distance from the sample to the separating hyperplane is found and the label of the positive class that produces the maximum distance from the test sample to the separating hyper plane is identified as the correct label. The second strategy is to treat the sample as difficult to classify and assign the label 'Ambiguous'. This will open up the possibility for an expert to further closely examine these cells before arriving at a diagnostic conclusion. The results provided in Tables 5.1 through 5.4 follow the first strategy and Table 5.5 follows the second strategy. We have used Matlab (R2014a) for implementing the classification on a DELL machine with Intel i-3 processor, and having 4 GB RAM. We could analyse 16 cells per second when there was no cell clump and could process 15 cells per second when experimented with cell clumps each containing 2 cells. This includes the time needed for cell localisation, feature extraction and classification.

In this section, we detail the efficiency of classification with different morpho-

Table 5.1: Accuracy on Training Samples (Individual Features)

Index	Feature	Accuracy (%)		
		K562	MOLT	HL60
1	Area	80.65	33.02	81.96
2	Perimeter	77.50	70.60	77.60
3	Equivalent Diameter	80.65	45.28	78.35
4	Major Axis Length	74.19	71.70	76.29
5	Minor Axis Length	75.81	0	94.33
6	Eccentricity	62.10	39.62	57.99
7	Solidity	25.00	86.79	59.54
8	Convex Area	79.84	52.83	79.38
9	Number of Objects	82.26	0	78.09
10	Contrast	91.94	9.50	88.66
11	Correlation	91.94	42.45	77.32
12	Energy	59.68	1.89	80.16
13	Homogeneity	74.19	5.66	79.38

Table 5.2: Training Accuracy for Different Combinations of Features

Feature Combinations	Correctly Classified			Overall
(indices)	K562(124)	MOLT(106)	HL60(388)	Accuracy
Combination – I (FC_1) 1, 3, 4, 7, 9, 11, 12, 13	122	99	381	97.41(%)
Combination – II (FC_2) 1, 3, 5 – 13	124	100	382	98.06(%)
Combination – III (FC_3) 1 – 13	124	100	382	98.06(%)

Table 5.3: Confusion Matrices for Table 5.2

(FC_1)	K562	MOLT	HL60
K562	122	1	1
MOLT	5	99	2
HL60	4	3	381

(FC_2), (FC_3)	K562	MOLT	HL60
K562	124	0	0
MOLT	5	100	1
HL60	3	3	382

metric features (one at a time) and also with the combinations of different features. Table 5.1 shows the accuracy of classification, when individual features are used to classify cells. The complete sample set was used both for training and testing the SVM classifier. From the accuracy values obtained, it is clear that individual features are not sufficient to classify all the three types of cells with good level of accuracy. In order to improve the accuracy of classification, different combinations of features are used to classify cells. Of all the different feature combinations, it is found that the three feature combinations shown in Table 5.2 have produced the top results (with 100% samples for training).

Table 5.3 shows the true positives, true negatives, false positives and false negatives for the samples in each class when the feature combinations considered in Table 5.2 are used for training. The diagonal entries provide the true positives. Non-diagonal entries in each row represent the number of wrongly classified samples. For example, it can be seen from the Table 5.3 (Combination - I) that, out of the 124 K562 cells, 122 are correctly classified and one sample each is misclassified as MOLT and HL60. The result shows that for the K562 class there are 122 true positives, 9 false positives (5+4), 2 false negatives (1+1), and 485 (99+2+3+381) true negatives when tested on 618 cells from all the three classes.

Cross validation experiments are conducted on all feature combinations provided in Table 5.2 by taking 50% training samples and have found that FC_1 has produced the best result. Hence, it was further tested with different sample sets for training. We have chosen 33.3%, 50% and 75% samples for training and for each run the classifier is tested with the remaining set. In each given run, certain part of the sample set (say 33.3%) from each cell-line is randomly taken for training and all other samples are used for testing. Each training and testing run are repeated 200 times and the average level of accuracies, for different percentages of

Table 5.4: SVM – Cross Validation Accuracy (%)

Training Samples	K562	MOLT	HL60	Overall
33.33%	84.47 (5.11)	91.08 (3.58)	95.28 (1.26)	92.39 (1.18)
50.00%	88.29 (4.74)	92.35 (3.27)	95.59 (1.36)	93.57 (1.25)
75.00%	90.90 (5.38)	93.48 (3.92)	95.95 (1.89)	94.52 (1.65)

training sample sets are shown in Table - 5.4. It can be seen that the accuracy has been improved considerably when compared to the classification results shown in Table. 4.1 which was obtained by the PCA signature based method discussed in chapter 4. It can also be noticed from Table 5.4 that even with 33.3% training samples, combination - I provides appreciable classification accuracy. Further investigations carried out on larger populations of different leukaemia cell lines can provide a deeper insight into the ideal feature combination required to identify leukaemia cell. In addition, similar investigations have to be carried out on other cancerous cell lines to identify appropriate feature combinations.

Table 5.5 shows the classification accuracy (in %) obtained when 33% of samples from each class is used for training and tested with the remaining samples. In this case, we have separated out the difficult cases and given the label 'Ambiguous'. As noted earlier, we have followed one-against-all strategy to build SVM models. Being a three class classification problem, there are 3 SVM models each of which is trained by keeping a specific class as positive and the rest as negative. For a test sample, if there is only one claim, the label of the positive class is assigned to it, provided the distance of the test sample is at least 0.8 unit (margin is at 1 units) away from the separating hyperplane. Otherwise the sample is assigned with the label 'Ambiguous'. The mean accuracy and standard deviation for 200 runs are shown in Table 5.5. Note that the classification accuracy is increased considerably but at the cost of keeping the difficult cases as 'Ambiguous'. For example, the first row shows that 72.29% samples of K562 tested are correctly assigned to K562, 1.44% and 1.52% respectively are placed in MOLT and HL60 while 25.17% are assigned the label 'Ambiguous'. The last row shows that on an average only 20.81% samples are assigned the label 'Ambiguous' while 96.05% classified as K562, 98.77% classified as MOLT and 96.32% samples classified as

Table 5.5: Labelling Difficult cases as 'Ambiguous'

	K562	MOLT	HL60	Ambiguous
K562	72.29 (0.06)	1.44 (0.01)	1.52 (0.01)	25.17 (0.06)
MOLT	0.82 (0.01)	75.55 (0.07)	0.08 (0.01)	24.02 (0.07)
HL60	1.72 (0.01)	1.28 (0.01)	78.81 (0.04)	18.71 (0.03)
POSTV	96.05 (2.47)	98.77 (1.96)	96.32 (1.21)	20.81 (2.23)

HL60 are correct.

5.6 Publications

1. **G. Gopakumar**, G.R.K.S. Subrahmanyam, S.S. Gorthi. Morphology based classification of leukaemia cell-lines: K562 and MOLT in a microfluidics based imaging flow cytometer. Ninth Indian conference on computer vision, graphics and image processing (ICVGIP'14), *ACM pp. 34:1–34:7*, 2014.

2. **G. Gopakumar**, V.K. Jagannadh, S.S. Gorthi, G.R.K.S. Subrahmanyam. "Framework for morphometric classification of cells in imaging flow cytometry", *J. Microsc.*, **261**:307–319, 2016.

5.7 Summary

In this chapter, we have proposed a framework for automatically processing cells in microfluidics based imaging flow cytometer. The proposed non-iterative segmentation produced good results with very little false segmentation. The features described capture information such as size, shape and complexity of cells. Using the presented framework, we have demonstrated the classification of three types of leukaemia cell-lines (K562, MOLT and HL60) that are cultured in the lab. The use of cultured cell-lines have provided us with the ground truth and thereby enabled us to evaluate the performance of the proposed framework. We could classify the cell-lines with good accuracy and the results were quite consistent across different cross validation experiments. We hope that imaging flow cytometers equipped

with the proposed framework for image processing would enable cost-effective, automated and reliable disease screening in over-loaded facilities, which cannot afford to hire skilled personnel in large numbers. Such platforms would potentially facilitate screening camps in low income group countries; thereby transforming the current health care paradigms by enabling rapid, automated diagnosis for diseases like cancer.

CHAPTER 6

Cytopathological Image Analysis using Deep Learning Networks in Microfluidic Microscopy

In Chapter 4, we have introduced a low-cost high-throughput microfluidic microscopy instrumentation together with a signature based cell classification for point-of-care diagnosis/screening. Chapter 5 discussed a complete automated framework for processing cells captured using the prototype device. Basically, this framework was a hand-engineered feature based cell classification system and it improved the classification accuracy as reported in chapter 5 for the leukaemia cell-line classification. In this chapter, we explore the possibility of using deep learning networks for cytopathology to improve the results further. We demonstrate that without any conventional fine segmentation and explicit feature extraction, the proposed deep learning algorithms effectively classify the coarsely localised cell lines. The designed deep belief network as well as the deeply pre-trained CNN outperforms the conventionally used decision systems and are important in medical domain where the availability of labeled data is limited for training. We hope that our work enables the development of a clinically significant high-throughput microfluidic microscopy based tool for disease screening/triaging especially in resource limited settings.

6.1 Introduction

Biologists have identified that the power of human brain comes from large number of massively interconnected neurons capable of parallel computation (Deco et al. (2008)). Several Artificial Neural Network (ANN) architectures were studied (SHI and He (2011)) to provide the expertise to the machines, some of them bypassing the need for hand engineered features. A review on the use of artificial neural network in cytopathology can be found in Pouliakis et al. (2016). Traditional

classification systems are typically modelled to contain steps such as cell segmentation, feature extraction, and classification using SVM (Irshad et al. (2014)). For large datasets, SVM takes more time for learning and results in large number of support vectors, particularly when the decision problem is hard. Fully connected normal artificial neural network is also not that feasible to learn such complex decision problems from big dataset. Such problems often requires large number of layers and neurons and hence large number of parameters need to be learned making the learning process slower. Also, the problem of vanishing of gradients in lower layers of ANN using back propagation makes the learning problem further difficult, if not impossible. Another difficulty with ANNs and SVMs is that they need labeled data and the amount of information that the system can learn is restricted by labels. Recently deep learning systems are emerging as reliable and de-facto model for image analysis and many groups across the globe are quickly entering the field and applying these techniques to variety of applications. A recent review on deep learning in medical domain can be found in Greenspan et al. (2016).

As discussed in earlier chapters, mIFC (Barteneva et al. (2012); Lisa et al. (2014); Beers et al. (2014)) is a nascent technology that combines the statistical power of flow cytometry with spatial and quantitative morphology of digital microscopy. We have a low-cost, high-throughput prototype microfluic microscopy system suitable for disease screening in resource limited settings and the proof of concept of the system is presented in chapter 4, where we have used advanced PCA signatures for the classification of leukaemia cell-lines. Further in chapter 5, a general framework for processing cells in mIFC is developed and cell classification based on hand-engineered morphological features of the cells have been carried out. Though the system offered high-throughput at low cost, the reported accuracy had to be improved so as to make it clinically usable. In this chapter, we explore and propose the possibility of using deep learning for cytopathology for a reliable classification system. Though we demonstrate the performance of mIFC based classification system only on three important leukaemia cell lines K562, MOLT and HL60, the approach is quite general as it does not require explicit segmentation and feature extraction. When augmented with other diagnostic modalities, such systems enable early detection of diseases such as cancer.

In real scenarios, more often we will be having large amount of data for training but having a small percentage of them labeled. In order to overcome the difficulty of learning both from labeled as well as unlabeled data, a system that model the structure of the data is needed. We introduce one such model where the idea is to tune the weights to learn a general abstract representation of the structure of the data without considering the labels. This is done by building a Deep Belief Network (DBN) using Restricted Boltzmann Machines (RBM) and capturing the distribution of the training vectors using the parameters of the RBM; the weights and biases. The model keeps the efficiency and simplicity of the gradient method for learning. Also we discuss the use of CNN for cytopathology analysis using a network deeply trained on the popular imaging database ImageNet (Deng et al. (2009); Chatfield et al. (2014)). This also overcomes the need for large labeled data and produce discriminative features for an accurate classification.

The major contributions of this chapter are 1) the proposal of a highly accurate classification framework based on deep learning for unstained, unlabeled IFC data for the first time and is an improvement over the methods/frameworks proposed in chapters 4 and 5, there by moving in a direction one step ahead for a reliable screening/triaging tool for cost-effective disease diagnosis. The classification framework proposed in this chapter does not require fine segmentation and explicit feature extraction unlike the method described in earlier chapters, still producing better classification accuracy. 2) the proposal of a DBN based cell classifier for better accuracy and faster response particularly when the availability of labeled medical data is limited and 3) the finding that the CNN pre-trained on ImageNet database can generate discriminant features leading to very good classification accuracy for the leukaemia cell-lines K562, MOLT and HL60. Just like the DBN based classifier, CNN pre-trained on ImageNet is also very useful in medical field when there is only limited labeled medical data for training (to come up with a trained deep CNN classifier).

This chapter is organised such that section 6.2 proposes a new and simple framework for cytopathological analysis and classification of IFC data using deep learning. Section 6.3 discusses the designed DBN for learning the structure of the data and section 6.4 introduces the CNN ImageNet model to extract discriminative features for classification. Results and discussion are provided in section

6.5 followed by publications in section 6.6. The chapter is concluded with the summary report in section 6.7.

6.2 Framework for Analysis of Cells in Microfluidics based IFC

The image dataset employed for the proposed deep learning based classification is the leukaemia cell-line dataset, and is the one and the same that we had presented in chapter 4. The dataset was captured using the relatively new flow imaging modality microfluidic microscopy (Mf-Ms). As opposed to conventional IFCs, the method leverages unconventional optics and microuidics based sample handling to meet the required imaging throughput and fidelity specifications for cytopathology. As noted in earlier chapters, the dataset contains 618 cells and are localised from the raw video stream of the leukaemia cell lines K562, MOLT and HL60.

The basic steps in proposed framework for making cytopathology decision are pre-processing, rough localisation and classification and are shown in the block diagram in Fig. 6.1. The first major step in automating the cytopathology analysis is segmentation of the cells. Often, segmentation is difficult and computationally intensive. In the proposed approach rather than going for an accurate segmentation, we look for good classification with the features extracted using deep learning networks from the roughly localised cells. The rough segmentation is achieved by finding a rectangular bounding box containing the cell.

Figure 6.1: Block diagram showing overview of the system.

In order to enhance features of cells, a simple preprocessing is done by subtracting the background. A background frame for this purpose can be readily captured by keeping only the sheath fluid in the channel but before pumping the cells. Before subtracting the background both foreground and background frames are filtered using an average mask of size 5×5 to reduce the effect of noise. The

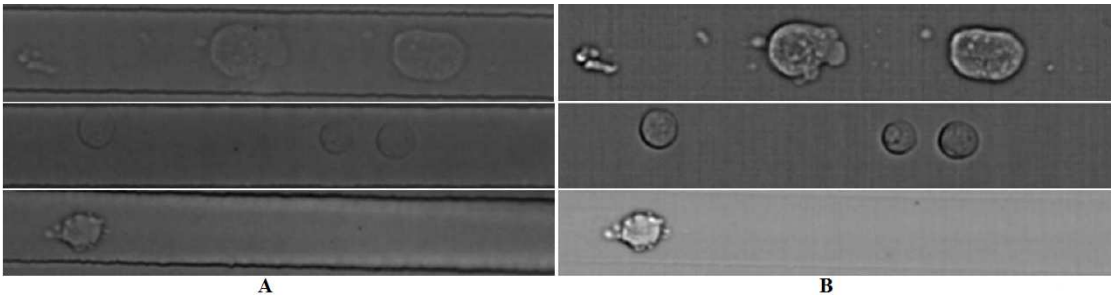

Figure 6.2: a) Frames containing K562 cells (first row), MOLT cells (second row) and HL60 cells (third row); b) the corresponding background subtracted enhanced frames.

Figure 6.3: a) Roughly localised cells from K562 frame in Fig. 6.2; b) bounding box containing left most object from (a), and left most cells from background subtracted frames in Fig. 6.2

effect of the operation is shown in Fig. 6.2. We have roughly localised the cells as discussed in section 5.3 and identified the cell region constrained by a rectangular bounding box. The roughly localised, left most cells from the frames in Fig. 6.2 are shown in Fig. 6.3 **b**.

Recently, deep learning networks based on RBM and CNN are found to be effective in learning complex features for higher level visual recognition task. In subsequent sections we introduce these networks that operate on cell images. The effectiveness in cell classification by these networks is going to be demonstrated with the good results achieved in classifying the leukaemia cell lines HL60, K562 and MOLT. Note that, unlike the framework discussed in chapter 5, the updated framework shown in Fig. 6.1 does not need the fine segmentation and explicit feature extraction and it uses deep learning based classifiers.

6.3 Deep Belief Network for Classification

We design a Deep Belief Network (DBN) for the classification of leukaemia cell lines. A DBN can be thought of as multiple layers of hidden units with connections

between the layers but not between the units within each layer. In DBN, every two adjacent layers except at the fully connected final layer can be treated as Restricted Boltzmann Machine (RBM). Typically DBN is trained in greedy way by training each of the RBM (Hinton (2012)), one at a time starting from the lowest layer. This is an unsupervised training and is going to be discussed in the following subsection. Finally, the weights learned are fine tuned by back-propagation using the available labeled data.

6.3.1 Restricted Boltzmann Machine

Restricted Boltzmann Machine is a popular generative model. A pictorial representation of typical RBM architecture is shown in Fig. 6.4. Note that the connections are undirected and hence bidirectional. In RBM, no connections are made between the hidden units in the hidden layer and no connections are made between the visible units in the visible layer. This helps to reasonably assume that *i*) for a given input at the visible layer, the output at different hidden units are independent and *ii*) for a given output observed at the hidden layer, the output that can be induced at different visible units are also independent. Though RBM can be used as a stand alone classifier (Larochelle et al. (2012)), typically they are used in DBN where a discriminative fine tuning is applied on top of the structure of the data learned by the RBM. It has been shown that generative model learning with RBM improved the discriminative classification result in studying fMRI images of patients recovering from stroke (Schmah et al. (2009)). In Nayak et al. (2013), a variant of RBM is used to classify tumour histopathology images. In Hinton and Salakhutdinov (2006), RBM is used to learn low dimensional effective features and in Brosch and Tam (2013), RBM is used to learn low dimensional effective features in manifold learning of brain MRI. The applicability of RBM in medical data is discussed in Aalto (2014).

Consider the architecture provided in Fig. 6.4. In this particular configuration, there are m visible nodes (v_i) and n hidden nodes (h_j). The visible node biases are labeled as $\{b_i\}_{i=1}^{m}$ while the hidden node biases are $\{c_j\}_{j=1}^{n}$. The energy of this network at any instant can be defined as

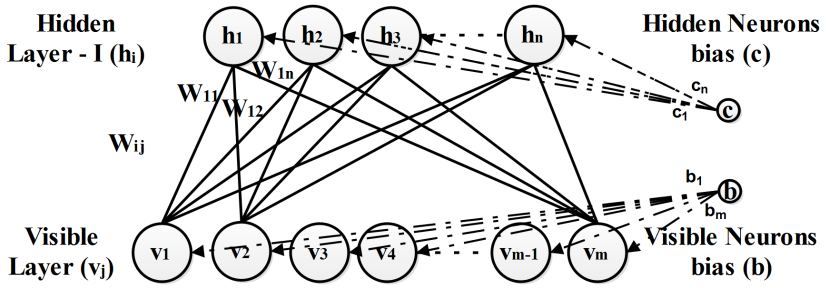

Figure 6.4: Architecture of Restricted Boltzmann Machine

$$E(v,h) = -\left[h^T W v + b^T v + c^T h\right]; \tag{6.1}$$

Where,

$$W = \begin{bmatrix} W_{11} & . & . & W_{1m} \\ . & . & . & . \\ . & . & . & . \\ W_{n1} & . & . & W_{nm} \end{bmatrix}; \tag{6.2}$$

$$b = \begin{bmatrix} b_1 \\ . \\ . \\ b_m \end{bmatrix}; c = \begin{bmatrix} c_1 \\ . \\ . \\ c_n \end{bmatrix}; v = \begin{bmatrix} v_1 \\ . \\ . \\ v_m \end{bmatrix}; h = \begin{bmatrix} h_1 \\ . \\ . \\ h_n \end{bmatrix} \tag{6.3}$$

With a given configuration (of parameters), the RBM can be thought of representing a joint probability distribution for observing v and h together.

$$p(v,h) = \frac{\exp^{-E(v,h)}}{\Sigma_{v',h'} \exp^{-E(v',h')}} \tag{6.4}$$

If there is an algorithm to train a RBM to capture (different aspects of) the probability distribution of the training data (Eq. 6.5), it can be very useful in classification (Fischer and Igel (2014)). With training vectors v and target labels ω to predict, a subset of the factors explaining the distribution of the training vectors explain much of ω, given v. Hence representations that are useful for $p(v)$ tend to be useful when learning $p(\omega|v)$. Learning RBM corresponds to fitting

its parameters such that the distribution represented by the RBM models the distribution underlying the training data, $p(v)$. More specifically, the aim is to find weights and biases that define a Boltzmann distribution (Hinton (2007)) in which the training vectors have high probability.

$$p(v) = \frac{\Sigma_h \exp^{-E(v,h)}}{\Sigma_{v',h'} \exp^{-E(v',h')}} \tag{6.5}$$

As noted, the visible units constitute the first layer of RBM and correspond to the components of an observation (e.g., one visible unit for each pixel of a digital input image). The hidden units model dependencies between the components of observations (e.g., dependencies between the pixels in the images) and can be viewed as non-linear feature detectors (Hinton (2007)). Several such RBM layers can be then stacked up by treating the hidden layer of the lower RBM as input layer to the RBM in the immediate upper layer and can be trained one after the other starting from the lower RBM. By stacking RBMs in this way, one can learn features from features in the hope of arriving at a high-level representation. It was empirically shown that this has produced better feature representations both in terms of classification error (Larochelle et al. (2009)) and in terms of the invariance properties of the learned features (Goodfellow et al. (2009)).

In order to train RBM, we need to maximise the chance of observing the training vectors in the underlying data distribution. This can be done by maximising $p(v)$ which is equivalent to minimising the average negative log likelihood using gradient descent. Let the parameter that we want to learn is θ. The parameter θ can be the weight of the connections W_{ij}, bias of visible nodes b_j or bias of the hidden nodes c_i of RBM.

It turns out that (Bengio (2009))

$$
\begin{aligned}
\frac{\partial \log p(v)}{\partial \theta} &= \frac{\partial}{\partial \theta} \log \left(\frac{\sum_h \exp^{-E(v,h)}}{\sum_{v,h} \exp^{-E(v,h)}} \right) \\
&= \frac{\sum_{v,h} \exp^{-E(v,h)}}{\sum_h \exp^{-E(v,h)}} \left(\frac{\sum_{v,h} \exp^{-E(v,h)} \frac{\partial}{\partial \theta} \sum_h \exp^{-E(v,h)}}{\left(\sum_{v,h} \exp^{-E(v,h)} \right)^2} \right) \\
&- \frac{\sum_{v,h} \exp^{-E(v,h)}}{\sum_h \exp^{-E(v,h)}} \left(\frac{\sum_h \exp^{-E(v,h)} \frac{\partial}{\partial \theta} \sum_{v,h} \exp^{-E(v,h)}}{\left(\sum_{v,h} \exp^{-E(v,h)} \right)^2} \right) \qquad (6.6) \\
&= -\frac{1}{\sum_h \exp^{-E(v,h)}} \sum_h \exp^{-E(v,h)} \frac{\partial E\left(v,h\right)}{\partial \theta} \\
&+ \frac{1}{\sum_{v,h} \exp^{-E(v,h)}} \sum_{v,h} \exp^{-E(v,h)} \frac{\partial E\left(v,h\right)}{\partial \theta} \\
&= -\sum_h p(h|v) \frac{\partial E(v,h)}{\partial \theta} + \sum_{v,h} p(v,h) \frac{\partial E(v,h)}{\partial \theta}
\end{aligned}
$$

$$
\frac{\partial log p(v)}{\partial \theta} = -\mathbb{E}_{h|v} \left[\frac{\partial E(v,h)}{\partial \theta} \right] + \mathbb{E}_{(v,h)} \left[\frac{\partial E(v,h)}{\partial \theta} \right] \qquad (6.7)
$$

Where $\mathbb{E}$ is the expectation operator. It directly follows from Eq. 6.1 that

$$
\frac{\partial E(v,h)}{\partial W_{ij}} = -h_i v_j; \frac{\partial E(v,h)}{\partial b_j} = -v_j; \frac{\partial E(v,h)}{\partial c_i} = -h_i \qquad (6.8)
$$

The two unknown quantities left out in computing the gradient using Eq. 6.6 are $p(h|v)$ and $p(v,h)$.

$$
p(h|v) = \frac{p(v,h)}{\sum_{h'} p(v,h')} \qquad (6.9)
$$

Considering stochastic binary inputs and outputs,

$$
\begin{aligned}
p(h|v) &= \frac{\exp^{\left(h^T W v + b^T v + c^T h\right)} / \Sigma_{v',h'} \exp^{\left(h'^T W v' + b^T v' + c^T h'\right)}}{\sum_{h' \in \{0,1\}^H} \exp^{\left(h'^T W v + b^T v + c^T h'\right)} / \Sigma_{v',h'} \exp^{\left(h'^T W v' + b^T v' + c^T h'\right)}} \quad (6.10)\\
&= \frac{\exp^{\left(h^T W v + c^T h\right)}}{\sum_{h' \in \{0,1\}^H} \exp^{\left(h'^T W v + c^T h'\right)}}\\
&= \frac{\exp^{\left(\sum_i h_i W_i v + c_i h_i\right)}}{\sum_{h'_1 \in \{0,1\}} \sum_{h'_2 \in \{0,1\}} \cdots \sum_{h'_n \in \{0,1\}} \exp^{\left(\sum_i h'_i W_i v + c_i h'_i\right)}}\\
&= \frac{\prod_i \exp^{(h_i W_i v + c_i h_i)}}{\sum_{h'_1 \in \{0,1\}} \sum_{h'_2 \in \{0,1\}} \cdots \sum_{h'_n \in \{0,1\}} \prod_i \exp^{\left(h'_i W_i v + c_i h'_i\right)}}\\
&= \frac{\prod_i \exp^{(h_i W_i v + c_i h_i)}}{\{\sum_{h'_1 \in \{0,1\}} \exp^{\left(h'_1 W_1 v + c_1 h'_1\right)}\} \ldots \{\sum_{h'_n \in \{0,1\}} \exp^{\left(h'_n W_n v + c_n h'_n\right)}\}}\\
&= \frac{\prod_i \exp^{(h_i W_i v + c_i h_i)}}{\prod_i \sum_{h'_i \in \{0,1\}} \exp^{\left(h'_i W_i v + c_i h'_i\right)}}\\
&= \prod_i \frac{\exp^{(h_i W_i v + c_i h_i)}}{\sum_{h'_i \in \{0,1\}} \exp^{\left(h'_i W_i v + c_i h'_i\right)}}\\
&= \prod_i \frac{\exp^{(h_i W_i v + c_i h_i)}}{1 + \exp^{(W_i v + c_i)}}\\
&= \prod_i p(h_i|v)
\end{aligned}
$$

Thus,

$$
\begin{aligned}
p(h_i = 1|v) &= \frac{\exp^{(W_i v + c_i)}}{1 + \exp^{(W_i v + c_i)}} \quad (6.11)\\
&= \frac{1}{1 + \exp^{-(W_i v + c_i)}}\\
&= Sigmoid(W_i v + c_i)
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
p(v_j = 1|h) &= \frac{1}{1 + \exp^{-(b_j + h^T W_j)}} \quad (6.12)\\
&= Sigmoid(b_j + h^T W_j)
\end{aligned}
$$

Where W_i and W_j denote the i^{th} row and j^{th} column of the weight matrix respectively. Note that based on Eq. 6.11, the first term of Eq. 6.6 can be computed

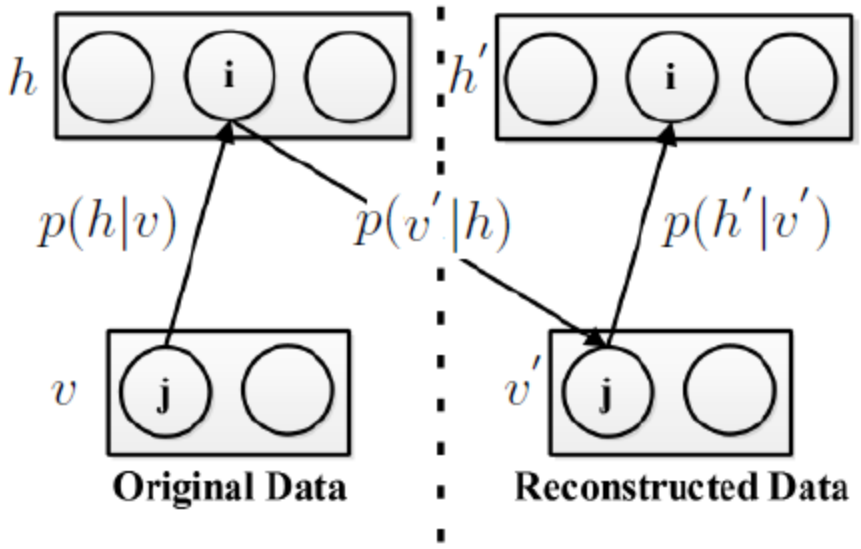

Figure 6.5: Contrastive divergence (CD–1) depiction

analytically. The only term left out in computing the gradient in Eq. 6.6 is the expectation of $\partial E(v,h)/\partial\theta$ over the model distribution $p(v,h)$. Though samples from the model distribution can be generated using Gibbs sampling (Geman and Geman (1984)), computing $p(v,h)$ is intractable due to the partition function at the denominator in Eq. 6.4. Geoffrey Hinton has shown that contrastive divergence (CD) (Hinton (2002)) algorithm can be used to replace this expectation with a point estimate at $p(h'|v')$. Such an estimate captures the direction of the gradient in Eq. 6.6 and works well for all practical applications. A pictorial depiction of the CD algorithm is shown in Fig. 6.5. For every training vector v, compute $p(h|v)$ and get h' by sampling (i.e., For each node in the hidden layer, compute $p(h_i = 1|v)$ using Eq. 6.11 and turn on the node with the computed probability. If the computed probability is greater than a random number selected from a uniformly distributed random variable in the range 0 and 1, make the output value at the corresponding node as 1. Otherwise, set the value as 0. Now compute the value at visible layer and sample v' using the Eq. 6.12. Finally, recompute $p(h' = 1|v')$ using Eq. 6.11). These steps are repeated multiple times (K) and use the final point estimate to replace the model probability $p(v,h)$ in Eq. 6.6. This is K contrastive divergence (CD-K) and Fig. 6.5 shows the depiction of CD for $K = 1$ (CD-1). Thus the main term in computing the gradient in Eq. 6.6 has reduced to $\sum_h p(h|v)\partial E/\partial\theta$, since the $p(h,v)$ in the second term can now be replaced by the point estimate $p(h'|v')$.

$$\begin{aligned}\sum_h p(h|v)\frac{\partial E}{\partial W_{ij}} &= \sum_h p(h|v)(-h_i v_j) \\ &= p(h_i=1|v)(-1\times v_j)+p(h_i=0|v)(-0\times v_j) \\ &= p(h_i=1|v)(-v_j)\end{aligned} \tag{6.13}$$

$$\begin{aligned}\sum_h p(h|v)\frac{\partial E}{\partial b_j} &= \sum_h p(h|v)(-v_j) \\ &= p(h_i=1|v)(-v_j)+p(h_i=0|v)(-v_j) \\ &= \left[p(h_i=1|v)+p(h_i=0|v)\right](-v_j) \\ &= -v_j\end{aligned} \tag{6.14}$$

$$\begin{aligned}\sum_h p(h|v)\frac{\partial E}{\partial c_i} &= \sum_h p(h|v)(-h_i) \\ &= p(h_i=1|v)(-1)+p(h_i=0|v)(-0) \\ &= -p(h_i=1|v)\end{aligned} \tag{6.15}$$

6.3.2 Summary of RBM learning procedure

In the last subsection we have made the necessary derivation needed to train RBM. It turns out from Eq. 6.13 – 6.15 that the gradient for weight (W_{ij}) update is $-[v_j P(h_i|v) - v_j^{'} P(h_i^{'}|v^{'})]$, the gradient for bias update b_j is $-[v_j - v_j^{'}]$ and the gradient for bias c_i is $-[P(h_i|v) - P(h_i^{'}|v^{'})]$. The second term in each of these expressions is due to the expectation over $p(h'|v')$, the point estimate for $p(h, v)$. Also, note that $v^{'}$ is computed using Eq. 6.12 taking vector h computed from Eq. 6.11, for the given input vector v (CD-1). Now, we can use gradient descent to train RBM, where for each epoch we will update based on the average gradient for all training samples. Thus the steps involved in learning RBM are

1. Initialise W_{ij} from samples randomly selected from normal distribution. Also

initialise bias terms (b_j, c_i) to zero.

2. Start with a training vector on the visible units. i.e, the input image of size $m \times n$ is normalised and vectorised. Each component (v_j) is treated as the chance of turning the corresponding visible node ON (set to 1).

3. Update all the hidden units in parallel using Eq. 6.11.

4. Update all the visible units in parallel using Eq. 6.12 to get the reconstruction.

   If the computed probability is greater than a threshold selected from a uniformly distributed random variable which takes values in the range [0, 1], the corresponding node is turned ON. i.e., the nodes are turned ON (set to 1) with the computed probability.

5. Now, try to reproduce h from the reconstructed input $v_j^{'}s$ using Eq. 6.11. Let $h_i^{'}s$ denote these reconstructed nodes.

6. Update the weights and biases (W_{ij}, b_j, c_i) as shown below.

$$\begin{aligned} W_{ij}^{t+1} &= W_{ij}^{t} + \eta \Delta W_{ij}^{t+1} \\ b_j^{t+1} &= b_j^{t} + \eta \Delta b_j^{t+1} \\ c_i^{t+1} &= c_i^{t} + \eta \Delta c_i^{t+1} \\ \Delta W_{ij}^{t+1} &= \mu \Delta W_{ij}^{t} + \alpha [v_j P(h_i|v) - v_j^{'} P(h_i^{'}|v^{'})] \\ \Delta b_j^{t+1} &= \mu \Delta b_j^{t} + \beta (v_j - v_j^{'}) \\ \Delta c_i^{t+1} &= \mu \Delta c_i^{t} + \gamma (P(h_i|v) - P(h_i^{'}|v^{'})) \end{aligned} \tag{6.16}$$

Here η is the learning parameter, μ is the momentum term and α, β, γ are the parameters that decide the weights on the corresponding gradient. In our implementation the parameters are set by empirical estimation and are $\mu = 0.5$, $\alpha = \beta = \gamma = 0.1$

6.3.3 Design and implementation of DBN

The designed DBN for classification of Leukaemia cell lines is shown in Fig. 6.6. The system is trained by considering 3 RBMs and a final fully connected layer.

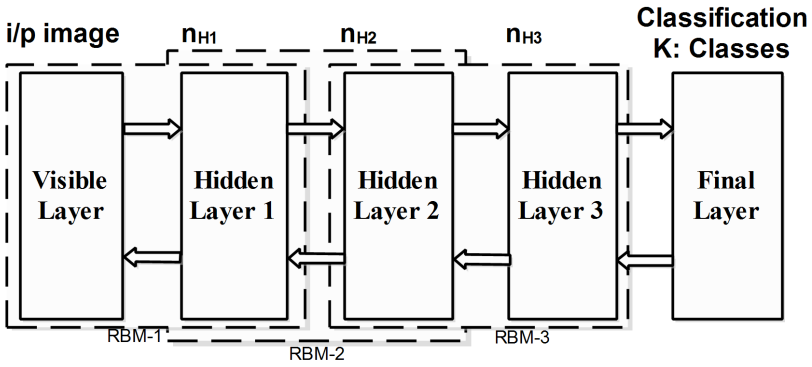

Figure 6.6: The DBN for classification : Depicting the back propagation based fine tuning across the layers

Thus, it has a visible layer, three hidden layers and an output layer. The number of neurons in the visible layer is made same as the number of pixels in the image. Assuming that the input image is of size $m \times n$, the number of neurons is made as mn. In our implementation all roughly segmented cells are made to standard size $m \times n$. The parameters $m = 52$ and $n = 48$ are selected (These are the mean number of rows and columns observed for the roughly segmented cell images). Being a three class classification problem, there are 3 neurons in the output layer. The number of neurons in the hidden layer are $n_{H1} = 100$, $n_{H2} = 300$, and $n_{H3} = 1000$. The number of RBMs in DBN is fixed as 3 for our cell-line dataset after experimenting with 1, 2, 3, and 4 layers of RBMs in the stack. The number of hidden neurons are also fixed in a similar way. This configuration is then trained by choosing different percentage of input data.

Each of the first 3 RBMs are trained independently starting from the first RBM considering all the data (labeled as well as unlabeled) available for training. The algorithm explained in section 6.3.2 is used to train the RBMs. Finally with the labeled data available for training, the parameters are fine tuned. This is done by back propagation algorithm, treating the architecture as a feed forward neural network with 5 layers. As noted earlier, the hidden layer of RBM produces non-linear features and since the subsequent layers operate on these features, they in turn produce high-level feature representations. All the training data is used to learn these features irrespective of whether the samples are labeled or not. It is on top of this pre-trained network, we apply the discriminative fine tuning using the target labels. Since the network parameters are already set during pre-training,

using RBMs in this way overcome some of the problems (such as undesired local minima and slow learning) in learning multi-layer feed forward neural network with backpropagation (Hinton and Salakhutdinov (2006)). The implemented model is tested for classification accuracy, training and testing time. The DBN is also tested for its effectiveness in learning from a mixture of labeled and unlabeled data. The results are discussed in section 6.5.

6.4 CNN as Feature Extractor

As discussed in section 3.3.2, a CNN is a feed-forward artificial neural network mapping an input vector X into an output vector Y. These types of neural networks has proved their greater ability to surpass the skilled human in certain classification task (He et al. (2015)). The discriminative power of CNN is used in a number of medical cases such as in classifying breast tissues (Sahiner et al. (1996); Xu et al. (2014)) and detecting micro calcifications on mammograms (Lo et al. (1995)), classifying interstitial lung disease (Li et al. (2014)), detecting pathologic cases in chest Xrays (Bar et al. (2015)), in thyroid cytopathology (Kim et al. (2016)) and detecting lung nodules in chest radiographs (Lo et al. (1995)). Cireşan has reported the use of deep max-pooling CNN for detecting mitosis in breast histology images (Cireşan et al. (2013)). In all these cases CNN is used as a classifier. We have already discussed about the building blocks of CNN and used it as a classifier in Chapter 3. One of the difficulties in training a CNN for a classification task is that it needs a large dataset. The problem that we are going to address is the Leukemia cell-line classification for which we have only a small dataset. It has only 618 cells (124 K562, 106 MOLT & 388 HL60) which is insufficient to train a CNN classifier. In this section, we discuss the transfer learning capability of the CNN. In transfer learning setting, the knowledge that a CNN has learned for a relatively complex classification task using a large dataset is effectively transferred to a completely different setting. In such setting, CNN is used as a feature extractor and not as a classifier. These features are then used to train a classifier like SVM that needs only training data of moderate size. The transfer learning capability of CNN is studied in Zhang et al. (2015) and utilized in Zeng et al. (2015) to annotate the gene expression patterns in mouse brain.

As noted in section 3.3.2, the main building blocks of a CNN are i) Convolution ii) ReLU and iii) Sub sampling. We will revisit each of these blocks with an emphasis to give intuitive idea on the transfer learning capability of CNN.

Convolutional Layer : For every convolution layer, a number of Kernels are learned during training. A general discussion on CNN learning can be found in Appendix A. Normal convolution is performed between the learned kernels and the input instance but select only valid part of the convolution. This procedure is depicted in Fig. 6.7. We know that the kernels can extract features by convolutions. For example, Sobel kernel find edges, Laplacian kernel detect blobs. Each of these kernel is applied only locally and extract features. Thus depending on the kernel, convolution can extract features like edges, blobs, corners, etc. and are valid features for any images. In CNN, the only learned parameters are the kernel weights and biases, and hence we are learning very local feature detectors rather than the actual features. Suppose that we are learning a CNN for a complex classification task and we have millions of images to learn the classification problem. If we have effectively learned the CNN from such a large training set, it is reasonable to believe that the learned kernels have the capability to extract an exhaustive set of features even to capture the small inter class variability on the original classification problem. Since the kernels learned acts very locally, we can believe that these exhaustive sets of feature detectors are valid for any images irrespective of the classification problem that we are addressing. This fact is the corner stone of transfer learning capability of CNN.

Rectified Linear Unit : ReLU as explained in section 3.3.2 acts as a non linear activation function and it usually follows the convolution block. This will introduce the non-linearity on the output map. Since the subsequent convolutions operate on this non-linear output map, ReLU in turn help to generate non-linear features. It will also enable faster learning by avoiding the problem of vanishing gradients during backpropagation especially at lower layers of CNN (Rohan (2016)).

Sub-sampling : The sub sampling layer helps to learn relevant feature detectors by ignoring small amount of intra class variability in noise, shift and distortion of

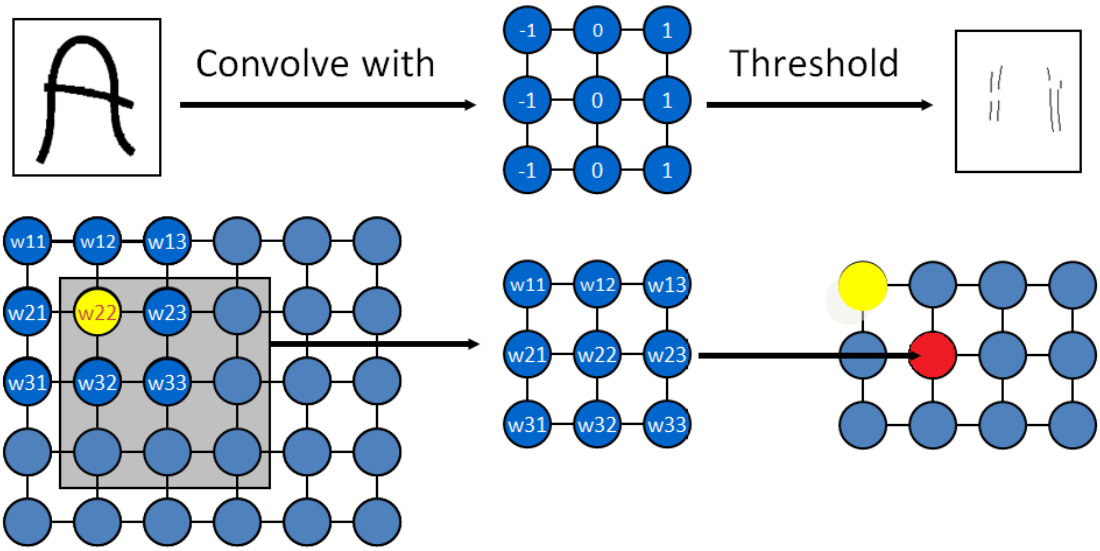

Figure 6.7: Convolution operation: 3×3 kernel on 5×6 image producing an output map of size 3×4

the training samples. The operation is depicted in Fig. 6.8 where sub-sampling by two is shown. It can be seen that the distortion between the two 'A's has reduced as we move for sub-sampling which has reduced the intra class variability between them. Note that these sub-sampling can be done either by taking average of the neighbourhood (average pooling), or by picking max value from the neighbourhood (max pooling).

In a nutshell, a heavily trained CNN for a complex classification task using a large image data set, must have learned an exhaustive set of feature detectors that can capture very local features valid for any images, and have the capability to introduce non-linearity in the detected features due to the architecture involving intermediate ReLU and sub-sampling layers. This enables us to use such a CNN as a feature extractor for our leukaemia cell-line classification problem, and then use these features to build a suitable classifier using the available small training set that we have.

Trained CNN can outperform human in certain classification tasks (He et al. (2015)). As noted, the only bottleneck in using CNN is the need for large amount of labeled data for training, which is often limited in medical domain. In this section, we explore the possibility of doing medical image analysis using a CNN pre-trained on a large scale non medical image database, ImageNet; i.e., we wish to investigate

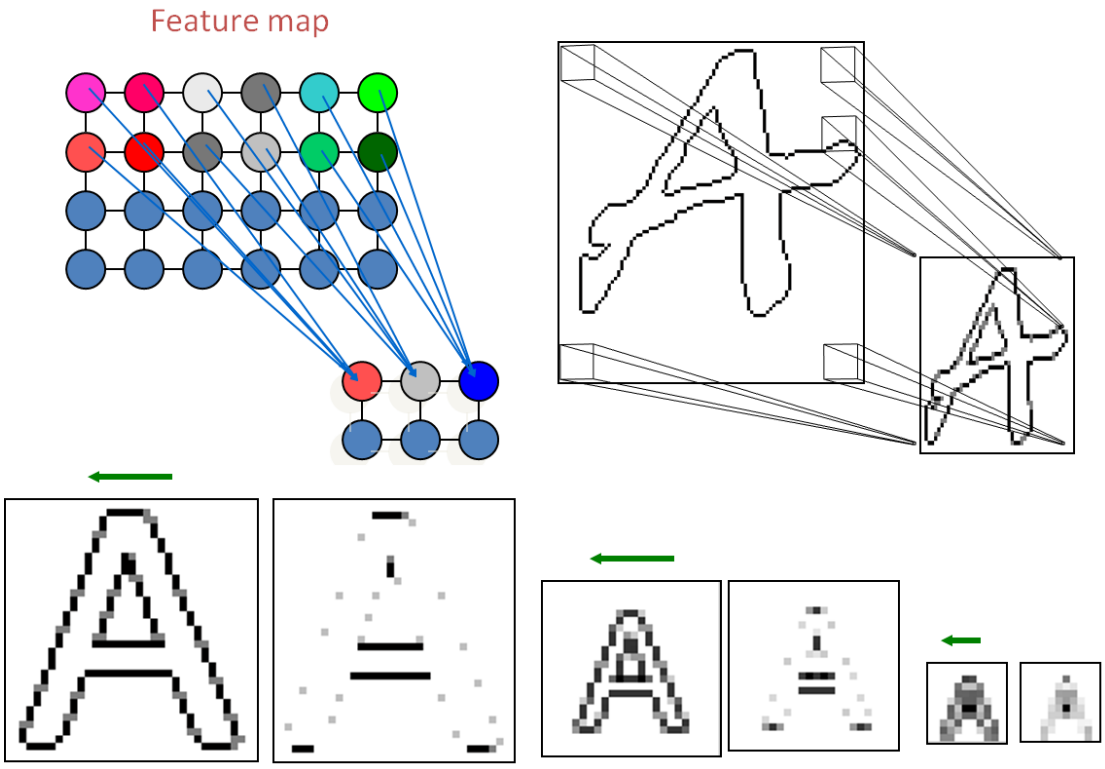

Figure 6.8: Sub Sampling to reduce the output map size by half to reduce the effect of small amount of noise, shift and distortion

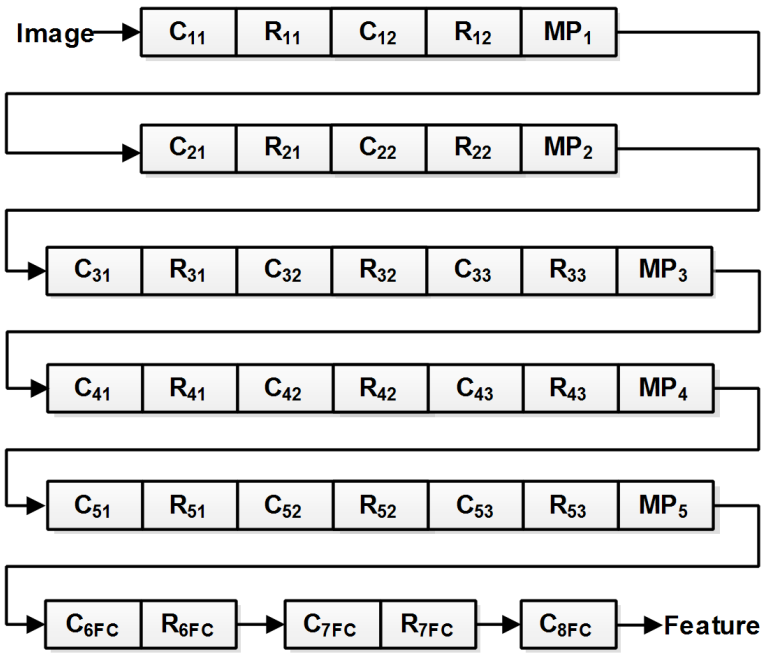

Figure 6.9: CNN architecture pre-trained with ImageNet, used to extract the features for leukaemia cell-line classification

the transfer learning capability of CNN for cell analysis. The ImageNet CNN deep model readily available (Chatfield et al. (2014); Vedaldi and Lenc (2014)) is heavily trained with several hundreds of non medical images (Deng et al. (2009)) for each of the 1000 classes (such as birds, cars, tools, etc.). It has 37 layers. First 36 layers used to generate the cell features for our experiment are shown in Fig. 6.9. The last 3 blocks represent the convolution and ReLU operation in fully connected (represented using the subscript FC) layers (Vedaldi and Lenc (2014)). Note that, the operations in these 36 layers are grouped into 8 blocks where in each block the layer is represented by O_{ij}. The O stands for the operation (Convolution (C), ReLU (R), Max Pooling (MP)), i stands for the block id, and the j stands for the j^{th} instance of the operation O in the block. For each input image, the output of the 36^{th} layer which is a 4096 element vector is taken as the feature. As noted earlier, it is reasonable to believe that these features are generated by an exhaustive set of feature detectors each looking for valid features like edges, blobs, corners etc. very locally in the image. These features are then used to train a classifier that does not require a large training set to determine the decision boundary. We have experimented with Support Vector Machine (SVM), Feed Forward Neural Network (FFN), Naive Bayes (NBS) and K-Nearest Neighbour (KNN) and has shown that the features generated have good discriminative power for the classification of leukaemia cell-lines.

6.4.1 Pre-trained CNN for classification of leukaemia cells

We use the CNN (Fig. 6.9) deeply trained on ImageNet (Deng et al. (2009); Chatfield et al. (2014)) to classify the leukaemia cell-lines. The roughly localised cell image is given as input to the network which will give a 4096 element feature vector at the output. The dimensionality of these vectors is reduced from 4096 to 20 using PCA (Jolliffe (2002)). The feature vectors of the reduced dimensions are used to train the classifiers. The use of CNN which was deeply trained on a specific data set and its use as a general feature extractor in a completely different setting is an example of transfer learning (Bar et al. (2015); Zhang et al. (2015); Zeng et al. (2015)) capability. This can be attributed to be following. The CNN model used was trained on a few million challenging images of 1000 broad categories. In

CNN, the kernel maps are learned by observing local image patches and hence the learned detectors (kernels) look for very local features such as curves, edges, corners and blobs. As most of these features are intrinsic for any image dataset, we can transfer this knowledge for other classification settings. This capability helps us to use deep CNN in classification task even if there is only limited labeled data for training. In such cases CNN is used as a deep feature extractor to find discriminant features. These features are then used with a classifier that does not require large training set in deciding the classification boundary.

6.5 Results and Discussion

In this section we discuss the classification of unlabeled unstained leukaemia cell-lines K562, MOLT and HL60 imaged using the low-cost, high-throughput microscopic imaging paradigm: the microfluidics based imaging flow cytometer. Altogether 618 cells (124 K562, 106 MOLT, and 388 HL60) were used in the experiment. The classification of these cell lines by deep learning networks (DBN and CNN) is compared for training time, testing time, and accuracy. The ability of the system for semi supervised learning is also considered. The classification accuracy achieved is compared with the the SVM based system in chapter 5 where the features reflecting size, shape, texture and complexity of the finely segmented cells are used for classification.

6.5.1 DBN classifier on roughly segmented cell images

In our experiment, we have normalized pixel intensities and treated them as the probability with which the visible nodes are turned ON in RBM. In order to assess the quality of training achieved, cross validation experiments are conducted at different folds ($K = 2, 3, 4, 5, 10$). In one cross validation experiment, training data from each class is randomly divided into K parts, each containing almost the same number of samples. Now, a cross validation test set is constructed by selecting one part from each class. The system is trained by all other samples and tested with this test set. The cross validation testing is repeated K times so that all samples are used for testing in some step. The entire experiment is repeated

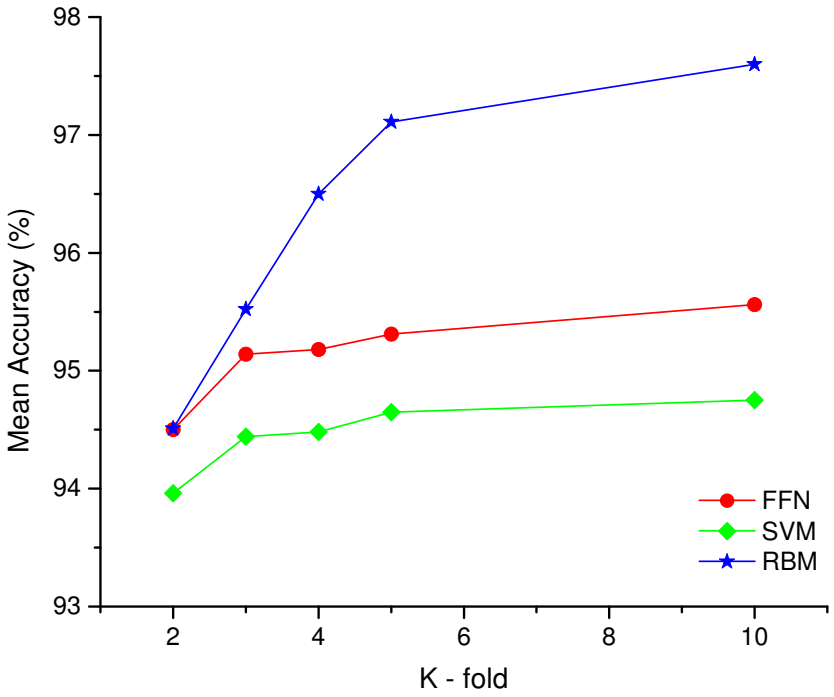

Figure 6.10: Effectiveness of RBM classification on raw cell images

100 times, each time selecting different combinations of samples for training as well as testing. The mean classification accuracy (mean) and standard deviation (std) are computed. The same set of experiments are conducted using SVM with linear kernel and feed forward neural network with back propagation (FFN). Being a three class classification problem, the majority voting based classification strategy is adopted to decide the class label in SVM. The results are shown in Table 6.1. It can be seen that the highest accuracy (with minimum standard deviation) in classifying cell images is achieved by RBM and is shown in Fig. 6.10. For FFN, we have used only one hidden layer. When the number of hidden neurons is varied ($H_n = 5, 10, 25, 50, 100, 250$), the average cross validation accuracy did not improve significantly for $H_n > 10$. So, number of hidden neurons in FFN is fixed as 10 in our implementation.

6.5.2 DBN classifier : ability to learn from unlabeled data

In order to check the effectiveness of RBM based classifier in learning structure of the data rather than learning the labels, the following testing strategy has been adopted. Entire data is divided into 3 sets. (**Case 1**) holding 50%, 30%, and 20% samples respectively and (**Case 2**) holding 67%, 13% and 20% samples

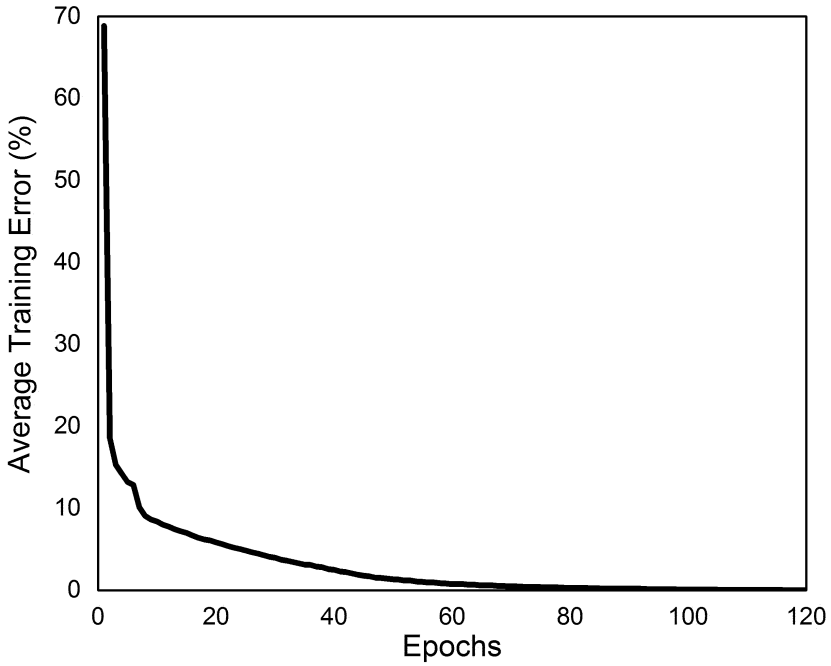

Figure 6.11: Training error of DBN during discriminative fine-tuning

respectively. The first set along with class labels and the second set without class labels are used for training and the system is tested with samples from second and third set. Note that, we have used samples from second set for testing since we did not use their labels during the training process. For both the cases (**Case 1 and Case 2**), the experiment is repeated for 100 times (each time re-initialising the weights) and average classification accuracy (mean) as well as standard deviation (std) is computed. The results are shown in Table 6.2. Note that RBM makes use of samples from set-1 and set-2 for training but uses labeled data (set-1) only during back propagation to fine tune the system. The training error for every pass through the entire training data (an epoch) is computed and the average error for each epoch over 100 iterations is plotted in Fig. 6.11 for **Case 1**. The training error profile is similar to that of the normal feed forward network, and in our case the training converges close to 80 epochs. The result of classification is shown in Table 6.2. The accuracies are slightly better than the results shown in Table 6.1 (2 and 3 fold (i.e., 50% and 67% training data)). Note that the unlabeled data cannot be used for learning FFN and SVM.

Table 6.1: Cross Validation Accuracy in % (mean (std)) - FFN, SVM and RBM on Cell Images

Kfold	**10**	**5**	**4**	**3**	**2**
FFN	95.56 (2.57)	95.31 (1.92)	95.18 (1.73)	95.14 (1.49)	94.50 (1.38)
SVM	94.75 (2.71)	94.65 (1.73)	94.48 (1.64)	94.44 (1.47)	93.96 (1.16)
RBM	**97.60 (0.48)**	**97.11 (0.62)**	**96.50 (0.60)**	**95.52 (0.20)**	**94.51 (0.17)**

Table 6.2: Classification Accuracy — Learn Structure from Data

Cases	Training Accuracy		Testing Accuracy	
	mean	std	mean	std
1	99.98	0.08	95.22	0.71
2	99.99	0.05	95.93	0.86

6.5.3 Classification on features from pre-trained CNN

The architecture of the CNN used to generate discriminative features from the cell images is shown in Fig. 6.9. As noted earlier in section 6.4, output of 36^{th} layer which is a 4096 element feature vector is used for CNN based classification. Before making use of these features, their dimension is reduced to 20 using PCA (Jolliffe (2002)). We have selected the number of principal components starting from 13 (this being the number of features used in chapter 5) and set at 20 since there was no considerable increase in the mean accuracy beyond 20. Once the feature descriptors are generated, the classifiers SVM, FFN, Naive Bayes (NBS), and K nearest neighbour (KNN, $K = 5$) are applied on top of it for 3 class classification. The cross validation experiments are conducted as explained earlier and the results are shown in Table 6.3. These results show that the accuracy is better and consistent across different cross validation experiments by different classifiers. Note that the CNN ImageNet model that we had used to extract features was never trained on the cell images, still producing high classification accuracy. This is a supporting result for the CNN transfer learning capability that we had discussed in section 6.4.

Table 6.3: Cross Validation Accuracy in % (mean (std)) on CNN Features

Kfold	**10**	**5**	**4**	**3**	**2**
SVM	97.80 (1.82)	97.69 (1.38)	97.62 (1.20)	97.47 (1.02)	97.19 (0.87)
FFN	98.26 (1.57)	98.18 (1.16)	98.16 (1.02)	98.05 (0.93)	97.96 (0.72)
NBS	98.40 (1.51)	98.37 (1.08)	98.36 (0.90)	98.40 (0.78)	98.38 (0.53)
KNN	98.42 (1.49)	98.40 (1.04)	98.37 (0.94)	98.35 (0.81)	98.36 (0.59)

Table 6.4: Cross Validation Accuracy in % (mean (std)) on Morphological Features (Chapter 5)

Kfold	**10**	**5**	**4**	**3**	**2**
SVM	95.22 (2.83)	94.47 (1.85)	94.63 (1.52)	94.20 (1.40)	93.57 (1.16)
FFN	92.92 (3.48)	92.74 (2.63)	92.57 (2.35)	92.47 (2.23)	91.76 (3.00)
NBS	90.89 (3.60)	90.77 (2.34)	90.70 (2.18)	90.67 (1.74)	90.35 (1.44)
KNN	80.40 (4.81)	79.93 (3.13)	79.48 (2.90)	78.94 (2.63)	77.75 (0.59)

6.5.4 Comparison with the classification on morphometric features discussed in chapter 5

Table 6.4 shows the result of cross validation experiments performed using cellular features, discussed in chapter 5, by different classifiers. The roughly segmented cells used as input to the RBM are processed further. The cells are localised by finding the cell contour, extracted features and are classified using SVM. The re-

Table 6.5: Run Time Analysis

Method	Rough Loclzn (18ms)	Fine Loclzn chapter 5 (42 ms)	Simple Features chapter 5 (14 ms)	CNN Features Chatfield(2014) (324 ms)	Classifier	Total Time (ms)
Segment & classify (chapter 5)	✓	✓	✓		SVM (1 ms)	75 (ms)
CNN ImageNet (Chatfield(2014))	✓			✓	SVM (1 ms)	342 (ms)
DBN Classifier	✓				RBM (0.05 ms)	18.05 (ms)

sults in Tables 6.1, 6.3 and 6.4 show that the deep learning methods work superior to this method.

Table 6.5 shows the time taken for different processing steps involved in the classification method provided in chapter 5, the ImageNet CNN model (Chatfield et al. (2014)) and the DBN classifier model. It can be seen that the DBN model not only gives good classification accuracy but also a faster response when compared to the other methods.

6.5.5 Performance with small percentage of training data

In order to check the effectiveness of deep learning networks in classification, experiments are carried out, even with less amount of training data and the results are shown in Table 6.6. The entire data set is divided into (S1, S2, S3) such that S1 and S2 together hold the training data (50% of the dataset) while S3 holds the testing data (remaining 50% of the dataset). We assume that the class labels of data contained in S2 are unavailable, and cannot be used for supervised training. Still the data from S2 can be used to train the individual RBMs for initialising the weights. The results of classification i.e., the classification by DBN, SVM (linear) on CNN-ImageNet features, and SVM on morphological features are shown for different training cases; 5%, 10%, 20%, 30%, and 40% labeled training data in S1. The results shown are the mean and standard deviation for 100 runs each time selecting random samples from the set for training. The mean accuracy achieved for the said 100 runs, when used 5, 10, 20, 30, 40, 50, and 75% of the available data for training, is provided in the graph in Fig. 6.12. It can be seen that the classification by deep learning method is effective even for small number of training samples in the case of RBM. Also note that, though the CNN-ImageNet model was heavily pre-trained on non-medical images, the generated features are very good in a classification point of view. These are reflected by the high average accuracy as well as low standard deviation for RBM and CNN in Table 6.6.

Table 6.6: Classification Accuracy: Training with less than 50% Samples

Labelled Data (%)	Morph. Features (chapter 5)	RBM	CNN Features
5	78.29 (3.72)	90.88 (1.18)	91.43 (3.33)
10	85.00 (2.74)	91.40 (1.06)	95.00 (1.40)
20	90.23 (1.49)	93.15 (0.85)	96.36 (0.97)
30	92.06 (1.26)	93.55 (0.61)	96.77 (0.76)
40	93.09 (1.13)	94.01 (0.49)	96.93 (0.90)

6.5.6 Comparison of class-specific accuracy

In order to show that the classification is not too much biased to any class, we report, in Table 6.7, the confusion matrix for each of the system developed. The results shown here are for the 10 fold (K = 10) cross validation experiment. As explained in section 6.5.1, 90% of samples are used for training and remaining 10% of data is used for testing. Such experiment is repeated 10 times such that every samples are used for testing exactly once. Note that Table 6.7 (A) reports the result from SVM running on morphological features (discussed in chapter 5), Table 6.7 (B) reports the results from the classification by RBM and Table 6.7 (C) reports the result from SVM running on CNN features. For example, the first row of 6.7 (A) shows that out of the total 388 (373 + 8 + 7) HL60 cells, 373 are correctly classified as HL60, 8 are wrongly classified as K562, and 7 are wrongly classified as MOLT. We have also presented in Table 6.8, the associated precision and Recall. These measures are based on the number of True Positives (TP), number of False Positives (FP), and number of False Negatives (FN). The precision is defined as $TP/(TP+FP)$ and recall is defined as $TP/(TP+FN)$. As these measures are primarily meant for binary classification, we have considered one-versus-all strategy. Thus we have three cases. In Case - 1, HL60 constitute positive class and K562&MOLT together forms negative class. Similarly, Case - 2 is K562 Vs HL60&MOLT, and Case - 3 is MOLT Vs HL60&K562. These measures tell us how much the system has learned to pick a particular class. The result in Table 6.8 shows that there is significant improvement in both measures with deep networks when compared to the method discussed in chapter 5.

Table 6.7: Comparison of Class Specific Accuracy

A. Morphologic Features (chapter 5)				B. Classification by RBM			C. CNN Features		
	HL60	**K562**	**MOLT**	**HL60**	**K562**	**MOLT**	**HL60**	**K562**	**MOLT**
HL60	373	8	7	382	2	4	385	1	2
K562	4	116	4	1	120	3	1	119	4
MOLT	2	5	99	3	2	101	1	5	100

Table 6.8: Comparison of Precision and Recall in % (One Against All)

	Case - 1		Case - 2		Case - 3	
	Precision	Recall	Precision	Recall	Precision	Recall
Method (chapter 5)	98.42	96.13	89.92	93.55	90.00	93.40
RBM	98.96	98.45	96.77	96.77	93.52	95.28
CNN Features	99.48	99.23	95.20	95.97	94.34	94.34

From the results shown so far, it is understood that RBM and CNN bypass the step of extracting hand-engineered features and performs better job than using a few important features (discussed in chapter 5). RBM based classifier not only classify the data but also extract the structure of the data while CNN extract discriminative features of the data. Further, the proposed deep learning based methods are quite general that it can be used for cell classification studies in microfluidic microscopic setup without the need for precise segmentation as well as explicit feature extraction, still producing an improved level of accuracy. The system could better capture the inter class variability of the cancerous cell-lines compared to the morphometric and textural features explored in chapter 5 and has significantly improved the precision as well as recall of the system when compared to the PCA signature based methods (Jagannadh et al. (2016)) and the SVM trained on morphometric features.

The leukaemia dataset used in this research is prepared by a custom-built, cost-effective microfluidic microscopy system and the accuracy reported on this dataset in this chapter set the benchmark for future studies. The central idea of the proposed method is to employ inexpensive optofluidic instrumentation for automated image acquisition and subsequent deep learning based cell recognition. While the opto-fluidic architecture enables cost-effective automation of image acquisition, the proposed deep learning based system enables implementation of effective de-

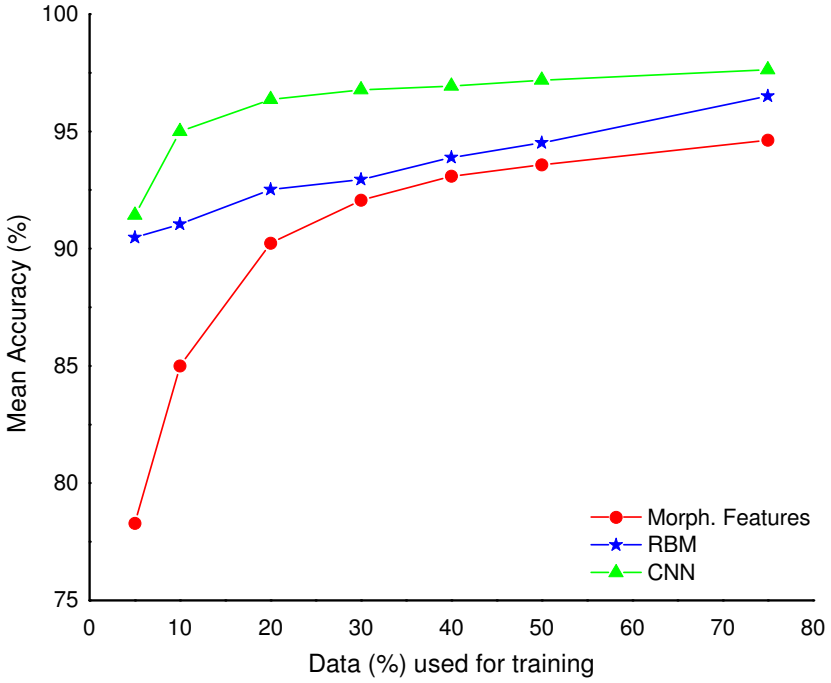

Figure 6.12: Classification accuracy of RBM, and SVM on features generated by CNN–ImgeNet as well as on the morphometric features (chapter 5).

cision making even in resource limited settings, where personnel trained in the art of diagnostic decision making are scarce/unavailable. As demonstrated, the image extraction and identification perform with fair level of accuracy thereby moving a promising step towards implementation of good quality health care even in resource-limited settings.

6.6 Publications

1. **G. Gopakumar**, K. Haribabu, Deepak Mishra, S.S. Gorthi, G.R.K.S. Subrahmanyam. "Cytopathological image analysis using deep learning networks in microfluidic microscopy", *J. Opt. Soc. Am. A*, **34(1)**:111–121, 2017.

6.7 Summary

In this chapter, we have proposed an approach for automatic cytopathologic analysis using deep learning methods. The DBN is found effective compared to the

classifier system proposed in chapter 5. The RBM model not only improved the classification accuracy but also avoided the more demanding accurate segmentation of cells. We have also noted the capability of RBM based system for learning structure of the data rather than learning labels which will be very helpful in medical image domain where often large dataset is available for training but only a small fraction labeled. We have also studied the applicability of CNN for cell analysis and found that a readily available CNN extensively trained on non medical image database ImageNet produces good discriminative features for classifying the leukaemia cell lines K562, MOLT, and HL60. In our investigation, we have found that deep learning methods outperformed the conventional systems in the classification of these cell lines. To the best of our knowledge, such a reporting on cytopathology images is first of its kind and we believe that it holds great promise in terms of enabling cost-effective cancer screening in resource-limited settings.

CHAPTER 7

Thesis Summary and Future Scope

In this research report, we have proposed suitable image processing algorithms for cytopthological analysis with recently developed/custom built portable, low-cost high-throughput microscopy, and microfluidics based imaging flow cytometry. We have presented a framework for focus stack patch based cell analysis using a custom designed convolutional neural network for malarial parasite detection due to protozoan of type falciparum and have shown that the detection accuracy outperformed the methods without using the focus stack. As part of the system, we have proposed a cascaded segmentation strategy to facilitate quantitative analysis moving in a direction to facilitate automated malaria diagnostic platforms for resource-poor settings.

We have also presented a completely automated cell analysis platform for a very cost-effective high-throughput microfluidic based imaging flow cytometry. The cell analysis platform includes feasible pre-processing methods, non-iterative, graph-based accurate cell segmentation strategy, hand-engineered feature extraction and a classification framework. We have explored cell signature (advanced PCA signature) based classification, hand-engineered morphologic feature based classification as well as deep learning based classification for leukaemia cell-lines; K562, MOLT and HL60. In our investigation, we have found that deep learning methods outperformed the conventional systems in the classification of these cell lines. We hope that imaging flow cytometers equipped with the proposed frameworks for image processing would enable cost-effective, automated and reliable disease screening in over-loaded facilities, which cannot afford to hire skilled personnel in large numbers. Such platforms would potentially facilitate screening camps in low income group countries; thereby transforming the current health care paradigms by enabling rapid, automated diagnosis for diseases like cancer.

However, all the works presented in this research report have used cultured cells. This includes both the malaria samples used in focus stack based classification and the leukaemia cell-lines used in the microfluidic microscopy setup.

The cultured cell-line helped us to generate the correct ground truth enabling quantitative analysis. A natural extension to this work could be testing the proposed methods on samples from real patients. Also, the proposed cell classification method in mIFC could naturally be extended for differential WBC counting. The method can be extended for a 3 part classification of WBCs. However, the 5 part differential classification requires the usage of an appropriate stain. Thus, for differential cell counting, one interesting work could be to bring up with a suitable stain at right concentration so that it can be used in flow.

We have addressed the leukemia cell-line classification mainly using the morphological features extracted from cell images. However, there are methods that address the problem based on the biomechanical properties of cells (Otto et al. (2015a)). These properties such as cell's deformability, cytoadherance, and time-dependent response to an applied stress change on the onset and progression of cancer (Suresh (2007)). There are quite a few design modifications in microfluidics channels (Yavuz (2009)) to study these parameters. One straight forward approach could be to include tiny constriction in the channel allowing the cells to squeeze through. The collected images can be used to extract required deformability parameters. For example, Kim et al. (2015) uses a node-pore sensor and a contraction channel to measure a deformability index of the cells thereby enabling differentiation of particular cancer-cell types within a heterogeneous cell population. There is also the concept of pinched flow microfluidics (J.S. Dudani (2013)) which uses flow channels to pinch and release the cells. Cells having different elasticity and viscosity might have separate lateral dynamic equilibrium position when they pass through the pinch. This fact is used to selectively mark cancer cells against healthy cells (Hur et al. (2011)). More recently Gangadhar et al. (2015) has studied extraction of deformability index of cells even with a straight channel microfluidic device and has validated the method by studying the change in deformability index of healthy, diabetic and sphered red blood cells. Thus, the study on the changes of biomechanical properties of leukemia cells for cell differentiation could be an important extension to the research problem that we were addressing. Also note that, we have addressed leukaemia cell-line classification problem from cultured cell-lines. For its applicability to end goal, leukaemia has to be identified from naive blood samples containing WBCs, RBCs, platelets, etc.

The methods discussed need to be extended/adapted for other types of diseases such as cervical cancer (Du et al. (2006)). This includes the modification that has to be brought on both microfluidics channel construction and on the processing framework. First, the modifications required to be brought in the design of microfluidics channels. Herein, the channel widths and depths have to be modified appropriately to incorporate the largest possible cervical cells (squamous cell can be as big as 1618 μm). Also, it would be essential for incorporating other strategies like micro-pillar arrays(Kim et al. (2017); Kaminaga et al. (2015)) or obstacle based methods (Huang et al. (2004)) to ensure that the cells are isolated without forming clumps in microfluidic channels. Further, in accordance with the modifications made to the channels, the optics have to be modified. Like the magnification might have to be increased or decreased and the depth of field of imaging has to be extended if the channel depth is increased. Coming to the question of image processing, in order to identify duplicate cells, we can make use of image processing algorithms like digital image correlation (McCormick and Lord (2010)). Further downstream processing might remain fairly the same, except for the parameters like circularity, which will have minimal relevance in case of cervical cells as they are largely irregular. Other measures of internal complexity like texture, variance etc., would still remain relevant and the developed framework would be directly applicable with minimal modifications. Also, the mIFC experiments that we had conducted could be tried for a larger dataset unlike the relatively smaller dataset (618 cells) used in this study.

Note that, the parameters used in the algorithm for processing cells used in this research work are explicitly defined. Most of these parameters as explained in relevant sections are defined based on the size of cell in image plane. When we are using the same processing framework for the analysis of different type of cell images, these hard-wired parameters need to be changed taking the cell image size into consideration. The size of cell in image plane is decided by the parameters of optics & hardware components used (Wayne (2014)). These include resolution of microscope objectives, magnification employed, and pixel size of the sensor. Given that the hardware components are standardized in the final design iteration and their arrangement remains fairly constant, the algorithms may not need any modifications for the type of cells analysed in this research. However, the

changes in the implementation may lead to changes in the illumination intensity, which affects the hard-wired parameters such as intensity thresholds used in focus stack slide scanner. In such a case, one can include an intensity calibration step to decide these parameters. This calibration can be performed, when no microfluidic device is placed, using the Gray level intensity recorded at the sensor. Coming to the case of pixel to real distance (magnification), the channel width is a known dimension, which can be used to assess this parameter from the video itself and set all the relevant parameters which are defined based on the cell size in image plane. The prototypes developed as part of this research work are still in the final stages of commercialization. At this point in time, the beta prototype is soon to go into clinical evaluation and would be brought to market soon after that. The bill of materials of both the microfluidics system and the focus stack collecting microscope as discussed in Chapter 1 is around 1500 US $. In general, the cost of the final prototype tends to be 4 - 5 times the bill of materials (Chaturvedi (2016)). So, even with that into consideration, this would still be a cheaper alternative, when compared to the current industry standards like Amnis 199,000 US $ (Amn (2016)).

REFERENCES

1. (2016). Amnis Corporation ® ISX - MKII Brochure. `https://www.amnis.com/documents/brochures/ISX-MKII20Brochure_Final_Web.pdf`. Accessed: 2016-07-28.

2. (2016). LBP/VAR implementation; centre for machine vision and signal analysis; university of oulu. `http://www.cse.oulu.fi/CMV/Downloads/LBPMatlab`. Accessed: 2016-10-15.

3. Aalto, M. (2014). *Classification of Medical Data using Restricted Boltzmann Machines*. Master's thesis School of Information Sciences - University of Tampere.

4. Adan, A., Alizada, G., Kiraz, Y., Baran, Y., and Nalbant, A. (2017). Flow cytometry: basic principles and applications. *Critical Reviews in Biotechnology*, *37*, 163–176.

5. Anggraini, D., Nugroho, A. S., Pratama, C., Rozi, I. E., Iskandar, A. A., and Hartono, R. N. (2011). Automated status identification of microscopic images obtained from malaria thin blood smears. In *Electrical Engineering and Informatics (ICEEI), 2011 International Conference on* (pp. 1–6).

6. Bancroft, J. (2008). *Theory and Practice of Histological Techniques*. Elsevier Health Sciences, Amsterdam.

7. Bar, Y., Diamant, I., Wolf, L., Lieberman, S., Konen, E., and Greenspan, H. (2015). Chest pathology detection using deep learning with non-medical training. In *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)* (pp. 294–297).

8. Barteneva, N., Fasler-Kan, E., and Vorobjev, I. (2012). Imaging flow cytometry: coping with heterogeneity in biological systems. *Journal of Histochemistry and Cytochemistry*, *60*, 723–733.

9. Basiji, D. A., Ortyn, W. E., Liang, L., Venkatachalam, V., and Morrissey, P. (2007). Cellular image analysis and imaging by flow cytometry. *Clinics in Laboratory Medicine*, *27*, 653 – 670. Flow Cytometry.

10. Beers, E. J., Samsel, L., Mendelsohn, L., Saiyed, R., Fertrin, K. Y., Brantner, C. A., Daniels, M. P., Nichols, J., McCoy, J. P., and Kato, G. J. (2014). Imaging flow cytometry for automated detection of hypoxia-induced erythrocyte shape change in sickle cell disease. *American journal of hematology*, *89*, 598–603.

11. Bengio, Y. (2009). Learning deep architectures for AI. *Found. Trends Mach. Learn.*, *2*, 1–127.

12. de Bitencourt, E. D. S., Voegeli, C. F., dos Santos Onzi, G., Boscato, S. C., Ghem, C., and Munhoz, T. (2013). Validation of the sysmex sp-1000i automated slide preparer-stainer in a clinical laboratory. *Rev Bras Hematol Hemoter*, *35*, 404–408.

13. Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* COLT '92 (pp. 144–152). New York, NY, USA: ACM.

14. Breen, E. J., and Jones, R. (1996). Attribute openings, thinnings, and granulometries. *Comput. Vis. Image Underst.*, *64*, 377–389.

15. Brosch, T., and Tam, R. (2013). Manifold learning of brain MRIs by deep learning. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013* (pp. 633–640). Springer Berlin Heidelberg volume 8150 of *Lecture Notes in Computer Science*.

16. Brown, M., and Wittwer, C. (2000). Flow cytometry: Principles and clinical applications in hematology. *Clinical Chemistry*, *46*, 1221–1229.

17. Canny, J. (1986). A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (pp. 679–698).

18. Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*.

19. Chaturvedi, J. (2016). Common misconceptions among indian doctors. In J. Chaturvedi (Ed.), *Inventing Medical Devices: A Perspective from India* chapter 2. (pp. 10–22). Chennai: Notion Press.

20. Cireşan, D. C., Giusti, A., Gambardella, L. M., and Schmidhuber, J. (2013). Mitosis detection in breast cancer histology images with deep neural networks. In K. Mori, I. Sakuma, Y. Sato, C. Barillot, and N. Navab (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013: 16th International Conference, Nagoya, Japan, September 22-26, 2013, Proceedings, Part II* (pp. 411–418). Berlin, Heidelberg: Springer Berlin Heidelberg.

21. Clark, ., George, (U.S.), B. S. C., and Clark, G. (1981). *Staining procedures*. (4th ed.). Baltimore : Published for the Biological Stain Commission by Williams & Wilkins.

22. Cohen, S. J., Punt, C. J. A., Iannotti, N., Saidman, B. H., Sabbath, K. D., Gabrail, N. Y., Picus, J., Morse, M., Mitchell, E., Miller, M. C., Doyle, G. V., Tissing, H., Terstappen, L. W. M. M., and Meropol, N. J. (2008). Relationship of circulating tumor cells to tumor response, progression-free survival, and overall survival in patients with metastatic colorectal cancer. *Journal of Clinical Oncology*, *26*, 3213–3221.

23. Das, D., Ghosh, M., Pal, M., Maiti, A., and Chakraborty, C. (2013). Machine learning approach for automated screening of malaria parasite using light microscopic images. *Micron*, *45*, 97–106.

24. Deco, G., Jirsa, V. K., Robinson, P. A., Breakspear, M., and Friston, K. J. (2008). The dynamic brain: From spiking neurons to neural masses and cortical fields. *PLoS Computational Biology*, *4*.

25. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

26. Dey, P. (2010). Cancer nucleus: Morphology and beyond. *Diagnostic Cytopathology*, *38*, 382–390.

27. Di Caprio, G., Stokes, C., Higgins, J. M., and Schonbrun, E. (2015). Single-

cell measurement of red blood cell oxygen affinity. *Proceedings of the National Academy of Sciences*, *112*, 9984–9989.

28. Di Cataldo, S., Ficarra, E., Acquaviva, A., and Macii, E. (2008). Segmentation of nuclei in cancer tissue images: Contrasting active contours with morphology-based approach. In *BioInformatics and BioEngineering, 2008. BIBE 2008. 8th IEEE International Conference on* (pp. 1–6).

29. Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, *1*, 269–271.

30. Dimopoulos, S., Mayer, C. E., Rudolf, F., and Stelling, J. (2014). Accurate cell segmentation in microscopy images using membrane patterns. *Bioinformatics*, *30*, 2644–2651.

31. Du, Z., Colls, N., Cheng, K., Vaughn, M., and Gollahon, L. (2006). Microfluidic-based diagnostics for cervical cancer cells. *Biosensors and Bioelectronics*, *21*, 1991 – 1995.

32. Elter, M., HaBlmeyer, E., and ZerfaB, T. (2011). Detection of malaria parasites in thick blood films. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 5140–5144).

33. Fabbri, R., Costa, L. D. F., Torelli, J. C., and Bruno, O. M. (2008). 2D Euclidean distance transform algorithms: A comparative survey. *ACM Comput. Surv.*, *40*, 2:1–2:44.

34. Firestone, L., Cook, K., Culp, K., Talsania, N., and Junior, K. P. (1991). Comparison of autofocus methods for automated microscopy. *Cytometry*, *12*, 195–206.

35. Fischer, A., and Igel, C. (2014). Training restricted boltzmann machines. *Pattern Recogn.*, *47*, 25–39.

36. Flow, C. (2016). Thermofisher ^{TM} flow cytometry tutorials. `https://www.thermofisher.com/in/en/home/support/tutorials.html`. Accessed: 2017-1-2.

37. Gangadhar, E., Rajesh, S., and Gorthi, S. (2015). Deformability measurement of single-cells at high-throughput with imaging flow cytometry. *Journal of Lightwave Technology*, *33*, 3475–3480.

38. Geman, S., and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-6*, 721–741.

39. Goda, K., Ayazi, A., Gossett, D. R., Sadasivam, J., Lonappan, C. K., Sollier, E., Fard, A. M., Hur, S. C., Adam, J., Murray, C., Wang, C., Brackbill, N., Di Carlo, D., and Jalali, B. (2012). High-throughput single-microparticle imaging flow analyzer. *Proceedings of the National Academy of Sciences*, *109*, 11630–11635.

40. Goodfellow, I. J., Le, Q. V., Saxe, A. M., Lee, H., and Ng, A. Y. (2009). Measuring invariances in deep networks. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems* NIPS'09 (pp. 646–654). USA: Curran Associates Inc.

41. Gopakumar, G., Jagannadh, V. K., Gorthi, S. S., and Subrahmanyam, G. R. K. S. (2016). Framework for morphometric classification of cells in imaging flow cytometry. *Journal of Microscopy*, *261*, 307–319.

42. Gorthi, S. S., Schaak, D., and Schonbrun, E. (2013). Fluorescence imaging of flowing cells using a temporally coded excitation. *Optics Express*, *21*, 5164–5170.

43. Gorthi, S. S., and Schonbrun, E. (2012). Phase imaging flow cytometry using a focus-stack collecting microscope. *Optics Letters*, *37*, 707–709.

44. Greenspan, H., van Ginneken, B., and Summers, R. M. (2016). Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*, *35*, 1153–1159.

45. Hamm, P., Schulz, J., and Englmeier, K.-H. (2010). Content-based autofocusing in automated microscopy. *Image Analysis and Stereology*, *29*, 173–180.

46. Haralick, R., and Shapiro, L. G. (1992). *Computer annd Robot Vision* volume 1. (1st ed.). Addison-Wesley Publishing Company.

47. Haralick, R. M., Shanmugam, K., and Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-3*, 610–621.

48. Hartigan, J. A. (1975). *Clustering Algorithms (Probability & Mathematical Statistics)*. John Wiley & Sons Inc.

49. He, G., Li, J., Huang, X., and Zou, Y. (2010). An integrated auto-focusing system for biomedical digital microscope. In *2010 3rd International Conference on Biomedical Engineering and Informatics* (pp. 1420–1423). volume 4.

50. He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *ArXiv e-prints*, .

51. He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*.

52. Henel, G., and Schmitz, J. L. (2015). Basic theory and clinical applications of flow cytometry. *Laboratory Medicine*, *38*, 428.

53. Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Comput.*, *14*, 1771–1800.

54. Hinton, G. E. (2007). Boltzmann machine. *Scholarpedia*, *2*, 1668. Revision #91075.

55. Hinton, G. E. (2012). A practical guide to training restricted boltzmann machines. In *Neural Networks: Tricks of the Trade* (pp. 599–619). Springer.

56. Hinton, G. E., and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*, 504–507.

57. Huang, L. R., abd R. H. Austin, E. C. C., and j. C. Sturm (2004). Continuous particle separation through deterministic lateral displacement. *Science*, *304*, 987–990.

58. Hunter, R. S. (1958). Photoelectric color difference meter. *J. Opt. Soc. Am.*, *48*, 985–993.

59. Hur, S., Henderson, N., McCabe, E., and Carlo, D. (2011). Deformability-based cell classification and enrichment using inertial microfluidics. *Lab Chip*, *11*, 912–920.

60. Irshad, H., Veillard, A., Roux, L., and Racoceanu, D. (2014). Methods for nuclei detection, segmentation, and classification in digital histopathology: A review – 2014; current status and future potential. *Biomedical Engineering, IEEE Reviews in*, *7*, 97–114.

61. Jagannadh, V. K., Gopakumar, G., Subrahmanyam, G. R. K. S., and Gorthi, S. S. (2016). Microfluidic microscopy-assisted label-free approach for cancer screening: automated microfluidic cytology for cancer screening. *Medical & Biological Engineering & Computing*, (pp. 1–8).

62. Jolliffe, I. (2002). *Principal Component Analysis*. Springer Series in Statistics. Springer.

63. J.S. Dudani, H. T. D. C., D.R. Gossett (2013). Pinched-flow hydrodynamic stretching of single-cells. *Lab Chip*, *13*, 3728–3734.

64. Junior, A. K., Costa, M. G. F., Filho, C. F. F. C., Fujimoto, L. B. M., and Salem, J. (2010). Evaluation of autofocus functions of conventional sputum smear microscopy for tuberculosis. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology* (pp. 3041–3044).

65. Kachel, V., Benker, G., Lichtnau, K., Valet, G., and Glossner, E. (1979). Fast imaging in flow: a means of combining flow-cytometry and image analysis. *Journal of Histochemistry & Cytochemistry*, *27*, 335–341.

66. Kaminaga, M., Ishida, T., Kadonosono, T., Kizaka-Kondoh, S., and Omata, T. (2015). Uniform cell distribution achieved by using cell deformation in a micropillar array. *Micromachines*, *6*, 409–422.

67. Kareem, S., Kale, I., and Morling, R. C. S. (2012). Automated malaria parasite detection in thin blood films:- a hybrid illumination and color constancy insensitive, morphological approach. In *Circuits and Systems (APCCAS), 2012 IEEE Asia Pacific Conference on* (pp. 240–243).

68. Kay, D. B., Cambier, J. L., and Wheeless, L. L. (1979). Imaging in flow. *Journal of Histochemistry & Cytochemistry*, *27*, 329–334.

69. Kim, E., Corte-Real, M., and Baloch, Z. (2016). A deep semantic mobile application for thyroid cytopathology. (pp. 97890A–97890A–9). volume 9789.

70. Kim, J., Lei, A., and Sohn, L. (2015). Characterising mechanical properties of cancer cells by node-pore sensing. In *Proceedings 19th International Conference on Miniaturized Systems for Chemistry and Life Sciences* (pp. 499 – 501).

71. Kim, K., Koo, J., Moon, S., and Lee, W. G. (2017). Role of micropillar arrays in cell rolling dynamics. *Analyst*, *142*, 110–117.

72. Kirby, M., and Sirovich, L. (1990). Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*, 103–108.

73. Kubota, T. (2010). Fixation driven contour completion with angular ordering. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on* (pp. 57–63).

74. Kubota, T. (2012). Contour completion around a fixation point. *CoRR*, *abs/1208.3512*.

75. Laerum, O. D., and Farsund, T. (1981). Clinical application of flow cytometry: A review. *Cytometry*, *2*, 1–13.

76. Larochelle, H., Bengio, Y., Louradour, J., and Lamblin, P. (2009). Exploring strategies for training deep neural networks. *J. Mach. Learn. Res.*, *10*, 1–40.

77. Larochelle, H., Mandel, M., Pascanu, R., and Bengio, Y. (2012). Learning algorithms for the classification restricted boltzmann machine. *J. Mach. Learn. Res.*, *13*, 643–669.

78. LeCun, Y., and Bengio, Y. (1998). The handbook of brain theory and neural networks. chapter Convolutional Networks for Images, Speech, and Time Series. (pp. 255–258). Cambridge, MA, USA: MIT Press.

79. Lewis, M., Bain, B. J., Bates, I., and Laffan, M. A. (2012). *Dacie and Lewis Practical Haematology: Expert Consult: Online and Print*. (11th ed.). Churchill Livingstone, Elsevier.

80. Li, Q., Cai, W., Wang, X., Zhou, Y., Feng, D. D., and Chen, M. (2014). Medical image classification with convolutional neural network. In *Control Automation Robotics Vision (ICARCV), 2014 13th International Conference on* (pp. 844–848).

81. Liang, Z., Powell, A., Ersoy, I., Poostchi, M., Silamut, K., Palaniappan, K., Guo, P., Hossain, M. A., Sameer, A., Maude, R. J., Huang, J. X., Jaeger, S., and Thoma, G. (2016). Cnn-based image analysis for malaria diagnosis. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (pp. 493–496).

82. Linder, N., Turkki, R., Walliander, M., Mårtensson, A., Diwan, V., Rahtu, E., Pietikäinen, M., and Lundin, M. (2014). A malaria diagnostic tool based on computer vision screening and visualization of plasmodium falciparum candidate areas in digitized blood smears. *PLoS ONE*, *9*, e104855.

83. Lisa, M., Kathleen, E., John, C., and James, P. (2014). Improved quantitative analysis of primary bone marrow megakaryocytes utilising imaging flow cytometry. *Cytometry Part A: the journal of the International Society for Analytical Cytology*, *85*, 302–312.

84. Lo, S.-C. B., Chan, H.-P., Lin, J.-S., Li, H., Freedman, M. T., and Mun, S. K. (1995). Artificial convolution neural network for medical image pattern recognition. *Neural Networks*, *8*, 1201 – 1214. Automatic Target Recognition.

85. Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, *60*, 91–110.

86. Makkapati, V. V., and Rao, R. M. (2009). Segmentation of malaria parasites in peripheral blood smear images. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 1361–1364).

87. McCormick, N., and Lord, J. (2010). Digital image correlation. *Materials Today*, *13*, 52 – 54.

88. Mehrjou, A., Abbasian, T., and Izadi, M. (2013). Automatic malaria diagnosis system. In *Robotics and Mechatronics (ICRoM), 2013 First RSI/ISM International Conference on* (pp. 205–211).

89. Najman, L., and Ronse, C. (2005). *Mathematical Morphology: 40 Years On*. Springer.

90. Nayak, N., Chang, H., Borowsky, A., Spellman, P., and Parvin, B. (2013). Classification of tumor histopathology via sparse feature learning. In *Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on* (pp. 1348–1351).

91. Nayar, R. (2014). *Cytopathology in Oncology*. Springer.

92. Ng, E. X., Miller, M. A., Jing, T., and Chen, C.-H. (2016). Single cell multiplexed assay for proteolytic activity using droplet microfluidics. *Biosensors and Bioelectronics*, *81*, 408 – 414.

93. Ng, E. X., Miller, M. A., Jing, T., Lauffenburger, D. A., and Chen, C.-H. (2015). Low-volume multiplexed proteolytic activity assay and inhibitor analysis through a pico-injector array. *Lab Chip*, *15*, 1153–1159.

94. Nugroho, H. A., Akbar, S. A., and Murhandarwati, E. E. H. (2015). Feature extraction and classification for detection malaria parasites in thin blood smear. In *2015 2nd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)* (pp. 197–201).

95. Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*, 971–987.

96. Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, *9*, 62–66.

97. Otto, O., Rosendahl, P., Mietke, A., Golfier, S., Herold, C., Klaue, D., Girardo, S., Pagliara, S., Ekpenyong, A., Jacobi, A., Wobus, M., Topfner, N., Keyser, U., Mansfeld, J., Fischer-Friedrich, E., and Guck, J. (2015a). Real-time deformability cytometry: on-the-fly cell mechanical phenotyping. *Nature Methods*, *12*, 199–202.

98. Otto, O., Rosendahl, P., Mietke, A., Golfier, S., Herold, C., Klaue, D., Girardo, S., Pagliara, S., Ekpenyong, A., Jacobi, A., Wobus, M., Topfner, N., Keyser, U. F., Mansfeld, J., Fischer-Friedrich, E., and Guck, J. (2015b). Real-time deformability cytometry: on-the-fly cell mechanical phenotyping. *Nat Meth*, *12*, 199–202.

99. Oud, P. S., Haag, D. J., Zahniser, D. J., Ramaekers, F. C., Huysmans, A. C., Veldhuizen, J. A., Verheyen, R. H., Verrijp, K., Broers, J. L., and Herman, C. J.

(1986). Cytopress: automated slide preparation of cytologic material from suspension. *Cytometry*, *7*, 8–17.

100. Pantanowitz, L., Valenstein, P., Evans, A., Kaplan, K., Pfeifer, J., Wilbur, D., Collins, L., and Colgan, T. (2011). Review of the current state of whole slide imaging in pathology. *Journal of Pathology Informatics*, *2*, 36–45.

101. PathScope (2016). Pathscope ^{TM} slide scanner; digipath inc. : Pathology delivered digitally. `http://www.digipath.biz/pr/PathScope.pdf`. Accessed: 2016-12-7.

102. Pinkaew, A., Limpiti, T., and Trirat, A. (2015). Automated classification of malaria parasite species on thick blood film using support vector machine. In *2015 8th Biomedical Engineering International Conference (BMEiCON)* (pp. 1–5).

103. Pouliakis, A., Karakitsou, E., Margari, N., Bountris, P., Haritou, M., Panayiotides, J., Koutsouris, D., and Karakitsos, P. (2016). Artificial neural networks as decision support tools in cytopathology: Past, present, and future. *Biomedical Engineering and Computational Biology*, *7*, 1–18.

104. Preedanan, W., Phothisonothai, M., Senavongse, W., and Tantisatirapong, S. (2016). Automated detection of plasmodium falciparum from giemsa-stained thin blood films. In *2016 8th International Conference on Knowledge and Smart Technology (KST)* (pp. 215–218).

105. Purnama, I. K. E., Rahmanti, F. Z., and Purnomo, M. H. (2013). Malaria parasite identification on thick blood film using genetic programming. In *Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME), 2013 3rd International Conference on* (pp. 194–198).

106. Purwar, Y., , Shah, S. L., Clarke, G., Almugairi, A., and Muehlenbachs, A. (2011). Automated and unsupervised detection of malarial parasites in microscopic images. *Malaria Journal*, *10*, 364.

107. Ravendran, A., de Silva, K. W. T. R. T., and Senanayake, R. (2015). Moment invariant features for automatic identification of critical malaria parasites. In *2015 IEEE 10th International Conference on Industrial and Information Systems (ICIIS)* (pp. 474–479).

108. Rohan, K. (2016). The vanishing gradient problem. https://ayearofai.com/rohan-4-the-vanishing-gradient-problem-ec68f76ffb9b. Accessed: 2017-04-10.

109. Rojo, M., Garcia, G., Mateos, C., Garcia, J., and Vicente, M. (2006). Critical comparison of 31 commercially available digital slide systems in pathology. *Int. J. Surg. Pathol*, *14*, 285–305.

110. Sahiner, B., Chan, H.-P., Petrick, N., Wei, D., Helvie, M. A., Adler, D. D., and Goodsitt, M. M. (1996). Classification of mass and normal breast tissue: a convolution neural network classifier with spatial domain and texture images. *IEEE Transactions on Medical Imaging*, *15*, 598–610.

111. Samir, P., Chitra, P., and Urhekar, A. (2013). A comparative evaluation of conventional staining methods and immunological techniques for the diagnosis of malaria. *Journal of Evolution of Medical and Dental Sciences*, *2*, 712–724.

112. Sathpathi, S., Mohanty, A. K., Satpathi, P., Mishra, S. K., Behera, P. K., Patel, G., and Dondorp, A. M. (2014). Comparing Leishman and Giemsa staining for the assessment of peripheral blood smear preparations in a malaria-endemic region in india. *Malaria Journal*, *13*, 1–5.

113. Savkare, S. S., and Narote, S. P. (2015). Automated system for malaria parasite identification. In *Communication, Information Computing Technology (ICCICT), 2015 International Conference on* (pp. 1–4).

114. Schmah, T., Hinton, G. E., Small, S. L., Strother, S., and Zemel, R. S. (2009). Generative versus discriminative training of rbms for classification of fmri images. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou (Eds.), *Advances in Neural Information Processing Systems 21* (pp. 1409–1416). Curran Associates, Inc.

115. Schonbrun, E., Di Caprio, G., and Schaak, D. (2013). Dye exclusion microfluidic microscopy. *Optics Express*, *21*, 8793–8798.

116. Schonbrun, E., Gorthi, S. S., and Schaak, D. (2012). Microfabricated multiple field of view imaging flow cytometry. *Lab Chip*, *12*, 268–273.

117. Schonbrun, E., Malka, R., Di Caprio, G., Schaak, D., and Higgins, J. M. (2014). Quantitative absorption cytometry for measuring red blood cell hemoglobin mass and volume. *Cytometry Part A*, *85*, 332–338.

118. SHI, Z., and He, L. (2011). Current status and future potential of neural networks used for medical image processing. *Journal of Multimedia*, *6*.

119. Simonyan, K., and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*.

120. Sobel, I. (2014). *History and definition of the Sobel operator*. Technical Report. First presented at Stanford A.I. Project 1968.

121. Soille, P. (2003). *Morphological image analysis: principles and applications*. (2nd ed.). Springer.

122. Solomon, D. (2003). Chapter 14: Role of triage testing in cervical cancer screening. *JNCI Monographs*, *2003*, 97–101.

123. Stewart, B. W., and Wild, C. (2014). *World Cancer Report 2014*. World Health Organization.

124. Sung, Y., Lue, N., Hamza, B., Martel, J., Irimia, D., Dasari, R. R., Choi, W., Yaqoob, Z., and So, P. (2014). Three-dimensional holographic refractive-index measurement of continuously flowing cells in a microfluidic channel. *Physical Review Applied*, *1*, 014002.

125. Suresh, S. (2007). Biomechanics and biophysics of cancer cells. *Acta Biomaterialia*, *3*, 413–438.

126. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*.

127. Tabe, Y., Yamamoto, T., Maenou, I., Nakai, R., Idei, M., Horii, T., Miida, T., and Ohsaka, A. (2014). Performance evaluation of the digital cell imaging analyzer di-60 integrated into the fully automated sysmex xn hematology analyzer system. *Clinical Chemistry and Laboratory Medicine*, *53*, 281–289.

128. Tek, F. B., Dempster, A. G., and Kale, I. (2009). Computer vision for microscopy diagnosis of malaria. *Malaria Journal*, *8*, 153.

129. Terstappen, L., Safford, M., K´onemann, S., Loken, M., Zurlutter, K., B´uchner, T., Hiddemann, W., and B., W. (1992). Flow cytometric characterization of acute myeloid leukemia. part ii. phenotypic heterogeneity at diagnosis. *Leukemia*, *6*, 70–80.

130. Terstappen, L. W., Johnsen, S., Segers-Nolten, I. M., and Loken, M. R. (1990). Identification and characterization of plasma cells in normal human bone marrow by high-resolution flow cytometry. *Blood*, *76*, 1739–1747.

131. Thiran, J. P., and Macq, B. (1996). Morphological feature extraction for the classification of digital images of cancerous tissues. *IEEE Transactions on Biomedical Engineering*, *43*, 1011–1020.

132. Turk, M. A., and Pentland, A. P. (1991). Face recognition using eigenfaces. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 586–591).

133. Vedaldi, A., and Lenc, K. (2014). Matconvnet - convolutional neural networks for MATLAB. *CoRR*, *abs/1412.4564*.

134. Vincent, L., and Soille, P. (1991). Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*, 583–598.

135. Walts, A. E., and Thomas, P. (2002). Endometrial cells and the autopap system for primary screening of cervicovaginal pap smears. *Diagnostic Cytopathology*, *27*, 232–237.

136. Wayne, R. (2014). Chapter 4 - bright-field microscopy. In R. Wayne (Ed.), *Light and Video Microscopy (Second Edition)* (pp. 79 – 96). San Diego: Academic Press. (Second edition ed.).

137. W.H.O. (2010). *Basic malaria microscopy - Part I: Learner's guide*. World Health Organization.

138. Wu, J., Li, J., and Chan, R. K. Y. (2013). A light sheet based high throughput 3d-imaging flow cytometer for phytoplankton analysis. *Optics Express*, *21*, 14474–14480.

139. Xia, Y., and Whitesides, G. M. (1998). Soft lithography. *Annual Review of Materials Science*, *28*, 153–184.

140. Xu, Y., Mo, T., Feng, Q., Zhong, P., Lai, M., and Chang, E. I. C. (2014). Deep learning of feature representation with multiple instance learning for medical image analysis. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1626–1630).

141. Yan, Y., Boey, D., Ng, L. T., Gruber, J., Bettiol, A., Thakor, N. V., and Chen, C.-H. (2016). Continuous-flow c. elegans fluorescence expression analysis with real-time image processing through microfluidics. *Biosensors and Bioelectronics*, *77*, 428 – 434.

142. Yang, J., Zhang, D., Frangi, A. F., and yu Yang, J. (2004). Two-dimensional pca: a new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*, 131–137.

143. Yavuz, M. (2009). *The mechanical properties of cells measured using micro-fluidic devices – Application to atherosclerosis*. Master's thesis Department of Mechanical Engineering, Eindhoven University of Technology Netherlands.

144. Zahniser, D. J., and Hurley, A. A. (1996). Automated slide preparation system for the clinical laboratory. *Cytometry*, *26*, 60–64.

145. Zeng, T., Li, R., Mukkamala, R., Ye, J., and Ji, S. (2015). Deep convolutional neural networks for annotating gene expression patterns in the mouse brain. *BMC Bioinformatics*, *16*, 1–10.

146. Zhang, W., Li, R., Zeng, T., Sun, Q., Kumar, S., Ye, J., and Ji, S. (2015). Deep model based transfer and multi-task learning for biological image analysis. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* KDD '15 (pp. 1475–1484). New York, NY, USA: ACM.

147. Zhu, H., Isikman, S. O., Mudanyali, O., Greenbaum, A., and Ozcan, A. (2013). Optical imaging techniques for point-of-care diagnostics. *Lab Chip*, *13*, 51–67.

148. Zimmerman, P., and Howes, R. (2015). Malaria diagnosis for malaria elimination. *Curr Opin Infect Dis*, *28*, 446–454.

149. Zou, L., Chen, J., Zhang, J., and Garcia, N. (2010). Malaria cell counting diagnosis within large field of view. In *Digital Image Computing: Techniques and Applications (DICTA), 2010 International Conference on* (pp. 172–177).

APPENDIX A

BackPropagation in Deep Neural Networks

The backpropagation, is a widely used method in training artificial feed forward neural networks (FFN). Training ANN has two steps : error backward propagation and weight update. When an input vector is presented to the network, it is propagated forward through the network, layer by layer, until it reaches the output layer. The output of the network is then compared to the desired output, using a loss function, and an error value is calculated for each of the neurons in the output layer. The error values are then propagated backwards (hence the name backpropagation), starting from the output, until each neuron has an associated error value which roughly represents its contribution to the original output. Once the error at each neuron is determined, the weights are updated using an optimization method like gradient descent so as to minimize the loss function. we show that the error back propagation can be conveniently done in matrix notion in almost similar manner from layer to layer. Further, it will be shown that the weight update term in gradient descent can be easily computed based on the back propagated error at the rear end of the connection (carrying the weight) and the input of it. This process thus gives a simpler interpretation and derivation for weight update in deep networks. In this section, we discuss a generalized formulation of deep neural network training using back propagation.

A.1 Notations and Meaning

T_z	Target or desired output at node z in the final layer
O_z	Output produced at some arbitrary node z in some layer
ΔE_L	Error computed or backpropagated at layer L
ΔEF_L	Error weighted by the derivative of activation function at layer L
US_2	The operation up-sampling by 2
'*'	The convolution operator

$\odot$	The correlation operator
'.*'	Element by element multiplication

A.2 Training FFN using backpropagation

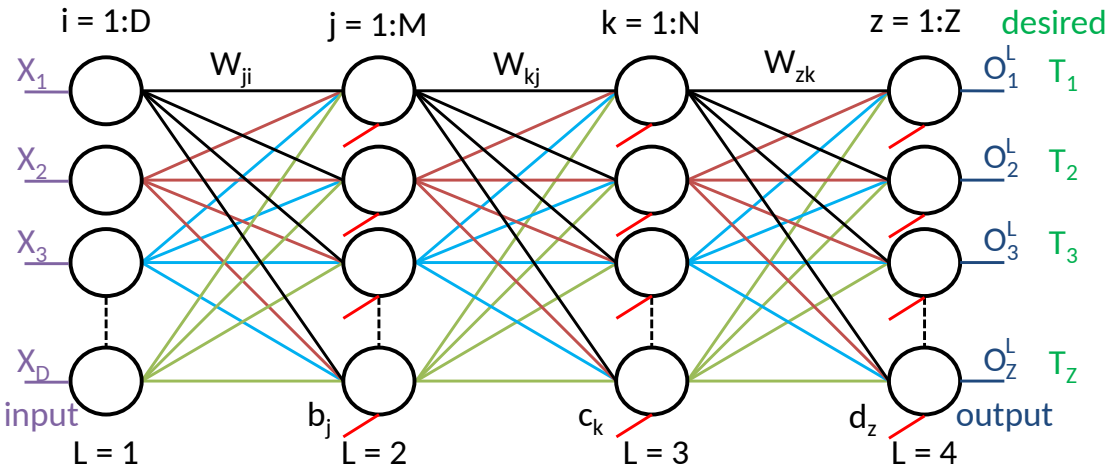

Figure A.1: Typical feed forward 4 layer neural network

Figure A.1 shows an arbitrary 4 layer feed forward neural network. There are D neurons in the input layer, M neurons in 2^{nd} layer, N neurons in 3^{rd} layer and Z neurons in the output layer. There is a weighted connection between each neuron in a layer to every other neurons in the layer immediately following it. Also there is a bias to every neurons in all layers except at the input layer. During training these parameters (weights and biases) need to be updated so as to minimise the loss function. A typical loss function could be the half of the sum of squared difference defined by

$$E = \sum_{z} \frac{1}{2}(T_z - O_z^L)^2 \tag{A.1}$$

Here T_z and O_z^L are the desired output and actual output produced at node z of the output layer L for any specific input $X = \{X_i\}_{i=1}^{D}$. Note that the notion of input to the normal feed forward neural network is a column vector. If the input is an image, it has to be vectorised to form the input vector. $\{X_i\}_{i=1}^{D}$ forms the

column vector X with D elements as X_i.

All nodes, except at the input layer, compute the weighted sum of inputs from the previous layer to produce an intermediate output Net. The activation function is applied to each of this Net to produce the final output at each neuron. We will consider the popular Sigmoid activation function defined by

$$O_z^L = f(Net_z^L) = \frac{1}{1 + \exp^{-Net_z^L}} \tag{A.2}$$

where Net_z^L is the intermediate output produced by node z in layer L and is defined by

$$Net_z^L = \sum_k W_{zk} O_k^{L-1} + d_z \tag{A.3}$$

Here O_k^{L-1} is the output at node k in $L-1^{th}$ layer and W_{zk} is the weights of the connections from node k in layer $L-1$ to node z in layer L.

We use gradient descent to update the parameters. Each parameter is updated in the negative direction of the gradient of the loss function (Eq. A.1) computed with respect to the parameter to be updated. The procedure is going to be explained in two steps. In the first step we will compute the contribution of the error to the over all loss function at each node. The error at each node in the output layer is first computed. Then the contribution by each node of the layers lower in the hierarchy is computed by backpropagating the error. In the second step, we will compute the gradient and update the parameters. Subsection A.2.1 discusses the error back propagation procedure while subsection A.2.2 discusses the parameter updating procedure. In all these discussion, we refer the final layer as L, pre-final layer as $L-1$, the next lower layer as $L-2$ and so on. The main theme of the presented formulation is that either error back propagation or the weight update based on it can be extended to any number of layers (on similar steps).

A.2.1 Backpropagating the Error across the layers

We will compute the error at each node in the output layer and is backpropgated to the layers lower in the hierarchy to determine the contribution of this error by each node in the network.

Finding Error at Output Layer L

Each node in the output layer contribute to the total error E. The contribution of Error by a node z at Layer L is due to its output O_z^L. Let ΔE_z^L denote this quantity and can be measured by computing $\frac{\partial E}{\partial O_z^L}$. By the definition of the error function (Eq. A.1), this turns out to be

$$\Delta E_z^L = \frac{\partial E}{\partial O_z^L} = O_z^L - T_z \tag{A.4}$$

Propagating the error from Layer L to layer $L-1$

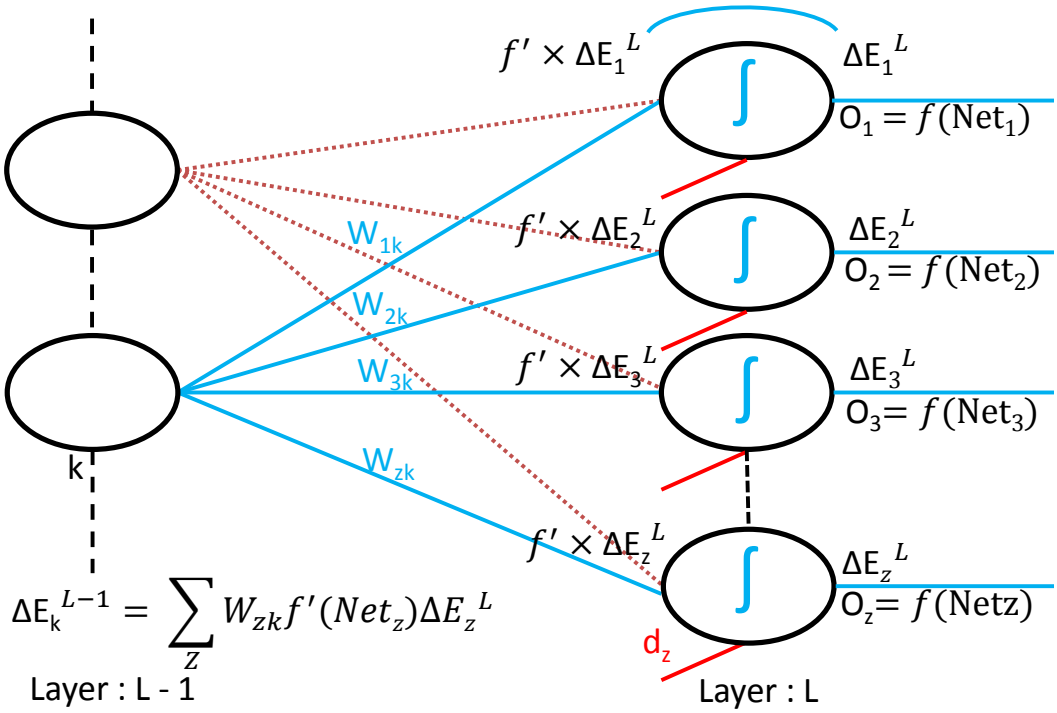

Figure A.2: Error backpropagation from nodes in layer L to node k in layer $L-1$.

First, we will give an intuitive idea behind the backpropagation of error and

then we will derive explicit expressions. In order to find the contribution of the final error E at layer L due to the output of a node k in layer $L-1$, the error that we have found out at each node in L has to be backpropagated. Once the error ΔE_z^L at all nodes z in an arbitrary layer L is computed, it has to cross these neurons during backpropagation of the error to compute the error at all nodes in the lower layer $L-1$. This is done by weighing ΔE_z^L by the corresponding derivative of the sigmoid function $f^{'}(Net_z^L)$, and then accumulating the shares through the weighted connections at each node k in layer $L-1$. The procedure can be explained better by referring to Fig A.2. It shows the relationship to each of ΔE_z^L from the node k in layer $L-1$. It can be seen that node k in layer $L-1$ influences the output at all nodes in L through the respective weight connections. Thus the error computed at each neuron in layer L contributes through the respective weight connection to each node k in layer $L-1$ and it has to cross each neuron z in layer L. Thus the Error at a node k in layer $L-1$ can be computed as

$$\Delta E_k^{L-1} = \frac{\partial E_z^L}{\partial O_k^{L-1}} = \sum_z \frac{\partial E_z^L}{\partial O_z^L} f^{'}(Net_z^L) W_{zk} \tag{A.5}$$

Where $f^{'}(Net_z^L$ represents the derivative of the activation function with respect to Net value at node z in layer L. By its definition (Eq A.2), this turns out to be $O_z^L(1-O_z^L)$. In matrix form, this can be written as

$$\Delta E^{L-1} = \mathbf{W^T} * \left[\Delta E^L . * f^{'}(Net^L)\right] \tag{A.6}$$

Where '.*' represents the element by element multiplication, '*' represents the matrix multiplication, and $\mathbf{T}$ represents the matrix transpose. The Eq. A.6 remains same for back propagation of error through any number of layers. The weight matrix $\mathbf{W}$ changes from layer to layer. The $f^{'}$ of the corresponding activation function (with associated outputs) and error at current layer are used to backpropagate the error to its previous layer.

The expression in Eq. A.5 can also be explicitly computed using the normal chain rule for gradient computation and is shown below.

$$\frac{\partial E_z^L}{\partial O_k^{L-1}} = \sum_z \frac{\partial E_z^L}{\partial O_z^L} \frac{\partial O_z^L}{\partial Net_z^L} \frac{\partial Net_z^L}{\partial O_k^{L-1}} \tag{A.7}$$

Where $\frac{\partial Net_z^L}{\partial O_k^{L-1}}$ can be computed as

$$\frac{\partial Net_z^L}{\partial O_k^{L-1}} = \frac{\partial}{\partial O_k^{L-1}} \sum_k W_{zk} O_k^{L-1} = W_{zk} \tag{A.8}$$

When substituted Eq. A.8 in Eq. A.7, it turns out that

$$\frac{\partial E_z^L}{\partial O_k^{L-1}} = \sum_z \frac{\partial E_z^L}{\partial O_z^L} \frac{\partial O_z^L}{\partial Net_z^L} W_{zk} \tag{A.9}$$

which is exactly the same equation provided in Eq. A.5. The consequence of Eq. A.6 is that the error backpropagation can be done independent of gradient update, and can be propagated back in each layers lower in the hierarchy one by one by simple matrix multiplication.

A.2.2 Updating the parameters

As noted earlier, the optimisation method used to minimise the loss function defined in Eq. A.1 could be gradient descent. The parameters are updated in the negative direction of the gradient of the error function computed with respect to the parameter to be updated.

Updating the weights

The update rule based on the gradient descent is

$$W_{zk}^{new} = W_{zk}^{old} - \eta \frac{\partial E}{W_{zk}} \tag{A.10}$$

Here η is the learning parameter and $\frac{\partial E}{W_{zk}}$ is the gradient of the loss function E with respect to the parameter W_{zk}. The $\frac{\partial E}{W_{zk}}$ can be computed using chain rule

$$\frac{\partial E}{\partial W_{zk}} = \frac{\partial E}{\partial O_z} \frac{\partial O_z}{\partial W_{zk}} \tag{A.11}$$

Here, the first term $\frac{\partial E}{\partial O_z}$ for all nodes in all layers is already computed by error backpropagation as discussed in the last section. The second term can be com-

puted by chain rule as

$$\frac{\partial O_z}{\partial W_{zk}} = \frac{\partial O_z}{\partial Net_z}\frac{\partial Net_z}{\partial W_{zk}} \tag{A.12}$$

Where $\frac{\partial O_z}{\partial Net_z}$ is the derivative of sigmoid function and is defined by

$$\frac{\partial O_z}{\partial Net_z} = O_z(1 - O_z) \tag{A.13}$$

$$\frac{\partial Net_z}{\partial W_{zk}} = \frac{\partial}{\partial W_{zk}}\sum_k W_{zk}O_k^{L-1} = O_k^{L-1} \tag{A.14}$$

W_{zk} is the weight connecting node k in layer $L-1$ to node z in layer L, $\frac{\partial Net_z}{\partial W_{zk}}$ turns out to be the output at the node k at layer $L-1$. Thus $\frac{\partial O_z^L}{\partial W_{zk}}$ turns out to be the output of the neuron k at layer $L-1$ weighted by the derivative of the sigmoid with respect to Net_z^L at node z in layer L.

Updating the weights for the network shown in Fig. A.1

Let $\mathbf{W}_{H_1}$, $\mathbf{W}_{H_2}$ and $\mathbf{W}_{H_3}$ be the weight matrices of the network holding respectively the weights between layer 1 & 2, 2 & 3 and 3 & 4.

$$W_{H_1} = \begin{bmatrix} W_{11} & W_{12} & W_{13} & . & . & . & W_{1D} \\ . & . & . & . & . & . & . \\ . & . & . & W_{ji} & . & . & . \\ . & . & . & . & . & . & . \\ W_{M1} & W_{M2} & W_{M3} & . & . & . & W_{MD} \end{bmatrix}$$

$$W_{H_2} = \begin{bmatrix} W_{11} & W_{12} & W_{13} & . & . & . & W_{1M} \\ . & . & . & . & . & . & . \\ . & . & . & W_{kj} & . & . & . \\ . & . & . & . & . & . & . \\ W_{N1} & W_{N2} & W_{N3} & . & . & . & W_{NM} \end{bmatrix}$$

$$W_{H_3} = \begin{bmatrix} W_{11} & W_{12} & W_{13} & . & . & . & W_{1N} \\ . & . & . & . & . & . & . \\ . & . & . & W_{zk} & . & . & . \\ . & . & . & . & . & . & . \\ W_{Z1} & W_{Z2} & W_{Z3} & . & . & . & W_{ZN} \end{bmatrix}$$

Let $O^4 = \{O_z\}_{z=1}^{Z}$, $O^3 = \{O_k\}_{k=1}^{N}$ and $O^2 = \{O_j\}_{j=1}^{M}$ be the output produced at each node in layers 4^{th}, 3^{rd} and 2^{nd} respectively when an input $X = \{X_i\}_{i=1}^{D}$ is fed to the network. The output at layer 1 is same as the input and hence $O^1 = \{O_i\}_{i=1}^{D} = \{X_i\}_{i=1}^{D}$. Also let $T = \{T_z\}_{z=1}^{Z}$ is the target or desired output at each node in the output layer. Now the error vector at layer l, Δ_l can be computed by error backpropagation.

$$\Delta_4 = \left[O^4 - T\right]_{[z\times 1]} \tag{A.15}$$

$$\Delta_3 = \left[\mathbf{W}_{H_3}^{\mathbf{T}}\right]_{N\times Z} * \left[O^4 .* (1 - O^4) .* \Delta_4\right]_{Z\times 1} \tag{A.16}$$

$$\Delta_2 = \left[\mathbf{W}_{H_2}^{\mathbf{T}}\right]_{M\times N} * \left[O^3 .* (1 - O^3) .* \Delta_3\right]_{N\times 1} \tag{A.17}$$

Note that Eq. A.16 and Eq. A.17 follows from the Eq. A.6.

Now the gradient at each neuron with respect to the parameter to be updated can be found and the final update rule in matrix form will be

$$\mathbf{W}_{H_3}^{new} = \mathbf{W}_{H_3}^{old} - \eta \left[O^4 .* (1 - O^4) .* \Delta_4\right]_{Z\times 1} * \left[O^3\right]_{1\times N}^{\mathbf{T}} \tag{A.18}$$

$$\mathbf{W}_{H_2}^{new} = \mathbf{W}_{H_2}^{old} - \eta \left[O^3 .* (1 - O^3) .* \Delta_3\right]_{N\times 1} * \left[O^2\right]_{1\times M}^{\mathbf{T}} \tag{A.19}$$

$$\mathbf{W}_{H_1}^{new} = \mathbf{W}_{H_1}^{old} - \eta \left[O^2 .* (1 - O^2) .* \Delta_2\right]_{M\times 1} * \left[O^1\right]_{1\times D}^{\mathbf{T}} \tag{A.20}$$

Thus to update the weight (W_{kj}) between neuron j in layer $L-2$ and neuron k in layer $L-1$, the gradient of the error function at the final layer with respect to W_{kj} need to be computed. It can be seen from the above set of equations that this gradient can be interpreted as the product of two terms. The first term is the error computed at node k in layer $L-1$ multiplied by the derivative of the sigmoid activation function with respect to the output at node k. This can be treated as

the error at the rear end of the connection holding the weight W_{kj}. The second term is the output at the node j in layer $L-2$, and can be considered as the input to the connection holding the weight W_{kj}. Thus, the change for updating the weight for a connection between any layers can be simply obtained by computing the product of error at rear end of the connection and input at front end of the connection. This is then weighted by the learning parameter η and the weight of the connection is updated by gradient descent.

A.3 Training CNN using backpropagation

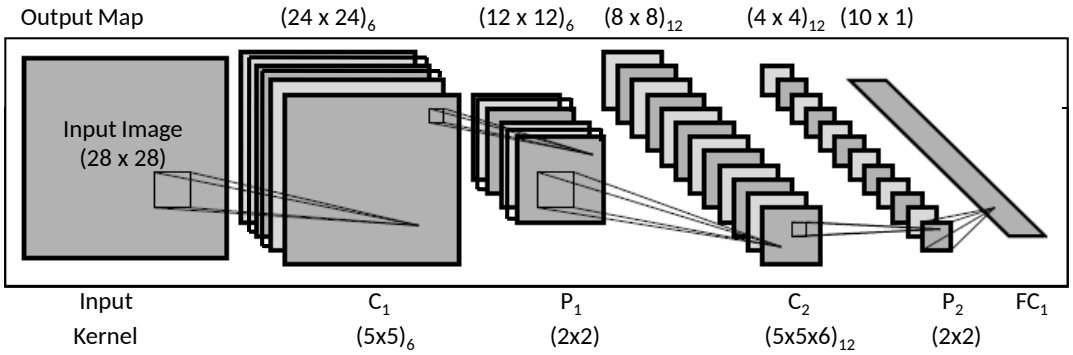

Figure A.3: LeNet architecture for digit recognition

The same backpropagation algorithm that we had discussed in last section is used to train convolutional neural networks as well. In this section, we will discuss this procedure taking the popular LeNet CNN architecture as an example. The LeNet architecture for digit recognition is shown in Fig. A.3. It has two convolution layers (C_1, C_2), two average pooling layers (P_1, P_2) and one fully connected layer (FC_1). There are 6 kernels in C_1 each of size 5×5 and 12 kernels in C_2 each of size $5 \times 5 \times 6$. The average pooling does a 2×2 pooling (thus the kernel weights are fixed as 0.25). The architecture was originally designed for digit recognition for input images of dimension 28×28. The corresponding output maps generated at each layer is shown above the individual blocks. Note that the output dimension is 10×1 since each input has to be mapped to one of the 10 digits. Note that the parameters to be learned in this CNN are 1) the weights of the kernels and their biases at the convolutional layers ($5 \times 5 \times 6 + 6 = 156$ in C_1

Table A.1: Parameters for Weight Initialisation for CNN in A.3

	C_1	C_2	FC_1
Fan_{in}	$1 \times (5 \times 5) = 25$	$6 \times (5 \times 5) = 150$	$12 \times (4 \times 4) = 192$
Fan_{out}	$6 \times (5 \times 5) = 150$	$12 \times (5 \times 5) = 300$	10
ζ	0.0756	0.0471	0.0704

and $5 \times 5 \times 6 \times 12 + 12 = 1812$ in C_2) and 2) the weights and biases at the fully connected layer ($4 \times 4 \times 12 \times 10 + 10 = 1930$).

A.3.1 Parameter Initialization

The parameters for each layer are initialised based on the number of input and output connections at that layer. Specifically these are initialised with random numbers selected from uniform distribution between $U[-\zeta, \zeta]$, where the bound ζ is determined by the fan_{in} and fan_{out} of the layer.

$$\zeta = \sqrt{\frac{1}{fan_{in} + fan_{out}}} \tag{A.21}$$

Here, for convolutional layers fan_{in} is defined as the product of number of input maps and kernel size and fan_{out} is defined as the number of output maps and kernel size. For fully connected layer, fan_{in} is the product of number of input maps and size of the input map while fan_{out} is the number output nodes. Table A.1 shows the parameters for weight initialisation for the CNN architecture shown in Fig. A.3.

A.3.2 Forward Propagation

The respective operations are performed on the input maps at each layer. At convolutional layers, each kernel is used to convolve with the input map, then kernel bias is used to offset the result, and then the sigmoid activation function is applied to produce an output map. Note that as discussed in section 6.4, only

valid part of result of convolution is used to generate the output map.

$$O[i] = \frac{1}{1 + \exp^{-\left(\sum_{d=1}^{D}[I(:,:,d) * K_i(:,:,d) + B(i)]\right)}} \tag{A.22}$$

In Eq. A.22, $*$ represents the convolution operation and $B(i)$ represents the bias of i^{th} kernel. $O[i]$ is the output map generated for i^{th} kernel when applied on the d^{th} input map. For convolutional layer 1, $D = 1$ and i varies from 1 to 6. For convolutional layer 2, $D = 6$ and i varies from 1 to 12. At the pooling layers, each 2×2 block is averaged to form a single pixel thereby reducing the output map size by 2 along each dimension. At the fully connected layer, the output is computed just like normal feed forward neural network as defined by the set of equations Eq. A.3 and Eq. A.2.

A.3.3 Backward Error Propagation

We use the same loss function defined in Eq. A.1. For any input image, we compute the result at the output layer which will be the 10×1 vector, and now we can compute $\frac{\partial E}{\partial O_k}$ using Eq. A.4. Now this error has to be backpropagated across different layers.

Error backpropagation across fully connected layer FC_1

The derivative of the loss function with respect to the output at any node k in the final layer $\frac{\partial E}{\partial O_k}$ is $O_k - T_k$. Thus the error in fully connected layer FC_1 is a 10×1 vector holding $(O_k - T_k)_{k=1}^{10}$.

$$\Delta E_{FC_1} = O - T \tag{A.23}$$

The error at nodes in the FC_1 has to be backpropagated to the pooling layer 2 (P_2). As per the Eq. A.9, this turns out to be ΔE_{S_2}

$$\Delta E_{P_2} = W. * O. * (1 - O). * \Delta E_{FC_1} \tag{A.24}$$

Here in Eq. A.24, '.*' represents the element by element multiplication. Here W is the weight matrix connecting 192×1 output map at P_2 to the 10×1 output vector. The 192×1 output map at P_2 is actually the vectorised representation of 12 maps each of size 4×4. Thus ΔE_{S_2} is of dimension $([192, 10] \times [10, 1] \rightarrow [192, 1])$. This is reshaped into $(4 \times 4)_{12}$ for representational convenience in propagating the error further down the layers.

Error backpropagation across pooling layer P_2

Since the pooling operation does an average pooling in 2×2 neighbourhood, the error backpropagation across this layer is just an up-sampling by 2.

$$\Delta E_{C_2} = US_2(\Delta E_{P_2}) \tag{A.25}$$

Where US_2 represents the operation, upsampling by 2. This leads to ΔE_{C_2} of size $(8 \times 8)_{12}$.

Error backpropagation across convolutional layer C_2

The error backpropagation across the convolution layer is also governed by the same equation Eq. A.9. But by the special construction of the convolution operation, the weights are related in a special way while propagating the error. It turns out to be the correlation of the kernel with the error to be backpropagated after weighted by the derivative of the activation function. First, we will show the correspondence between the correlation operation and the weighted error propagation by taking an example. Then we will give explicit equation for error backpropgation across C_2. Consider the following convolution operation (only valid part of convolution is considered) in Eq. A.26 and the correlation operation in Eq. A.27 (Note that there is adequate padding by zero to reconstruct the dimension of the input).

$$\begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ x_5 & x_6 & x_7 & x_8 \\ x_9 & x_{10} & x_{11} & x_{12} \\ x_{13} & x_{14} & x_{15} & x_{16} \end{bmatrix} * \begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix} = \begin{bmatrix} y_1 & y_2 \\ y_3 & y_4 \end{bmatrix} \tag{A.26}$$

Here

$$\begin{aligned} y_1 &= w_9x_1 + w_8x_2 + w_7x_3 + w_6x_5 + w_5x_6 + w_4x_7 + w_3x_9 + w_2x_{10} + w_1x_{11} \\ y_2 &= w_9x_2 + w_8x_3 + w_7x_4 + w_6x_6 + w_5x_7 + w_4x_8 + w_3x_{10} + w_2x_{11} + w_1x_{12} \\ y_3 &= w_9x_5 + w_8x_6 + w_7x_7 + w_6x_9 + w_5x_{10} + w_4x_{11} + w_3x_{13} + w_2x_{14} + w_1x_{15} \\ y_4 &= w_9x_6 + w_8x_7 + w_7x_8 + w_6x_{10} + w_5x_{11} + w_4x_{12} + w_3x_{14} + w_2x_{15} + w_1x_{16} \end{aligned}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \Delta y_1 & \Delta y_2 & 0 & 0 \\ 0 & 0 & \Delta y_3 & \Delta y_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \odot \begin{bmatrix} w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \\ w_7 & w_8 & w_9 \end{bmatrix} = \begin{bmatrix} \Delta x_1 & \Delta x_2 & \Delta x_3 & \Delta x_4 \\ \Delta x_5 & \Delta x_6 & \Delta x_7 & \Delta x_8 \\ \Delta x_9 & \Delta x_{10} & \Delta x_{11} & \Delta x_{12} \\ \Delta x_{13} & \Delta x_{14} & \Delta x_{15} & \Delta x_{16} \end{bmatrix} \tag{A.27}$$

Here

$$\begin{aligned} \Delta x_1 &= w_9\Delta y_1; & \Delta x_2 &= w_9\Delta y_2 + w_8\Delta y_1; & \Delta x_3 &= w_8\Delta y_2 + w_7\Delta y_1; \\ \Delta x_4 &= w_7\Delta y_2; & \Delta x_5 &= w_9\Delta y_3 + w_6\Delta y_1; & \Delta x_8 &= w_7\Delta y_4 + w_4\Delta y_2; \end{aligned}$$

$$\begin{aligned} \Delta x_{13} &= w_3\Delta y_3; & \Delta x_9 &= w_6\Delta y_3 + w_3\Delta y_1; & \Delta x_{12} &= w_4\Delta y_4 + w_1\Delta y_2; \\ \Delta x_{16} &= w_1\Delta y_4; & \Delta x_{14} &= w_3\Delta y_4 + w_2\Delta y_3 & \Delta x_{15} &= w_2\Delta y_4 + w_1\Delta y_3; \end{aligned}$$

$$\begin{aligned} \Delta x_6 &= w_9\Delta y_4 + w_8\Delta y_3 + w_6\Delta y_2 + w_5\Delta y_1; & \Delta x_7 &= w_8\Delta y_4 + w_7\Delta y_3 + w_5\Delta y_2 + w_4\Delta y_1; \\ \Delta x_{10} &= w_6\Delta y_4 + w_3\Delta y_2 + w_5\Delta y_3 + w_2\Delta y_1; & \Delta x_{11} &= w_5\Delta y_4 + w_2\Delta y_2 + w_4\Delta y_3 + w_1\Delta y_1; \end{aligned}$$

By analysing the above equations, it can be seen that the relationship between the weights and the output during convolution is reproduced during correlation and

hence can be used in backpropagating the error. For example, input x_6 at layer L influences all the output neurons (y_1, y_2, y_3, y_4) at layer $L+1$ during convolution operation through weights w_5, w_6, w_8 and w_9 respectively. Therefore, when back-propagating the error Δy computed at layer $L+1$, the error contribution at node corresponding to x_6 should be the aggregate sum of the error at $\Delta y_1, \Delta y_2, \Delta y_3$ and Δy_4 weighted exactly by the same weights w_5, w_6, w_8 and w_9. By analysing the expression for Δx_6 obtained after the correlation operation, it can be seen that this relationship is preserved. Thus back propagating the error across the convolution layers is equivalent to performing the correlation operation on the weighted error computed for the layer after weighting the derivative of the activation function. Thus ΔE_{P_1} can be computed as

$$\Delta EF_{C_2} = \Delta E_{C_2} .* O_{C_2} .* (1 - O_{C_2}) \tag{A.28}$$

$$\Delta E_{P_1}(:,:,i) = \sum_{l=1}^{L} [\Delta EF_{C_2}(:,:,l) \odot K_l(:,:,i)] \tag{A.29}$$

Where $\odot$ represent the correlation operation, i varies from 1 to 6 and $L = 12$. This will result in backpropagated error dimension as $(12 \times 12)_6$.

Error backpropagation across pooling layer P_1

As discussed earlier, the back propagated error is just an up-sampled version.

$$\Delta E_{C_1} = US_2(\Delta E_{P_1}) \tag{A.30}$$

A.3.4 Learning the Parameters by Gradient Descend

Once the error is propagated for all the neurons, the gradient is computed with respect to the parameter to be updated. The same set of equations provided in Eq. A.11 is used to compute the gradient. The parameters are then updated in the negative direction of the gradient to minimise the loss function.

Derivative of gradient with respect to weights at fully connected layer

$$(\Delta W)_{FC_1} = \Delta E_{FC_1} .* O .* (1 - O) .* I_{FC_1} \tag{A.31}$$

In Eq. A.31, I_{FC_1} represent the vectorized output map at the pooling layer 2 (P_2). Thus the dimension of the gradient is $[10, 1] \times [1, 192] \rightarrow [10, 192]$

Derivative of gradient with respect to kernel weights at convolution layers

$$\Delta EF_{C_l} = \Delta E_{C_l} . * O_{C_l} . * (1 - O_{C_l}) \tag{A.32}$$

$$(\Delta K)_l = \Delta EF_{C_l} * I \tag{A.33}$$

In Eq. A.33, '*' is the convolution operation. $l = 1$ for convolution layer 1 and $l = 2$ for convolution layer 2. I is the corresponding input map to the layer. Since we are taking only valid part of convolution, for C_1, the gradient dimension will be $[5, 5]_6$ ($[24, 24]_6 * [28, 28] \rightarrow [5, 5]_6$). Similarly for convolution layer 2, the gradient dimension will be $[5, 5, 6]_{12}$ ($[8, 8]_{12} * [12, 12]_6 \rightarrow [5, 5, 6]_{12}$)

Updating the kernel weights

Once the gradient is determined with respect to each parameter, we can use gradient descent so as to minimise the loss function.

$$K^{new} = K^{old} - \eta \Delta K \tag{A.34}$$

The gradient for the bias term for any node in the output layer, turns out to be the cumulative weight change computed for all the connections to that neuron. Similarly, the gradient for the bias of the kernels turns out to be the cumulative weight change computed for the weights in the respective kernel.

LIST OF PAPERS BASED ON THESIS

Journal Publications

1. V.K. Jagannadh, **G. Gopakumar**, Gorthi R.K Sai Subrahmanyam, S.S. Gorthi. "Microfluidic microscopy-assisted label-free approach for cancer screening: automated microfluidic cytology for cancer screening", ***Medical & Biological Engineering & Computing***, **55**:711, 2017. DOI: 10.1007/s11517-016-1549-y (Publisher : **Springer**, Impact Factor : 1.916)

2. **G. Gopakumar**, V.K. Jagannadh, S.S. Gorthi, Gorthi R.K Sai Subrahmanyam. "Framework for morphometric classification of cells in imaging flow cytometry", ***Journal of Microscopy***, **261**:307–319, 2016. DOI: 10.1111/jmi.12335 (Publisher : **Wiley**, Impact Factor : 2.136)

3. **G. Gopakumar**, K. Haribabu, Deepak Mishra, S.S. Gorthi, Gorthi R.K Sai Subrahmanyam. "Cytopathological image analysis using deep learning networks in microfluidic microscopy", ***Journal of the Optical Society of America A***, **34(1)**:111–121, 2017. DOI: 10.1364/JOSAA.34.000111 (Publisher : **OSA**, Impact Factor : 1.457)

4. **G. Gopakumar**, M. Swetha, S.S. Gorthi, Gorthi R.K Sai Subrahmanyam. "CNN based malaria diagnosis from focus-stack of blood smear images acquired using custom-built slide scanner", ***Journal of Biophotonics***, 2017. DOI: jbio.201700003 (Publisher : **Wiley**, Impact Factor : 4.328)

Conference Publications

1. **G. Gopakumar**, G.R.K.S. Subrahmanyam, S.S. Gorthi. "Morphology based classification of leukaemia cell-lines: K562 and MOLT in a microfluidics based imaging flow cytometer". Ninth Indian conference on computer vision, graphics and image processing (***ICVGIP'14***), pp. 34:1–34:7, **ACM** 2014.

2. **G. Gopakumar**, M. Swetha, S.S. Gorthi, G.R.K.S. Subrahmanyam. "Automatic Detection of Malaria Infected RBCs from a Focus Stack of Bright Field Microscope Slide Images". Tenth Indian conference on computer vision, graphics and image processing (***ICVGIP'16***), pp.16:1–16:7, **ACM** 2016.

3. K. S. Kalmady, S. K. Adithya, **G. Gopakumar**, Gorthi R.K. Sai Subrahmanyam, S.S. Gorthi. "Improved transfer learning through shallow network embedding for classification of Leukemia cells", Ninth International Conference on Advances in Pattern Recognition (***ICAPR'17***), **IEEE** 2017 (Accepted)