APPLICATIONS OF MACHINE LEARNING IN BIOPHOTONICS AND LASER METROLOGY

Roopam Kumar Gupta

A Thesis Submitted for the Degree of PhD
at the
University of St Andrews

2021

Applications of machine learning in biophotonics and laser metrology

Roopam K. Gupta

This thesis is submitted in partial fulfilment for the degree of

Doctor of Philosophy (PhD)

at the University of St Andrews

October 2020

Declaration

Candidate's Declaration

I, Roopam K. Gupta, do hereby certify that this thesis, submitted for the degree of PhD, which is approximately 36,700 words in length, has been written by me, and that it is the record of work carried out by me, or principally by myself in collaboration with others as acknowledged, and that it has not been submitted in any previous application for any degree.

I was admitted as a research student at the University of St Andrews in October 2016.

I received funding from an organisation or institution and have acknowledged the funder(s) in the full text of my thesis.

Date: 26 January 2021

Signature:

Supervisor's Declaration

I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of PhD in the University of St Andrews and that the candidate is qualified to submit this thesis in application for that degree.

Date: 26 January 2021

Signature:

Permission for publication

In submitting this thesis to the University of St Andrews we understand that we are giving permission for it to be made available for use in accordance with the

regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. We also understand, unless exempt by an award of an embargo as requested below, that the title and the abstract will be published, and that a copy of the work may be made and supplied to any bona fide library or research worker, that this thesis will be electronically accessible for personal or research use and that the library has the right to migrate this thesis into new electronic forms as required to ensure continued access to the thesis.

I, Roopam K. Gupta, confirm that my thesis does not contain any third-party material that requires copyright clearance.

The following is an agreed request by candidate and supervisor regarding the publication of this thesis:

Printed Copy

Embargo on all of print copy for a period of 2 years on the following ground(s):

- Publication would preclude future publication

Supporting statement for printed embargo request

My research was sponsored/funded by a commercial company and there are conditions on my research being made publicly available.

Electronic Copy

Embargo on all or part of electronic copy for a period of 2 years on the following ground(s):

- Publication would preclude future publication

Supporting statement for electronic embargo request

My research was sponsored/funded by a commercial company and there are conditions on my research being made publicly available.

Title and Abstract

I agree to the title and abstract being published.

Signature of Candidate: Date: 26 January 2021

Signature of Supervisor: Date: 26 January 2021

Underpinning Research Data or Digital Outputs

Candidate's declaration

I, Roopam K. Gupta, understand that by declaring that I have original research data or digital outputs, I should make every effort in meeting the University's and research funders' requirements on the deposit and sharing of research data or research digital outputs.

Date: 26 January 2021 Signature of Candidate: _____

Permission for publication of underpinning research data or digital outputs

We understand that for any original research data or digital outputs which are deposited, we are giving permission for them to be made available for use in accordance with the requirements of the University and research funders, for the time being in force.

We also understand that the title and the description will be published, and that the underpinning research data or digital outputs will be electronically accessible for use in accordance with the license specified at the point of deposit, unless exempt by award of an embargo as requested below.

The following is an agreed request by candidate and supervisor regarding the publication of underpinning research data or digital outputs:
Embargo on all of electronic files for a period of 2 years on the following ground(s):

- Publication would preclude future publication

Supporting statement for embargo request

My research was sponsored/funded by a commercial company and there are conditions on my research being made publicly available.

Signature of Candidate: Date: 26 January 2021

Signature of Supervisor: Date: 26 January 2021

"There is nothing noble in being superior to your fellow men, true nobility is being superior to your former self."

Ernest Hemingway

Abstract

Recently, optical technologies have found several applications in fields including biophotonics, precision metrology and wavelength scale sensors. However, to gather statistically relevant information and analysis these methods require large amount of measurements. Current linear multivariate methods such as principal component analysis or linear discriminant analysis are not sufficient to analyze these big datasets with non linear variability. Recently, the application of deep learning based artificial neural networks have found an upsurge in various areas of science ranging from quantum physics to evolutionary biology, providing an enhancement in the efficiency of various techniques. This thesis focuses on the applications of machine learning with the goal to enhance different aspects of biophotonics.

Firstly, this thesis explores the application machine learning to enhance the label-free characterization of cells of the immune system using Raman spectroscopy and digital holographic microscopy. The combination of deep learning with digital holographic microscopy provides a route towards a high throughput hemogram device which would be useful for the classification of clinically important immune cells with morphological similarities but different functions.

Following this, the applications of deep learning are explored in the regime of precision optical metrology for the development of a laser speckle wavemeter with a high dynamic range with an additional application for the development of a binary speckle based spectrometer.

Finally, the application of machine learning based methods are also explored to improve the sensitivity of the chirped guided mode biosensor. A comparison between the linear method of principal component analysis and direct Fano fitting is drawn which is followed by the application of multi layered perceptron for further improvement.

To my loving parents, Mrs Rama Gupta and Mr. Rajendra Kumar Gupta whose words of encouragement and push for tenacity ring in my ears.

Acknowledgements

I would like to acknowledge my supervisors Dr. Simon Powis and Prof. Kishan Dholakia who have provided me with the opportunity to work in this multi-disciplinary and very interesting scientific research. During the last four years as a PhD student under them, I learned new skills and gained expertise in various aspects of research.

With the excellent teaching and training provided by Dr. Simon Powis, I learned the cell culture techniques even when I had no experience in them. Prof. Kishan Dholakia provided me with the training to think outside the box in not only optics but in general to solve the problems. Because of both my supervisors support, I could gain expertise in the novel field of deep learning and could think about its applications in various directions.

I would like to thank my colleague Dr. Mingzhou Chen for his expert advises and insights throughout my PhD. He not only offered to discuss about Raman spectroscopy but in general gave me time to discuss about other studies as well. I would also like to thank Prof. Thomas Krauss and Dr. Kezheng Li for providing me support and insightful discussions for the collaborative project.

Last but not the least, I would like to thank my family. My father Mr. Rajendra Kumar Gupta, my mother Mrs. Rama Gupta, my brother Mr. Rohit Gupta, my sister in law Mrs. Varsha Gupta and my loving wife Mrs. Sakshi Varshney for their consistent support throughout my PhD.

This work was supported by Medical Research Scotland [Grant Ph.D. 873-2015], which provided an opportunity of an industrial PhD with M Squared Lasers.

Publications

Peer reviewed publications

1. **Gupta R. K.**, Chen M., Malcolm G. P. A., Hempler N., Dholakia K. and Powis S. J., "*Label–free optical hemogram of granulocytes enhanced by artificial neural networks*", Opt. Express 27, 13706-13720 (2019)

2. **Gupta R. K.**, Bruce G. D., Powis S. J. and Dholakia K., "*Deep learning enabled laser speckle wavemeter with a high dynamic range*", Laser & Photonics Reviews 2020, 14, 2000120 / arXiv:1910.10702 [physics.optics]

3. **Gupta R. K.**, Malcolm G. P. A., Hempler N., Dholakia K. and Powis S. J., "*High throughput hemogram of T–cell subsets using digital holographic microscopy and deep learning*", (in preparation)

Conferences and courses attended

1. 8th International graduate summer school- Biophotonics '17, June 2017, poster "*Label-free micro-spectroscopic characterization of cells of the immune system*"

2. 15th Conference on Optics Within Life Sciences, November 2018, contributed poster "*Label-free identification of the granulocytes enhanced by machine learning*"

3. International Conference on Biophotonics, May 2019, contributed poster "*Label-free optical hemogram of granulocytes enhanced by artificial neural networks*"

4. SPIE conference BIOS: High-Speed Biomedical Imaging and Spectroscopy V, January 2020, contributed talk "*High throughput label-free optical hemogram of granulocytes enhanced by artificial neural networks*"

Contents

List of Figures

List of Abbreviations

ANN Artificial Neural Network

GMR Guided Mode Resonance

ML Machine Learning

RL Reinforcement Learning

PCA Principal Component Analysis

LDA Linear Discriminant Analysis

t-SNE t-Stochastic Neighbourhood Embedding

MDP Markov Decision Process

MC Monte Carlo

TD Temporal Difference

DP Dynamic Programming

SVM Support Vector Machine

DL Deep Learning

MLP Multi Layered Perceptron

CNN Convolutional Neural Network

RNN Recurrent Neural Network

BPTT Back Propagation Through Time

LSTM Long Short Term Memory

GAN Generative Adversarial Network

DHM Digital Holographic Microscopy

WMRS Wavelength Modulated Raman Spectroscopy

OCT Optical Coherence Tomography

FFT Fast Fourier Transform

PBS Phosphate Buffer Saline

FBS Foetal Bovine Serum

CCD Charge Coupled Device

LOOCV Leave One Out Cross Validation

NA Numerical Aperture

CB Convolution Block

OPD Optical Path Difference

PBMC Peripheral Blood Mononuclear Cell

FOV Field Of View

ReLU Rectified Linear Unit

PSO Particle Swarm Optimization

GPU Graphics Processing Unit

TP True Positive

TN True Negative

FP False Positive

FN False Negative

TM Transmission Matrix

AOM Acousto Optic Modulator

SMF Single Mode Fibre

CMOS Complementary Metal Oxide Semiconductor

DSB Down Sampling Block

FC Fully Connected

PCE Probabilistic Classification Error

IgG Immunoglobulin G

Chapter 1

Introduction

1.1 Preface

Since the invention of light microscope in the 17^{th} century, biophotonics has developed into a popular field of study. Essentially, biophotonics refers to the interaction of light with biological samples which may also include any food related items in order to study the underlying aspects. For the analytic purposes, it can further be divided with respect to the dimensions of interest. These include single cells, tissues, organs and the whole body. To understand the underlying phenomenon, biophotonics has been a rapidly growing field with great potential to become a part of clinical trials. Following the invention of lasers in the 1960s, several optical techniques including imaging and spectroscopy have advanced exponentially. These advancements have found corresponding applications in biophotonics to provide it with an unparalleled outlook [1, 2].

One such biomedical field in which the light based techniques have shown promise is immunology. The understanding of human and animal physiological mechanisms for protecting their body against external environmental factors provides a great potential for direct clinical applications. After getting infected by any agent, the type and number of immune cells are altered which makes it medically very relevant to identify these cells and quantify their numbers. These white blood cells, also known as leucocytes, can be further divided into granulocytes, monocytes and lymphocytes. Each cell type has their own functionality and they act in a coordinated manner to combat infection [3].

Current methods of identifying and classifying the immune cells require labelling the cells using fluorescent tags or magnetic beads. These are expensive, time consuming and destructive. To overcome such shortcomings, optical techniques such as Raman spectroscopy, digital holographic microscopy, optical coherence tomography, optical diffraction tomography among others have been employed [4–11].

In order to achieve a statistically significant outcome, these techniques require a large number of measurements leading to exponential growth in the amount of biological data. This in turn results in the problems of efficient information storage and extraction of suitable information. The extraction of suitable information poses the main challenge requiring the development of modern and more reliable approaches. Specifically for the classification of samples, the information of non-linear variability in the sensitivity and specificity requires the development of tools and methods which can be capable of transforming data to understand the underlying biological mechanism. The application of computational approaches to analyse the biological data have been practiced in several biological domains including Genomics, Evolution, Systems biology, etc.

The requirement for flexibility and adaptability towards different types of data has been achieved by employing machine learning based methods. The ability for automated learning without explicitly defining the rules provided these methods with an edge against any other techniques. Initial methods for the analysis were applied by considering multivariate methods such as principal component analysis or linear discriminant analysis. However, these techniques being linear, present major disadvantage when applied for non-linearly categorized datasets - which is a general trend in biomedical data. Artificial neural networks (ANNs) and deep learning present the latest advancement in machine learning. For their ease of applicability for different types of data, ANNs and deep learning have found a plathora of application in, not only bioinformatics but also other fields of photonics.

The focus of this thesis is to explore the applications of machine learning and find the solutions which may enhance the working of optical instruments. Beginning with the application of machine learning in label – free characterization of immune cells, this thesis explores the applications of machine learning in precision optical metrology and for sensitivity enhancement in wavelength - scale devices.

1.2 Synopsis

This thesis will describe the research in the field of optics and photonics with the particular emphasis on the applications of machine learning to enhance and improve the optical devices.

Chapter 2 describes the fundamentals of machine learning with a brief overview of its applications in the field of optics. The characterization of machine learning into sub-types, such as, supervised, un-supervised and reinforcement learning are discussed. This chapter then introduces and provides a theoretical background to different linear and non-linear methods. Following this, the fundamentals of deep learning which includes feed forward networks, convolutional neural networks, recurrent neural networks and generative adversarial networks are discussed with an overview of a variety of their applications in optics.

Chapter 3 will provide a brief introduction to the fundamentals of Immunology and label-free optical techniques. Specifically the fundamentals and applications of Raman spectroscopy and digital holographic microscopy for the characterization of immune cells. This chapter then explains the research, as two studies, conducted to enhance the classification abilities of optical techniques using machine learning. The first study demonstrates the enhancement of optical hemogram for granulocyte cells using artificial neural networks. The second study explores the classification of $CD4^+$ and $CD8^+$ T cells using digital holographic microscopy. It explores the application of particle swarm optimization to identify optimal convolutional neural network (CNN) geometry and the application of cycle generative adversarial networks to enhance the throughput rate whilst maintaining the resolution.

Chapter 4 will describe the research conducted to develop a laser speckle based wavemeter. This chapter starts with a discussion of methods to accumulate the data and optimization of a CNN geometry. Then three research studies are described for the development of laser speckle wavemeter and spectrometer. First study explores the development of a speckle wavemeter based upon the understanding of wavelength modulation of the incident laser beam. Second study is the continuation from the first study to develop a speckle wavemeter with attometre scale precision and high dynamic range. Interesting aspect demonstrated in this study is the automated noise cancellation learned by the CNN after being trained as a wavemeter. The

third study describes the development of a speckle spectrometer which is capable of identifying two simultaneous wavelengths from a single speckle image.

Chapter 5 will demonstrate the application of machine learning methods to improve the sensitivity of a chirped guided mode resonance (GMR) biosensor device. This chapter is divided into two studies. First study explains the application of principal component analysis by considering the variation of resonance response from the GMR with respect to the incident light wavelength. This resulted in a two fold improvement compared to the generic Fano fitting method. The second study shall display the application of multi-layered perceptron to classify the resonance response of GMR with respect to the incident light beam. As a result, a possible improvement of 10,000 fold is reported when compared to the Fano fitting method.

The concluding chapter will give a summary of various studies described in the thesis. This chapter shall also circumscribe a discussion on future directions that would enhance the applicability of machine learning in various areas of optics and photonics.

Chapter 2

Machine learning: An overview

2.1 Introduction

The search for patterns in data is a fundamental problem which has found several important applications throughout history. Going back to the 16^{th} century, astronomical observations by Tycho Brahe allowed for the discovery of emperical laws of planetary motion leading to the development of classical mechanics. In the early twentieth century, the observation of regularities in atomic spectra played a key role in the development and verification of quantum physics. The automatic identification of regularities present in the data proved to be crucial in the discovery of fundamental aspects of sciences. This requirement of identifying the patterns in data (especially where the size of data set is very large) demands the automatic methods of analysis, which is provided by *Machine learning* (ML). ML is a sub-field of artificial intelligence which encompasses the study of computational algorithms that improve automatically without being explicitly programmed [12]. For simple tasks assigned to computers, it is possible to program algorithms which inform the machine to execute all steps required to solve the problem at hand without "learning". For more advanced tasks, such as email filtering, computer vision or natural language processing, it becomes very challenging to manually develop the needed algorithms. In practice, it seems to be more effective to employ statistical models in order to generalize over the patterns present in the data. Computational algorithms based on ML generally build statistical models based on a sample data, commonly referred as "training data", to identify their intrinsic trends and features.

The trained models are then employed to make predictions on a separate data-set with similar trends ("testing data").

The discipline of ML employs various approaches which may teach the machines to achieve tasks where no other algorithm performs well. Coined in 1959 by Arthur Samuel, the term machine learning has been commonly used in the recent scientific contributions. ML has found various applications in biomedical engineering, neuroimaging, fundamental physics, computational geoscience, and chemistry [13–18].

This chapter will give a brief introduction to the theory of machine learning and an overview of its applications in optics and photonics. The chapter starts with an introduction to the types of machine learning, which provides insights into supervised, unsupervised and reinforcement learning approaches. Then different methods of machine learning including principal component analysis, linear discriminant analysis, k-means algorithm, support vector machines, t-distributed stochastic neighbour embedding and artificial neural networks are discussed. Following which the fundamentals of deep learning are discussed which includes a brief overview and applications in the context of optics and photonics for feed forward networks, convolutional neural networks, recurrent neural networks and generative adversarial networks.

2.2 Types of Machine Learning

This section provides a detailed outlook of the types of machine learning approaches implemented and types of problems that are solved using them. Machine learning is generally divided into three main categories, namely, supervised learning, unsupervised learning and reinforcement learning [19].

The supervised learning approach deals with learning a mapping from inputs x to the outputs y by using a labelled set of input-output pairs $\mathcal{Q} = \{(x_i, y_i)\}_{i=1}^{N}$, here $\mathcal{Q}$ is the training set and N are the total number of training examples. Generally, x_i is a one, two or higher dimensional vector with k points. This vector can take the form of an image, a sentence, a time series, a sequence of digits, etc. These may represent the data directly or the features, attributes or covariates representing the intricate properties of the data. The output variable y_i, can in principle be

any real valued continuous variable or a discrete categorical representation of the input data. When y_i takes the form of a categorical vector, the problem is known as a *classification problem* whereas when y_i is a real-valued continuous scalar, the problem is known as a *regression problem.*

The second type of machine learning is descriptive or unsupervised learning approach. This methodology generally deals with identifying hidden patterns in a given dataset $\mathcal{Q} = \{x_i\}_{i=1}^{N}$. Due to the absence of any apparent error metric, this is not a well-defined problem. Generally, these problems are formalised in the form of density estimation by utilizing multivariate probability models.

Reinforcement learning (RL) is the third approach of ML which resembles the closest to the kind of learning that humans and other animals do. It deals with mapping the situations (states $\mathcal{S}$) to actions $\mathcal{A}$ in order to maximize a numerical reward signal $\mathcal{R}$. This algorithm works on the principle of trade-off between exploration and exploitation. The approach is to train a learning agent to interact (by taking actions) with its environment hence changing its state and yielding a reward.

2.2.1 Supervised learning

The majority of practical machine learning problems are solved using supervised learning approach. As explained before, this approach implements an algorithm which learns to map a function from input variable x to the output y as $y = f(x)$. Simply, the goal is to approximate the mapping function such that the input variables x_i can be used to predict the output variables y_i for a given data. Knowing the correct answers, an algorithm iteratively makes prediction on the training data and is corrected with respect to the error identified after each attempt. The learning process stops when an acceptable level of performance is reached over the novel inputs, hence showing the generalization capabilities of the algorithm.

Depending on the type of variable y, a supervised learning problem can either be considered as a classification problem or a regression problem.

Regression Problem

The class of regression problems deals with the prediction of one or more continuous variables y by considering the value of a D-dimensional input vector x. Linear

functions of the input variables form the simplest form of linear regression. However, a linear combination of non-linear functions of input variables can also be considered to form the basis functions. These models form linear functions of the parameters with simple analytical properties and non-linear with respect to the input variables.

For a given training data set with N observations x_n and targets y_n where $n = 1, ..., N$, the aim for using a regression approach is to predict the value of y for a new value of x. This problem can be formalised by directly constructing a function $y(x)$ whose values for new inputs x constitute the predictions for the corresponding values of y

$$y(x, w) = w_0 + w_1 x_1 + ... + w_D x_D \tag{2.1}$$

where $x = (x_1, ..., x_D)^T$. This type of model is known as linear regression such that it is a linear function of the parameters $w_0, ..., w_D$. In order to remove the limitations of the model, for being linear with respect to x_i, it is extended by considering the nonlinear functions of x_i, of the form

$$y(x, w) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(x) \tag{2.2}$$

here $\phi_j(x)$ are known as basis functions with the total number of parameters being equal to M. The functions exhibiting the form of Eq. 2.2, are called linear models, for their linear dependence on w. Depending on the given problem, a number of possible basis functions can be considered such as polynomial functions where $\phi_j(x) = x^j$, spline functions [20], Gaussian basis functions with $\phi_j(x) = e^{-\frac{(x-\mu_j)^2}{2s^2}}$; here μ_j governs the location of basis functions and s governs their spatial scale. Another choice is the sigmoidal basis function or the tan hyperbolic function which have the form:

$$\phi_j(x) = \sigma\left(\frac{x - \mu_j}{s}\right) \tag{2.3a}$$

$$\tanh(a) = 2\sigma(a) - 1 \tag{2.3b}$$

$$\sigma(a) = \frac{1}{1 + e^{-a}} \tag{2.3c}$$

here, $\sigma(a)$ is called the logistic sigmoid function. The 'tanh(x)' function is consid-

ered as it is related to logistic sigmoid (Eq. 2.3b), which implies that a general linear combination of logistic functions is equivalent to general linear combination of 'tanh(x)' functions. Fourier basis leading to an expansion in sinusoidal functions is also a popular choice as basis functions [21–23]. Each basis function represents a specific frequency with finite spatial extent. Signal processing applications have found it to be of interest to consider the basis functions that are localized in both space and frequency, leading to a class of functions known as wavelets [24–26].

Classification Problem

For a classification type of problem, the aim is to assign the input variable x to one of the $\mathcal{K}$ discreet classes $\mathcal{C}_k$ where $k = 1, ..., \mathcal{K}$. This problem can be formalised as a function approximation $y = f(x)$, mapping the inputs x to outputs y, such that $y \in \{1, ..., \mathcal{K}\}$, with $\mathcal{K}$ being the number of classes. For the common cases where the classes are mutually exclusive and disjoint, the latent space of input data is divided into decision boundaries. The cases with $\mathcal{K} = 2$ are called binary classification problems whereas the cases with $\mathcal{K} > 2$ are referred to as multi-class classification problems. The models implemented to solve classification problems try to predict output probabilities that lie in the range (0,1). To achieve this, a generalized model can be considered which transforms a linear function of parameters $\mathbf{w}$ using a non-linear function f, so that

$$y(x) = f(w^T x + \omega_0) \tag{2.4}$$

here f is known as an activation function. The class boundary/surface correspond to $y(x) = \kappa$, such that $w^T x + \omega_0 = \kappa$ (κ is a constant) and hence the class surfaces are linear function of x, even if the function f is nonlinear. For this reason, the class of models described in Eq. 2.4, are called generalized linear models [27].

There are three distinct approaches to tackle the classification problem, (1) constructing a discriminant function to assign each vector x to a specific class, (2) modelling the conditional probability distribution $p(y|x)$ in the inference stage and then using this distribution to make optimal decision and (3) modelling the class-conditional densities $p(x|y)$, together with $p(y)$ and then computing the required probabilities using Bayes' theorem [28]

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \tag{2.5}$$

Next few sections shall discuss about these approaches in the form of well known methods of machine learning.

2.2.2 Unsupervised learning

The problems of machine learning where only output results are known without any inputs, are dealt upon by implementing unsupervised learning. In order to identify the hidden features of the data, this approach works to determine the underlying distribution of data, known as *density estimation* or to discover groups of similar examples within the data, known as *clustering*, or to project the data from a high-dimensional space to two or three dimensions for the purpose of visualization, known as *dimensionality reduction*.

Density estimation

The density estimation problem is dealt by building an unconditional multivariate probability model of the form $p(x)$ of a random variable x, given a finite set of observations $x_1, ..., x_N$. Since there is a possibility of infinite probability distributions for a for a finite data set, the problem of density estimation is fundamentally ill-posed. The models considered for the density estimation of the underlying data can be thought of as parametric or non-parametric.

The application of parametric models such as, Gaussian distribution, Dirichlet distribution and others require a procedure to determine the values for parameters. For the initialization of the parameters, two types of treatments are considered, namely, *frequentist treatment* where specific values for the parameters are chosen by optimizing some criterion such as the likelihood function and the *Bayesian treatment* where a prior distribution is introduced over the parameter and Bayes' theorem (Eq. 2.5) is used to compute the corresponding posterior distribution. The parametric models, however, show a limitation of assuming specific form of the initial function for the distribution which may apply some restrictions for particular applications.

The non-parametric models based on histograms, nearest-neighbours or kernel,

result in the distributions which typically depend upon the size of input data set. The parameters contained by these types of models control the model complexity rather than the form of distribution.

Clustering

Generally, the problem of clustering for a data set $x \in \{x_1, ..., x_N\}$ is solved by considering the number of clusters (say $\mathcal{K}$) as a parameter. Then, a probability distribution is evaluated over the number of clusters, $p(\mathcal{K}|x)$ which informs about the presence of any sub-populations in the underlying data. Selection of a model $\mathcal{M}$ is performed by considering $\mathcal{M} = \arg\max_{\mathcal{K}} \ p(\mathcal{K}|x)$. A model is selected for further evaluation if the optimal number of clusters represent the statistical properties of the data. Model selection is followed by identification of cluster for each point. For this purpose, a latent variable, say $z_i \in \{1, ...\mathcal{K}\}$ is considered and a computation $z_i^* = \arg\max_{k} \ p(z_i = k|x_i)$ is performed to isolate the cluster containing this point. The major advantage of this approach is the comparison of multiple models with respect to the statistical significance of the data, generally considered as a function of likelihood.

There are a few popular clustering algorithms such as hierarchical clustering which builds a multilevel hierarchy of clusters by creating a cluster tree [29], k-Means clustering which partitions data into k distinct clusters based on distance to the centroid of a cluster [30, 31], and Self-organizing maps which uses neural networks that learn the topology and distribution of the data [32].

Dimensionality reduction

Analysing the raw high dimensional data ($x_i \in \{x_1, ..., x_N\}$) often leads to massive computational cost. To deal with such data, the dimensionality reduction approach becomes a requirement. It is, however, important to keep the metric properties of the data consistent while reducing its dimensionality. For keeping the important features, for e.g. variance, of data in reduced dimensions, the lower dimensional data ($z_i \in \{z_1, ...z_N\}$) can further be used as an input to other statistical models for better predictive accuracy. Additionally, these can also be very useful to visualize the higher dimensional data which is difficult to visualize in its original form.

A few algorithms have been popularly implemented for the purpose of dimensionality reduction. Two of the most popular linear methods are principal component analysis (PCA) and linear discriminant analysis (LDA). PCA reduces the dimension of data by maximizing the variance of each dimension [33] whereas LDA identifies a linear combination of features that characterize the subgroups of a given data set [34]. Nonlinear methods popularly used for this approach are autoencoders and t-stochastic neighbourhood embedding (t-SNE). Autoencoders learn non-linear dimension reduction function and its inverse by reconstructing the input data [35] whereas t-SNE models a higher-dimensional datapoint using a two or three dimensional point such that the probabilisitic distribution of the input data set is conserved [36].

2.2.3 Reinforcement learning

Reinforcement learning (RL) [37] approach of ML is based upon identifying given actions in a given environment to maximize a reward. In contrast to supervised learning where the models are given training examples of output, RL based agents are set to discover the outputs by low-level learning of trial and error or by high-level deliberative planning based learning. At each interaction of RL agent with the environment there is a change in state associated with a reward. This reward then encourages the agent to take action in a direction which yields maximum reward. Whilst RL is a massive topic and requires 100s of pages of discussion, this sub-section briefly describes the fundamentals of RL which shall be useful in understanding the applications in future chapters.

Fundamentally, RL based approach comprises of four elements: a *policy*, a *reward signal*, a *value function* and a *model* of environment. *Policy* (π) can be understood as a conditional probability from incident state $\mathcal{S}$ of environment $\mathcal{E}$ to actions $\mathcal{A}$. Formally, this can be defined as $\pi(\mathcal{A}|\mathcal{S})$, $\forall(\mathcal{S},\mathcal{A}) \in \mathcal{E}$. Simply, in psychological terms, policy can be defined as set of stimulus-response rules. A *reward signal* $\mathcal{R}$ can be defined as the stochastic function $\mathcal{R}(\mathcal{S},\mathcal{A})$ of the environment with respect to an action $\mathcal{A}$. In biological terms, reward signal can be understood as analogous to feelings of pleasure or pain. The reward signal forms a basis for variations in policy; if for a given action, with respect to the policy, the environment results in

a low reward, then the policy is modified such that the next action may result in higher future reward for similar state to action transitions, hence it provides the immediate response to the action. The *value function* is defined by the culmination of rewards acquired by an RL model for a particular state. This function is responsible for making long term decisions with the purpose of achieving larger rewards. It is desirable by an RL model to seek actions with result in highest value in contrast to higher rewards. Efficiently estimating the values represents the most important component for a given RL problem. The final element of an RL based approach is the *model* of environment. Model of an environment features its characteristics, functionality and behaviour. Simply, model of an environment provides the computational playground for the RL agent to explore and exploit.

Finite Markov decision processes

Generally, the RL problems can be set up as optimal control of the incompletely-known Markov decision process (MDP). A stochastic process where the conditional property (with respect to present and past states) of the future states only depend on the present state is called a Markov process. The processes of sequential decision making are classically formalized as MDPs. Here, the present actions by an agent influence the subsequent states/situations following an immediate reward signal and resulting in future or delayed rewards. This results in MDPs involving delayed rewards and a trade-off between immediate and delayed rewards.

In an MDP, the agent interacts with the environment at discrete or continuous time steps. For simplicity, this discussion will be restricted to discreet time steps which can easily be generalized to continuous case. At each time step t, the agent is input with a state $S_t \in \mathcal{S}$ of the environment such that it selects an action $A_t \in \mathcal{A}(\text{s})$. On the next time step, the agent receives a reward, $R_t \in \mathcal{R}$ for moving into the state S_{t+1}. This process leads to a sequence of trajectory inside the environment as, $S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, ...$ also popularly known as $SARSA$ trajectory. For a finite MDP, the states, actions and rewards have a finite number of elements hence a discrete conditional probability distribution, which is solely dependent on

the preceding state and action, can be defined as:

$$p(s', r|s, a) \doteq \mathrm{P}\{S_t = s', R_t = r|S_{t-1} = s, A_{t-1} = a\} \tag{2.6}$$

Here, p is also known as the dynamic deterministic function with four arguments, $p : \mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ of an MDP such that,

$$\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r|s, a) = 1, \forall s \in \mathcal{S}, a \in \mathcal{A}(s) \tag{2.7}$$

Using the dynamic function defined in Eq. 2.6, quantities such as state transition probabilities ($p : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$, Eq. 2.8a), expected reward for state-action pair ($r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, Eq. 2.8b) and expected reward for state-action-next state triples ($r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, Eq. 2.8c) can be easily defined.

$$p(s'|s, a) \doteq \sum_{r \in \mathcal{R}} p(s', r|s, a) \tag{2.8a}$$

$$r(s, a) \doteq \sum_{r \in \mathcal{R}} \sum_{s' \in \mathcal{S}} p(s', r|s, a) \tag{2.8b}$$

$$r(s, a, s') \doteq \sum_{r \in \mathcal{R}} r \frac{p(s', r|s, a)}{p(s'|s, a)} \tag{2.8c}$$

The above defined MDP framework can be understood as an abstraction of a goal-oriented learning problem via interactions. This kind of problem can be reduced to three fundamental signals of the environment $\mathcal{E}$ starting from an initial state signal ($\mathcal{S}$), secondly, the actions ($\mathcal{A}$(s)) taken by the agent and finally the reward signal $\mathcal{R}$ which defines the goal of the problem. However, until now the immediate rewards are discussed. Generally, the complete RL problem is divided into *episodes* such that the agent gains rewards for discreet instances in a episode. The aim of the agent is to maximize the cumulative rewards from the episodes in the long run. This cumulative reward can be defined as the sum of rewards for distinct time instances:

$$R_t^c \doteq R_{t+1} + R_{t+2} + ... + R_T \tag{2.9}$$

where T is the final time step which is a random variable and varies for each episode. This final step is defined when the interaction between agent and environment leads

to a break in the episode leading to a terminal state. After the episode is finished, the environment is reset to its initial state and the process is repeated for the next episode. For the cases where the episodes do not have a defined action for ending the episodes, the final time step may not be defined such that $T = \infty$. To model the uncertainty of agent towards the terminal state of the environment, the discount factor γ is considered which generally represents the weightage of future rewards. An important point to note is that the value of γ is bound to be smaller than one, i.e, $\gamma < 1$ such that the sum of mentioned infinite series could be finitely evaluated. For these cases, the cumulative reward is calculated as:

$$\begin{aligned} R_t^c &\doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + ... \\ &= R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + ...) \\ &= R_{t+1} + \gamma R_{t+1}^c \end{aligned} \tag{2.10}$$

After understanding the notion of reward and cumulative reward, it would be interesting to understand the formal definitions of value and policy functions. As mentioned earlier, the value function is an estimate based on a given policy on how well the agent may perform to gain better rewards for a longer period. The policy, as formalised earlier, is the conditional probability ($\pi(a|s)$) that an action ($\mathcal{A}_\mathrm{t} = a$) will be taken for a given state ($\mathcal{S}_\mathrm{t} = s$). On the other hand, the value function for a state s ($v_\pi(s)$) under a policy π is defined to evaluate the expected future rewards:

$$\begin{aligned} v_\pi(s) &\doteq \mathbb{E}_\pi[R_t^c | S_t = s] \\ &= \mathbb{E}_\pi\left[R_{t+1} + \gamma R_{t+1}^c | S_t = s\right] \\ &= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a)[r + \gamma \mathbb{E}_\pi[R_{t+1}^c | S_{t+1} = s']] \forall s \in S \end{aligned} \tag{2.11}$$

here, $\mathbb{E}_\pi[\cdot]$ denotes the expectation value of the random variable for a given policy π at a given time instant t. The equation 2.14, further results in the Bellman equation for v_π which delivers the relationship between the value of the present state and the values of the successive states:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s', r} p(s', r|s, a)[r + \gamma v_\pi(s')] \tag{2.12}$$

The Bellman equation (Eq. 2.12) provides a means to estimate an optimal value

$$v_{\text{opt}}(s) \doteq \max_{\pi} v_{\pi}(s) \tag{2.13}$$

for each state by averaging over all the probabilistically weighted possibilities. However, calculation of an optimal value does not guarantee that the RL agent would act optimally over each state, hence it is required to identify an optimal policy ($q_{\text{opt}}(s, a)$) value which is an action value function for policy π. The action value function can be defined as:

$$\begin{aligned} q_{\pi}(s, a) &\doteq \mathbb{E}_{\pi}[R_t^c | S_t = s, A_t = a] \\ &= \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a\right], \forall s \in \mathcal{S}, a \in \mathcal{A} \end{aligned} \tag{2.14}$$

This value provides the RL agent an insight to take make best decision (in terms of actions) with respect to a given state. Using the Eq. 2.13, the optimal value of action value function $q_{\text{opt}}(s, a)$ can be evaluated as:

$$q_{\text{opt}}(s, a) = \sum_{s', r} p(s', r|s, a)[r + \gamma \max_{a'} q_{\text{opt}}(s', a')], \forall \quad s' \in \mathcal{S}^1 \tag{2.15}$$

Here, $\max_{a'} q_{\text{opt}}(s', a')$ is the value of optimal action a' at the state s' and $\mathcal{S}^1$ denotes the set of states including a terminal state.

Dynamic Programming, Monte Carlo methods and Temporal difference learning

With an assumption of a perfect model, *dynamic programming* represents the collection of algorithms which can be used to compute optimal policies of an environment. Quite simply, it can also be understood as breaking of large problems into incremental steps to identify the optimal solutions of sub-problems at any given stage with an imperative requirement of a well posed model.

In non-ideal situations, the fundamentals of RL such as states $\mathcal{S}$, actions $\mathcal{A}$, dynamic function $p(s', r|s, a)$, immediate reward $\mathcal{R}$ and the discount factor γ are not all known for the same instant. The evaluation of these parameters can be achieved by initializing a random policy π and gaining episodic experiences. These

experiences can either be gained by following a policy and maintaining an average of rewards, this results in the estimation of state value function $v_\pi(s)$, or by averaging over different actions for each state and then averaging the reward, which results in action value function $q_\pi(s, a)$. These kind of estimation methods, fundamentally based upon hit and trial approach, are called the *Monte Carlo* (MC) methods.

The essential requirement for an MC based approach is the presence of a terminal state. In contrast to the Bellman equation, the values for each state are update solely based the final cumulative reward R_t^c and not on the neighbour states. The cases where a model is available, state value function is evaluated whereas if the model is not available, then action value function is estimated. To evaluate the state value function $v_\pi(s)$, of a state s with a policy π, two types of MC models are considered, first-visit MC method and every-visit MC method. The first-visit MC method estimates $v_\pi(s)$ as the average of rewards following the first visits to state s whereas, the every-visit MC method averages the rewards following all the visits to state s. The cases where a model is not available, the sole estimation of state values is not sufficient and an explicit evaluation of action values is required to identify an optimal policy for the problem. This may also be achieved by implementing the above mentioned first-visit and every-visit MC methods. Here, both methods estimate the rewards followed by the visits to state-action pair. However, due to a large number of possibilities, it is possible to miss visiting quite a few state-action pairs. Following a deterministic policy π, there is a possibility to observe the rewards for only one action from each state which result in a circumstance with zero improvement. This is a general problem of maintaining *exploration*. This problem can be overcome by initializing the state-action pair with a condition of a non-zero selection probability for each pair. Doing so, it is assumed that for an infinite number of episodes, all the state-action pairs shall be visited for an infinite number of times. In conclusion, the MC algorithm helps to estimate strategy for the problem where the model is unknown. However, this technique comes with a disadvantage of updating the strategy after the termination of each episode without complete use of an MDP learning structure.

The central idea to RL, *temporal-difference* (TD) learning, is a combination of MC method and dynamic programming. TD methods work without a defined model

of the environment and estimates rewards/returns without waiting for a terminal state. In contrast to MC methods which wait until the end of an episode to determine an estimate V of v_π for a non-terminal state S_t, TD methods wait only until the next step and are updated as:

$$V(S_t) \leftarrow V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)] \tag{2.16}$$

via a transition to S_{t+1} while receiving R_{t+1}. Here, α is a constant step-size parameter which may influence the convergence. This type of TD method is called TD(0) method which is a special case of TD(λ) method. In the above, the value $R_{t+1} + \gamma v_\pi(S_{t+1})$ is called TD target value and $R_{t+1} + \gamma v_\pi(S_{t+1}) - v_\pi(S_t)$ is called the TD error.

The generalization of TD method takes the form of TD(λ) also known as n-step TD methods. These methods generalize to smoothly meet the demands of a particular task which are not fulfilled by either TD(0) or MC method. MC method updates value function for each state based on the entire sequence of observed rewards from that state until the episode is terminated. On the other hand, TD(0) method updates the value based on the next reward. The TD(λ) method on the other hand lies between the two extremes by considering λ-time steps where the values are updated based on the given problem. Formally, a cumulative reward is considered for λ steps and the value is updates thereafter which can be generated as:

$$R_t^{c(1)} = R_{t+1} + \gamma V(S_{t+1}) \tag{2.17a}$$

$$R_t^{c(2)} = R_t + 1 + \gamma R_{t+2} + \gamma^2 V(S_{t+2}) \tag{2.17b}$$

$$R_t^{c(n)} = R_t + 1 + \gamma R_{t+2} + ... + \gamma^n V(S_{t+n}) \tag{2.17c}$$

$$R_t^{c(\lambda)} = (1-\lambda)R_t^{c(1)} + (1-\lambda)\lambda R_t^{c(2)} + ... + (1-\lambda)\lambda^{n-1} R_t^{c(n)} \tag{2.17d}$$

$$\implies R_t^{c(\lambda)} = (1-\lambda)\sum_{n=1}^{\infty} \lambda^{n-1} R_t^{c(n)} \tag{2.17e}$$

Hence the TD(λ) update rule can be formalised as:

$$V(S_t) \leftarrow V(S_t) + \alpha[R_t^{c(\lambda)} - V(S_t)] \tag{2.18}$$

Depending on the given problem, an optimal value of λ can be chosen with respect to an objective function or the rewards received while solving the problem. The policy evaluation using this method can be achieved by considering λ-step SARSA. With respect to the λ-step returns,

$$R_t^{c(\lambda)} \doteq R_{t+1} + \gamma R_{t+2} + ... + \gamma^{\lambda-1} R_{t+\lambda} + \gamma^{\lambda} Q_{t+\lambda-1(S_t,A_t)} \tag{2.19}$$

The above condition is valid for $\lambda \geq 1$ and $0 \leq t < T - \lambda$. The policy value $Q_{t+\lambda}$ can be defined as:

$$Q_{t+\lambda}(S_t, A_t) \doteq Q_{t+\lambda-1}(S_t, A_t) + \alpha[R_t^{c(\lambda)} - Q_{t+\lambda-1}(S_t, A_t)], 0 \leq t < T \tag{2.20}$$

TD methods present massive advantages over DP and MC methods. With respect to working with unavailability of a model of environment, these methods present advantage over DP. On being applied in an incremental and online fashion, these methods provide advantage over MC methods. With MC methods, it is an imperative requirement to wait until the termination of an episode, whereas with the simplest form of TD methods the only requirement is to wait for one time step.

RL has found several applications in physics. It has been shown to efficiently decode uncorrelated quantum errors in toric codes, to efficiently identify the controls to improve the precision of quantum parametric estimation, for frame by frame tracking of plaque using intravascular optical coherence tomography, to correct piston misalignment between segments in a segmented mirror telescope, to obtain approximation of eigenvectors of arbitrary Hermitian quantum operator [38–42].

For its outstanding applicability to a plethora of problems, RL has been implemented with deep learning where the agents, value function and policy functions are approximated using the deep neural networks. This field has found attention in research applications that encompass both supervised and unsupervised learning [43].

2.3 Methods of machine learning

In this section, different methods of machine learning shall be discussed. Starting from the linear methods of ML which include PCA, LDA, k-means clustering and

support vector machines; this section will also cover the non-linear methods namely t-SNE and artificial neural networks.

2.3.1 Principal component analysis

Principal component analysis (PCA) is a highly-celebrated statistical procedure to identify the presence of patterns in any kind of data set. PCA uses an orthogonal transformation to convert a set of correlated variables into eigenvalue - eigenvector problem; whereby the eigenvectors are called principal components. The number of principal components is generally less than the number of original variables.

PCA is implemented by collecting a data set in n dimensions. To remove the effects of intensity fluctuations, mean is subtracted from the collected data. Following which, a covariance matrix is calculated, and the eigenvectors and eigenvalues are extracted from this matrix.

First principal component describes the maximal variation which is directly associated with the eigenvector with largest eigenvalue. The first principal component P_{c1} can be calculated as the eigenvector with largest eigenvalue μ of the covariance matrix C_{ik}:

$$\mu P_{c1}(\nu_i) = C_{ik} P_{c1}(\nu_k) \tag{2.21}$$

where the covariance matrix is defined as:

$$C_{ik} = \sum_{j=1}^{N} (R_j(\nu_i) - \hat{R}(\nu_i))(R_j(v_k) - \hat{R}(v_k)) \tag{2.22}$$

$\hat{R}(v_i)$ and $\hat{R}(v_k)$ are the average of the data points for the variable parametric components ν_i and ν_k respectively.

The calculation of eigenvectors with their corresponding eigenvalues, help in the dimensionality reduction of the data such that the degree of variance is conserved in the reduced dimensions. For these remarkable abilities, PCA has found several applications in biophotonics such as in the analysis of fiber optic gyroscope vibration error, for the reconstruction of leaf reflectance spectra and retrieval of leaf biochemical contents, for the diagnosis of gastric cancer using hyperspectral imaging, for modulation format identification scheme in elastic optical networks, and for energy

reconstruction in position-sensitive semiconductor detectors [44–48].

2.3.2 Linear discriminant analysis

Linear discriminant analysis (LDA), a generalization of Fisher's linear discriminant, is a linear method of ML based on finding the linear combination of characteristic features for the classification of input data set. This algorithms is generally used for reducing the dimensionality of data and then linearly classifying it.

LDA is closely related to PCA as these both are the linear methods which work on the principal of linear combination of variables to reduce the dimensionality of the given data set. The difference between the two is that PCA identifies the degree of variance, representing the differences, between the classes whereas LDA models the similarities of various components in terms of features or patterns. LDA is used when the prior knowledge of classes in the underlying data is known whereas PCA can be implemented without any prior knowledge.

Formally, LDA can be understood as a form of Fisher's linear discriminant with the assumptions of normally distributed classes or equal class covariances. Considering a training set of observations $x_i \in \{x_1, x_2, ...\}$ where x_i is a k dimensional vector with $k \geq 1$ representing the attributes of a given object with class $y \in \{1, 2, ..., K\}$. LDA based analysis assumes a normal distribution of probability density functions with mean parameters to be μ_y and identical covariances C with full rank. With respect to the Bayesian rule, a discriminant function δ_y for each class y can be constructed

$$\delta_y(x_i) = \log f_y(x_i) + \log \pi_y \tag{2.23}$$

with prior probabilities π_y to represent the maximum likelihood among all the classes. A class boundary is defined where discriminant functions of different classes have the same value. With respect to the assumption of identical covariance matrix for each class, the discriminant function is modified to give:

$$\delta_y(x_i) = x_i^T C^{-1} \mu_y - \frac{1}{2} \mu_y^T C^{-1} \mu_y + \log \pi_y \tag{2.24}$$

The class boundaries calculated using Eq. 2.24, present to be the linear functions of x and hence this method is called linear discriminant analysis.

As a popular classification method, LDA has found several applications in biophotonics. It have been implemented to differential between ablation states using diffusely reflected spectra of tissue samples for for gastric mucosal devitalization treatments, for three-dimensional discriminant analysis of hyperspectral images, and for classification of Raman spectra of multiple genitourinary cancers [49–51].

2.3.3 k-means algorithm

k-means algorithm is a linear method of supervised machine learning which is based upon vector quantization. This method aims to partition the input data set into k distinct non-overlapping clusters such that each data point belongs to a given cluster with respect to the cluster's mean.

This algorithm follows the approach of expectation-maximization, such that the assignment of data points ($x_i \in \{x_1, x_2, ..., x_n\}$) is made by keeping the sum of squared distance to the cluster's centroid to be at a minimum. It is indeed beneficial to have a minimum variation within the clusters. An iterative approach is followed by first specifying the number of clusters k, then centroids for each cluster are initialized by randomly shuffling the data and selecting random k datapoints which may serve as centroids. This process is iterated until no change in the assignment of centroid is detected. Simply an objective function

$$J = \sum_{i=1}^{n} \sum_{k=1}^{K} w_{ik} |x_i - \mu_k|^2 \tag{2.25}$$

is considered. Here, $w_{ik} \in \{0, 1\}$ for the cases that the data point x_i belongs to a cluster k. The objective function J is minimized in two steps, first it is minimized with respect to w_{ik} keeping μ_k (Eq. 2.26a) fixed and then it is minimized with respect to μ_k keeping w_{ik} fixed (Eq. 2.26b).

$$\frac{\partial J}{\partial w_{ik}} = \sum_{i=1}^{n} \sum_{k=1}^{K} |x_i - \mu_k|^2 \implies w_{ik} = \begin{cases} 1, & \text{if } k = \arg\min\limits_{j} |x_i - \mu_j|^2 \\ 0, & \text{otherwise} \end{cases} \tag{2.26a}$$

$$\frac{\partial J}{\partial \mu_k} = 2 \sum_{i=1}^{n} w_{ik}(x_i - \mu_k) \implies \mu_k = \frac{\sum_{i=1}^{n} w_{ik} x_i}{\sum_{i=1}^{n} w_{ik}} \tag{2.26b}$$

With respect to the set of equations 2.26, k-means algorithm utilizes a distance based measurements to determine the similarity among data points. For this, it is required to normalize the data set with zero mean and unit standard deviation. Given the iterative nature of the algorithm, random initialization of centroids may converge to a local minima. Hence, multiple runs of algorithm need to be considered and the results with least squared error must be considered.

For its generic applicability k-means algorithm has found many applications. It has been applied to probabilistic-shaped 64 quadrature amplitude modulation coherent optical communication system to locate the decision points more accurately, for optic disc segmentation, and as a benchmark for digital staining of optical coherence tomography images [52–54].

2.3.4 Support vector machine

Support vector machines [55] are the supervised learning models of ML that are popularly implemented for classification and regression analysis [56–58]. SVM is a non-probabilistic binary classifier such that for a given training set (x_i,y_i) $\forall\ y \in \{-1, 1\}$, this algorithm builds a model which assigns new examples to one of the categories. This model represents a hyperplane with a classification boundary (learned during training phase) for the training data points. The novel data points are projected in the hyperplane such that for an optimal model these data points spread on either sides of the class boundary.

The SVM algorithm, works by considering the functional margins with respect to the training examples (x_i, y_i),

$$\gamma_i = y_i(w^T x_i + b) \tag{2.27}$$

here, γ_i represents the margin function of the parameters w and b. With respect to the equation 2.27, w represents an orthogonal vector to the class boundary (γ_i), and b is a linear parameter representing the shift of this boundary. For a general case, the class boundary can be evaluated as,

$$\gamma_i = y_i \left(\left(\frac{w}{|w|} \right)^T x_i + \frac{b}{|w|} \right) \tag{2.28}$$

The geometric margin with respect to the given training data set with n elements is given as $\gamma = \min_{i=1,...,n} \gamma_i$. For a linearly separable training set, the evaluation of parameters (w, b) is done by maximizing γ such that each training example may satisfy the condition of having a class boundary at γ. To efficiently solve this problem and to evaluate an optimal margin classifier, the value of γ is scaled to 1 since it can also be satisfied by scaling w, b and the problem is then transformed to minimize $|w|^2$ given that $y_i(w^T x_i + b) \geq 1$ with $i \in \{1, ..., n\}$. By solving the Lagrangian dual of this problem a simplified problem of maximizing

$$f(c_1, ..., c_n) = \sum_{i=1}^{n} c_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i c_i (x_i \cdot x_j) y_j c_j \tag{2.29}$$

subject to $\sum_{i=1}^{n} c_i y_i = 0$ and $0 \leq c_i \leq \frac{1}{2n\lambda}$ for all i is obtained. Here, λ determines the trade off between increasing the margin size and ensuring that x_i lies on the correct side of margin. The variables c_i are defined such that $w = \sum_{i=1}^{n} c_i y_i x_i$. The parameter b can be evaluated by finding an x_i on the margin's boundary and solving

$$y_i.(w \cdot x_i - b) = 1 \tag{2.30}$$

To deal with a training data with non-linear distribution, a transformation is performed such that the transformed data presents the property of linear separability. The transformation function $\phi(\vec{x_i})$ must satisfy the condition for a given kernel function k,

$$k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \tag{2.31}$$

The kernel function may take the form of a Gaussian function $k(x_i, x_j) = e^{-|x_i - x_j|^2}$, a linear function $k(x_i, x_j) = x_i^T x_j$, a polynomial function $k(x_i, x_j) = (1 + x_i^T x_j)^\zeta$ of order ζ, or any other specifically defined valid form with respect to the Mercer theorem. By invoking the kernels, the problem transforms to maximizing

$$\begin{aligned} f(c_1, ..., c_n) &= \sum_{i=1}^{n} c_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i c_i (\phi(x_i) \cdot \phi(x_j)) y_j c_j \\ &= \sum_{i=1}^{n} c_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i c_i k(x_i, x_j) y_j c_j \end{aligned} \tag{2.32}$$

such that, $\sum_{i=1}^{n} c_i y_i = 0$ and $0 \leq c_i \leq \frac{1}{2n\lambda}$ for all i. The parameter b can be evaluated

by considering the condition that $\phi(\vec{x_i})$ lies on the class boundary in the transformed space and solving,

$$\begin{aligned} b &= \vec{w} \cdot \phi(\vec{x_i}) - y_i \\ &= \left[\sum_{j=1}^{n} c_j y_j k(\vec{x_i}, \vec{x_j}) \right] - y_i \end{aligned} \tag{2.33}$$

The coefficients c_i can be solved in a straightforward fashion using quadratic programming.

To this end, SVMs have found several applications in photonics, for the classification of hyperspectral data, for infrared target fusion detection, and for remote sensing image classification [59–62].

2.3.5 T-distributed stochastic neighbor embedding

T-distributed stochastic neighbor embedding (t-SNE) [36] is a recently developed non-linear method of unsupervised ML. This algorithm is particularly used for visualizing a data set in a low-dimensional latent space by reducing the dimensionality. The dimensionality reduction is achieved by constructing a probability distribution over the pairs of higher-dimensional data points which is followed by constructing a similar probability distribution for the points in lower-dimension and minimizing the Kulllback-Leibler divergence between the two distributions.

For a high-dimensional data set $x_1, ..., x_n$, t-SNE works to first evaluate a probability distribution $p_{i,j}$ proportional to the similarity of the data points x_i and x_j,

$$p_{j|i} = \frac{e^{(-|x_i - x_j|^2/2\sigma_i^2)}}{\sum_{k \neq i} e^{(-|x_i - x_k|^2/2\sigma_i^2)}}, \text{s.t.} p_{i|i} = 0 \tag{2.34a}$$

$$p_{i,j} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{2.34b}$$

here, $p_{i,j} = p_{j,i}$, $p_{i,i} = 0$ and $\sum_{i,j} p_{i,j} = 1$. The parameter σ_i is set using the bisection method such that the perplexity of conditional distribution equals a predefined perplexity.

To map the similarities of higher-dimensional data into a lower d-dimensional data $y_1, ..., y_n$ such that $y \in \mathbb{R}^d$, this algorithm constructs a probability distribution

$q_{i,j}$,

$$q_{i,j} = \frac{(1 + |y_i - y_j|^2)^{-1}}{\sum_{k \neq i}(1 + |y_i - y_k|^2)^{-1}} \tag{2.35}$$

Here, a heavy-tailed student t-distribution is used to measure the similarities between low dimensional points. The locations of data points y_i are determined using minimizing the non-symmetric form of Kullback-Leibler divergence of the distribution P from Q,

$$\mathrm{KL}(P||Q) = \sum_{i/nej} p_{ij} \log \frac{p_{i,j}}{q_{i,j}} \tag{2.36}$$

The minimization of KL divergence with respect to data points y_i is performed using gradient descent.

For the remarkable ability to reduce the dimensionality of the data while keeping the metric properties consistent, t-SNE has found various applications in computer security research, music analysis, in biomedical research and to obtain the latent space representations learned by artificial neural networks [63–67].

2.3.6 Artificial neural networks

The current technological era presents the onset of massive data sets which demonstrate the requirement of faster, robust, automated and reliable methods to analyse the data. The solution to the above is by using the latest computing methodology of artificial neural networks (ANNs) which are inspired by the biological neural networks constituting the animal brains.

The ANNs are based on the collection of fundamental units called artificial neurons. These neurons are capable of both transmitting and receiving signals via connections (synapses) as is observed in their biological analogues. These connections form the parameters of the network. The receiving neuron processes the signal(s) and transmit to the next neuron in the form of real numbers. The synaptic connections, which are the weights and bias parameters, represent the memory of ANN. These parameters vary continuously with the learning process. Typically, the neurons are organized into layers such that different layers perform different transformations on their input. These signals enter the ANN from the first layer (generally known as input layer) and travel through with successive processing to the last layer (known as the output layer). To understand the functionality of the

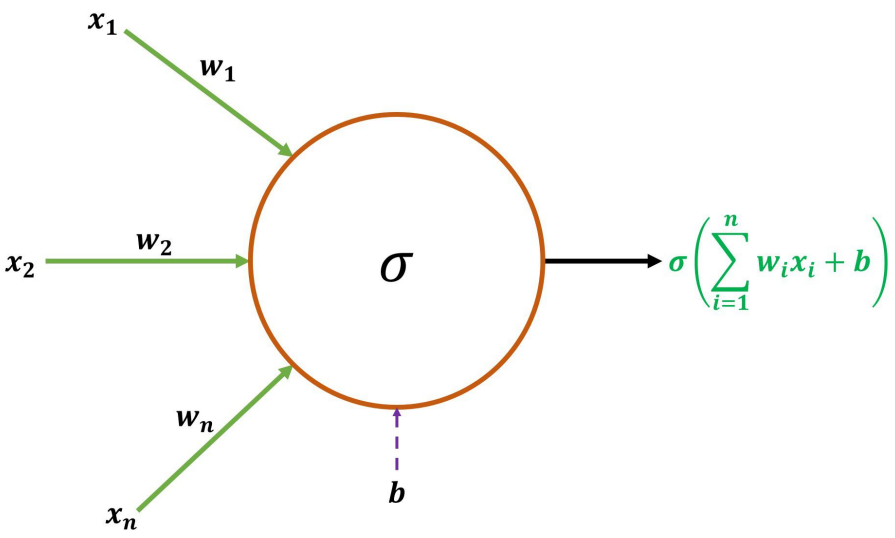

Figure 2.1: **Schematic of a perceptron**. The artificial neuron also known as perceptron colllects n inputs x_i with w_i and b parameters and processes the input using an activation function σ to provide an output.

ANNs its important to shine some light on their fundamental components. The fundamental unit that builds up the neural networks are called neurons, also known as perceptron: term coined by American psychologist Dr. Frank Rosenblatt [68]. These units have multiple inputs and one output. The output is evaluated using a transformation function (known as activation function) applied on weighted sum of all the input signals. This output is then treated as the input on to the next neuron to form a network. Connections between the neurons (known as weight parameters) also form the fundamental units of the ANN. Another fundamental component are the biases which are used for adjusting the output of the neuron. The functionality of the weights is to provide effectiveness to a particular input whereas the biases act as the intercept added to a linear equation helping to best fit the model. For a given neuron (depicted in Fig. 2.1), with n inputs x_i, weights w_i and a bias b, the output y is computed as

$$y = \sigma(\sum_{i=1}^{n} w_i x_i + b) \tag{2.37}$$

where σ represents the activation function of the given neuron. The activation functions may take any functional form such as tan hyperbolic (tanh), logistic sigmoid $\left(\frac{1}{1+e^{-x}}\right)$, rectified linear unit, softsign, arcTan and others.

To form an ANN, the neurons are typically organized in the form of layers such that the input layer consists of the same number of neurons as of the points present

in the representation vector of the input data example. The input neurons pass the values to the intermediate layers (known as hidden layers), which may contain the variable number of neurons optimized with respect to the training set. Each neuron with in the hidden layer processes the input as depicted in Eq. 2.37 to pass the output y_i (here i represnts the i^{th} neuron in the hidden layer) as the input to the next layer. After complete translation through the network, the output layer, which contains the number of neurons depending on the given problem, provides a processed output of the input dataset. For an ANN with one hidden layer, a chain of functional forms can be considered, $y = f(x) = f_2(f_1(x))$. Here, f_1 and f_2 represent the activation functions including the computations with weights and biases from hidden layer and output layer respectively. As the number of layers in the ANN increases this chain of functions increases which in turn results in a more complex and non-linear functional form. Provided sufficient number of neurons to be present in the hidden layer, the ANNs with as few as one hidden layer have been shown to be universal approximators [69].

To train and optimize an ANN, the input data set is divided into three subsets, namely training set, validation set and test set. The design of an ANN is optimized by - changing the number of neurons in the each hidden layer, changing the activation functions, and changing the number of hidden layers. The ANN geometry which results in maximum output accuracy with respect to the validation data set is chosen for further evaluation.

The training of an ANN is conducted by considering a cost function with respect to the output $O(x)$ and an expected vector $E(x)$. In the past few decades, a few cost functions have been proposed, for example:

1. Quadratic cost function: $C_{QC} = |O(x) - E(x)|^2$
2. Cross Entropy cost function: $C_{CE} = E(x)\ln(O(x)) + (1 - E(x))\ln(1 - O(x))$
3. Exponential cost function: $C_{EC} = \tau e^{\left(\frac{1}{\tau}(O(x)-E(x))^2\right)}$
4. Hellinger Cost function: $C_{HC} = \frac{1}{\sqrt{2}}(\sqrt{O(x)} - \sqrt{E(x)})^2$
5. Kullback-Leibler divergence: $C_{KL} = E(x)\log\left(\frac{E(x)}{O(x)}\right)$
6. Itakura-Saito distance: $C_{IS} = \frac{E(c)}{O(x)} - \log\left(\frac{E(x)}{O(c)}\right) - 1$

Depending on a given problem, a cost function is selected and the evaluated cost/error is fed back into the network in order to update the weights and biases by a process of back-propagation [35]. Popular algorithms namely, stochastic gradient descent, Newton's method, conjugate gradient descent, Quasi newton, Levenberg Marquardt or ADAM can be implemented for the gradient-based optimization of the stochastic cost function.

As mentioned in section 2.2, the training of an ANN can be achieved by following one of the three paradigms of supervised learning, unsupervised learning and reinforcement learning. For the supervised training, a paired set of inputs and outputs are considered and the learning is achieved by obtaining the expected output for each input. For a regression type problem, generally a quadratic cost function is considered which minimizes the mean squared error between the output of the network and the expected output. For the classification problem the cross entropy cost function is considered. The unsupervised training is conducted by providing the ANN with a training data x and and a cost function. Depending on the given problem, this cost function may take different forms. Most commonly used ANN for the unsupervised learning are the auto-encoders where the aim is to reconstruct the input data by considering a quadratic cost function for evaluating mean squared error. For RL based training, a general MDP process is considered and the ANNs serve as agents, value and state function approximators [70–72].

The structure and working of these ANNs makes them capable for extracting patterns from multidimensional datasets. For the same reason, these have been extensively employed for image recognition, speech processing, system control, game playing and decision making. ANNs have also found applications in geoscience for hydrology, ocean modelling and coastal engineering, and astronomy. For biological applications, these networks have been used to diagnose several class of carcinoma which include some major examples of lung cancer, prostate cancer and colorectal cancer. The compatibility for classification with these networks have also been harnessed to classify the blood cells based on both shape and spectroscopic data which have led to some very exciting studies of classification of different types of cancer cells and immune cells. [73–80]

2.4 Deep learning

Deep learning (DL) is a sub-field of machine learning based on artificial neural networks. The ANNs with more than one hidden layers may be considered as a part of deep learning paradigm. In fact, DL is the most recent advancement of ML which allows unbounded number of layers of bounded size which enables theoretical universal approximation and permits optimized implementation for any dataset. The layers are generally heterogeneous with different activation functions and connected in a structures fashion. DL comprises of algorithms that learn to progressively extract higher level features from the raw data.

For providing a powerful framework towards automated computations, DL has been popularly applied in all three approaches of machine learning. This section provides a brief overview of different types of deep neural networks, namely deep feedforward networks (also known as multi layered perceptrons), convolutional neural networks, generative adversarial networks, and recurrent neural networks.

2.4.1 Multi-layered perceptrons

The multi-layered perceptrons (MLPs) form the class of most fundamental neural networks and provide a conceptual mechanism for other neural networks. The aim of these type of networks is to approximate an arbitrary function f. These networks are also called feedforward networks because the information flows from the input layer with $\vec{x}$ as an input to the output layer with $\vec{y}$ as an output functional form.

These networks with multiple hidden layers transform the input data at each layer such that the final output of the network satisfies the approximation of the desired function. Mathematically, for a given ANN with an input as x, n hidden layers with $\sigma_i(\cdot)$ $i \in \{1, 2, ..., n\}$ as the functions, representing both activation function and arithmetic of weights and biases, and the output layer with σ_o as similar functional form, the output y can be evaluated as:

$$y = \sigma_o(\sigma_n(\sigma_{n-1}(...\sigma_1(x))) \tag{2.38}$$

The transformation of input vector x (shown in Eq. 2.38) is moderated by the training process (explained in section 2.3.6) which adjusts the weights and biases

for satisfying the approximation condition at the output. This training process also called the learning process is the recurring theme of DL and is followed by all the different types of neural networks.

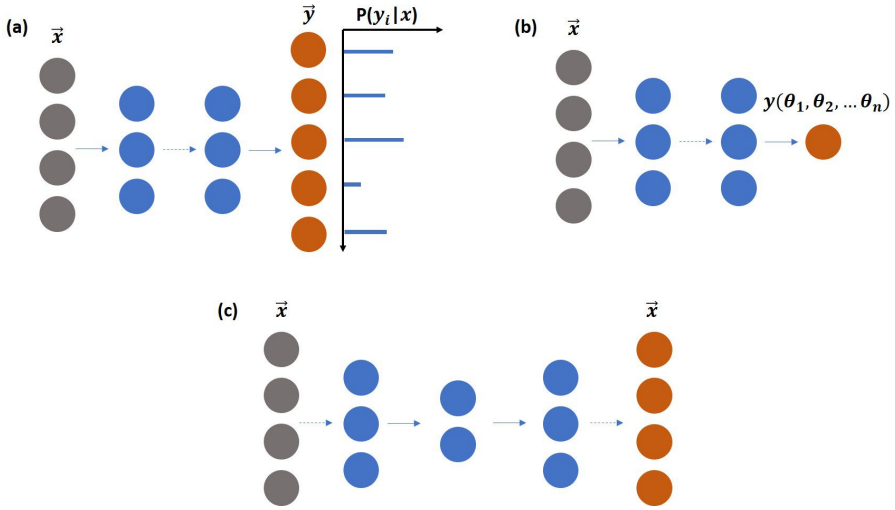

Figure 2.2: **Schematic of multilayered perceptrons for solving different problems.** The feedforward networks can be applied to solve supervised problems involving (a) classification and (b) regression. These can also be implemented to solve (c) unsupervised problems. Here dashed arrow represents the presence of multiple layers with weighted connections, whereas normal arrow represents the weighted connections.

Depending on a given problem, the MLPs can be applied in different forms to solve the supervised, unsupervised or reinforcement learning type of problems. As shown in Fig. 2.2, for different learning schemes the output layer of the MLP and the cost function is varied accordingly. For classification problems (Fig. 2.2(a)), the output layer consists of n outputs with n being the number of classes, the cost function considered is the cross entropy function. For the regression problems (Fig. 2.2 (b)), generally the output consists of one neuron which represents the functional form with respect to arbitrary parameters $\{\theta_1, \theta_2, ..., \theta_n\}$ and a mean squared error cost function is used. To solve an unsupervised learning problem (Fig. 2.2 (c)), the MLPs are designed in the form of an autoencoder. The autoencoder learns to encode the dimensionality of input vector $\vec{x}$ into a lower dimensional vector and systematically invert this reduced dimensional vector back to the input vector with the application of mean squared error cost function. The autoencoder can be applied for reducing the dimensionality of the input vectors or these can be staked and connected to another MLP for the classification or regression purposes.

For these capabilities, the feed forward networks have found several applications in photonics. They have been applied for medical image segmentation, in the implementation of optical character recognition for license plate recognition, for channel optimization in mode division multiplexing, for the classification of hyperspectral imagery, for the equalization of optical nonlinear waveform distortion, for computing optical properties of photonic crystal fibre, for identifying the similarity between reflected spectra obtained from skin lesions, for application to fluorescence lifetime imaging techniques, for particle characterization using digital inline holography, for the study of energy levels of a one-dimensional quantum particle moving in a disordered external field, and for single shot autofocusing applicable to a wide range of existing microscopes [81–91].

2.4.2 Convolutional neural networks

Convolutional neural networks (CNNs) [92] are the special form of feedforward networks which can be applied for processing any type of data which represents a grid-like topology. These networks use convolution operations in at least one layer of the network as opposed to only matrix multiplications. As explained before, MLPs use matrix multiplication which describes the transformation of each input unit into each output unit. This results in a major disadvantage for the case where the inter-input unit variation does not effect the output. Moreover, due to the large size of the parameter matrices, overall computational cost is exponentially large. Another disadvantage of MLP type networks is that each weight parameter is independent of other, and used only once while computing the output of a given layer. These problems are easily managed using sparse connections with parametric sharing. The layers exhibiting the property of parametric sharing also demonstrate the property of equivariance to translation. These properties motivate the application of convolution operation in the neural network geometry.

Formally, the convolution operation (Eq. 2.39) is defined as a mathematical operation between two functions $(x(a), w(t-a))$ such that the resultant $(y(t))$ expresses the modification in shape of first function with respect to the second.

$$y(t) = \int x(a)w(t-a)da \quad \text{s.t. } w = 0 \text{ if } (t-a) < 0 \tag{2.39}$$

For a CNN, the first argument x is called an input, w is called a kernel and y is called a feature map. For a given two-dimensional input image I and a two dimensional kernel K, the convolution operation in discreet notation can be expressed as

$$F(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \tag{2.40}$$

With respect to the commutative property of convolution, the above can also be written as

$$F(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n) \tag{2.41}$$

The equation 2.41, provides an assurance that increasing values of m increases the index for input while decreasing the index for kernel. Simply, if the size of an input image is $N \times N$ and the size of kernel is $G \times G$, the the size of output will be $(N - G + 1) \times (N - G + 1)$. The convolution operation can also be considered as matrix multiplication, however, this kind of matrix multiplication is very sparse in nature. The kernels are updated during the training process of the CNN.

Another important concept in the context of CNN is of strides and padding. Strides denote the step size of a kernel by which it shifts while convolving with the input, whereas padding can be understood as adding zero to the borders of input matrix in a symmetrical manner. Stride helps to control the degree of downsampling of the input data whereas padding allows to control the kernel width and the size of the output independently.

Any CNN consists of three fundamental units, first the convolution unit which convolves with the input to downsample the input, second a nonlinear activation function unit which modifies the linear output after the convolution process and third the pooling unit which modifies the output of layer to pool the most relevant features. The pooling unit helps to minimize the variance which may be caused due to small translations in the input data. When pooling is applied to parametric convolutions, the CNN learns to become invariant to any translated features present in the training data set, this in turn improves the statistical efficiency of the network. Depending on the underlying training dataset, different kinds of pooling can be implemented [93]. Figure 2.3 represents a schematic of a general CNN geometry

where the input two dimensional data is systematically downsampled by invoking the three fundamental units as explained.

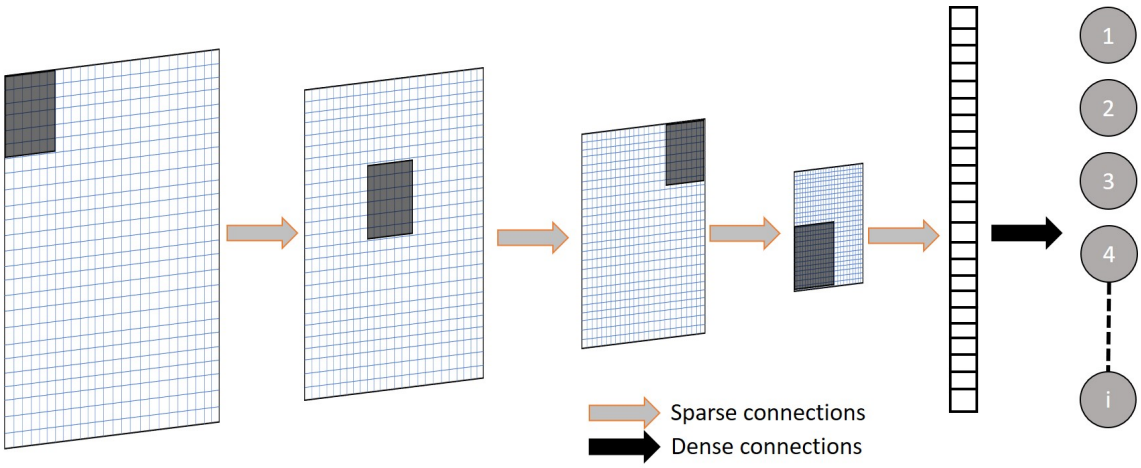

Figure 2.3: **Schematic of a general convolutional nerual network**. The input two dimensional image is systematically downsampled via the shared parameters and pooling functions. Finally the downsampled image is fully connected to the output layer. Similar to feedforward networks, these networks are also fundamentally implemented for supervised, unsupervised or reinforcement learning. Here the dark highlighted zones in individual layers represent the parametric filters.

For their excellent capabilities to be applied with supervised, unsupervised and reinforcement learning, the CNNs have found a plethora of applications in optics and photonics. They have been applied for Raman microscopy based cytopathology, for somatic mutation detection, for the identification of major cellular and acellular constituents of tissue using Fourier transform infrared spectroscopy, to classify T cell activation state using autofluorescence intensity images, for animal pose estimation, for for particle tracking in cryo-electron micrographs, to reduce speckle noise in digital holographic imaging, to reduce the speckle noise while protecting the structural information and preserving the edge features in optical coherence tomography, to improve the spatial resolution in optical microscopy, for three dimensional virtual refocusing of fluorescence microscopy images, to track optically trapped particles under noisy conditions, to solve end-to-end inverse problems in computational imaging for lensless imaging system, to image through scattering media while generalizing and maintaining high-quality object predictions, to estimate Zernike coefficients of aberrated wavefronts from a single intensity image, and for fast phase retrieval of holographic images using off-axis digital holographic microscopy [94–108].

2.4.3 Recurrent neural networks

Recurrent neural networks (RNNs) [35] are the class of ANNs which include the feedback connections allowing them to exhibit temporal dynamic behaviour. Simply, these neural networks take the input as a weighted combination of current state and the previously perceived state. For this property RNNs are generally used to process sequential data.

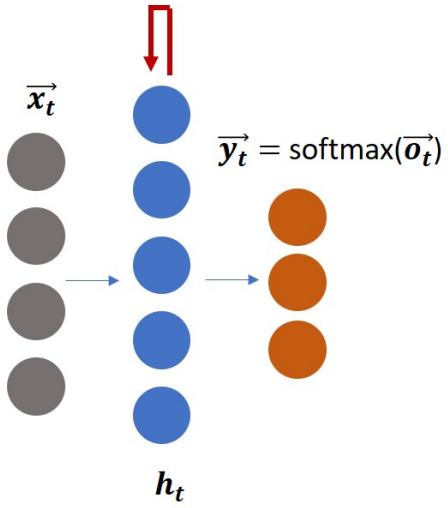

Figure 2.4: **Schematic of a general recurrent neural network**. The time dependent input x_t is processed through the hidden layer and a time dependent output y_t is evaluated. Here the backward red arrow describes the back propagation in time.

To process the sequence of discrete outputs y_t, the forward propagation of an RNN, with x_t as an input vector for a given time step (Fig. 2.4), for time steps from $t = 0$ to τ can be evaluated as,

$$a_t = b + Wh_{t-1} + Ux_t \tag{2.42a}$$

$$h_t = \sigma(a_t) \tag{2.42b}$$

$$o_t = c + Vh_t \tag{2.42c}$$

$$y_t = \text{softmax}(o_t) \tag{2.42d}$$

here, a_t is the output from hidden unit h_t which is initialized at h_0, b and c represent the biases and U, V, W represent the weight matrices. The loss function can be evaluated as the negative log-likelihood of y_t given $\{x_1, ..., x_t\}$ and can be evaluated

as,

$$L(\{x_1, ..., x_\tau\}, \{y_1, ..., y_\tau\}) = -\sum_t \log p(y_t|\{x_1, ..., x_t\}) \tag{2.43}$$

here, $p(y_t|\{x_1, ..., x_t\})$ is the reading for the entry y_t from the model output vector. The gradients for this kind of loss function are evaluated using back-propagation algorithm which is called back-propagation through time (BPTT) for this case.

The BPTT algorithms work to evaluate the gradients for each parameter. The gradient for each node is computed recursively. Starting from the nodes immediately preceding the final loss,

$$\frac{\partial L}{\partial L_t} = 1 \tag{2.44a}$$

$$(\Delta_{o_t} L)^i = \frac{\partial L}{\partial o_t^i} = y_i - 1_{i=y_t} \tag{2.44b}$$

In the above set of equations, the superscript i denotes the output class. Hence the gradient for $\Delta_{h_T} L$ can be calculated backward through time $t = T - 1$ to $t = 1$

$$\Delta_{h_\tau} L = V^T \Delta_{o_\tau} L \tag{2.45a}$$

$$\Delta_{h_T} L = W^T \text{diag}(1 - (h_{t+1})^2)(\Delta_{h_{t+1}} L) + V^T(\Delta_{o_t} L) \tag{2.45b}$$

where, $\text{diag}(1 - (h_{t+1})^2)$ indicates the diagonal matrix containing the elements $1 - (h_{t+1}^i)^2$ for the hidden unit i. With respect to the above, the gradients with respect to other parameters can be evaluated as,

$$\Delta_c L = \sum_t \left(\frac{\partial o_t}{\partial c}\right)^T \Delta_{o_t} L = \sum_t \Delta_{o_t} L \tag{2.46a}$$

$$\Delta_b L = \sum_t \text{diag}(1 - (h_t)^2)\Delta_{h_t} L \tag{2.46b}$$

$$\Delta_V L = \sum_t (\Delta_{o_t} L) h_t^T \tag{2.46c}$$

$$\Delta_W L = \sum_t \text{diag}(1 - (h_t)^2)(\Delta_{h_t} L) h_{t-1}^T \tag{2.46d}$$

$$\Delta_U L = \sum_t \text{diag}(1 - (h_t)^2)(\Delta_{h_t} L) x_t^T \tag{2.46e}$$

Truncated BPTT [109], an approximation variant of BPTT, was proposed for application to long sequences. This variation of BPTT saved massive computation

cost for the updating each parameter.

The RNNs seek to establish connections between final output and all the previous events. This makes it very difficult to identify on which event provides an important feedback with respect to updating the gradient. Derivatives with respect to each parameters, especially for deep neural networks, are susceptible to vanishing or exploding. The exploding gradients problems refers to when the gradient update for some of the parameters becomes massive or sometimes saturated. Vanishing gradients problem, on the other hand, refers to the gradients update being very small to compute. The problem of exploding gradients can easily be solved by truncating the gradients, however, the problem of vanishing gradients is very difficult to determine.

To solve the problem of vanishing gradients a variation of RNNs namely Long Short-Term Memory (LSTM) [110] units were proposed. LSTMs are the type of gated RNNs which maintain a constant error allowing the learning process to continue for more than 1000 steps. These units store information in the form of gated cells which are implemented on an element wise manner with a multiplication by sigmoids. These cells learn to allow the flow of data through the iterative back-propagating error and adjusting weights using gradient descent. A detailed review about various variants of LSTM can be found in [111].

LSTMs and other gated RNNs [112] provided workable solution to the vanishing gradient problem, however, they failed to provide very precise information about early temporal states. To this rescue, recently a Transformer neural network [113] model was introduced. These networks were based on attention mechanisms which allow the model to process any temporal state in the sequence. The attention layer, works by learning the measure of relevancy in terms of parametric weights for a given temporal state. Adding these attention layers to ANNs led to massive gain in performance which has lad to several applications [114–117].

For their abilities to learn from the temporal sequences, RNNs have found various applications in optics and photonics. They have been applied for particle tracking of virus structures in fluorescence microscopy image sequences, for microscopic cell segmentation to exploit cell dynamics, to delineate skin strata in stacks of reflectance confocal microscopy images collected at consecutive depths, for three dimensional

image segmentation while leveraging three dimensional image anisotropism, to efficiently characterize anomalous diffusion using a single short trajectory, to extract spatio-temporal features from an echocardiography video to classify the frames as cardiac events, and for diagnosing disease using spatio-temporal cellular dynamics in compact digital holographic microscopy [118–124].

2.4.4 Generative adversarial networks

Generative adversarial networks (GANs) [125] belong to the family of deep generative models. GANs are based on game theory where a generative neural network is competed against an adversary network, also known as discriminator network. The motivation behind the application of GANs is that the learning process does not require approximate inference or approximation of partition function gradient.

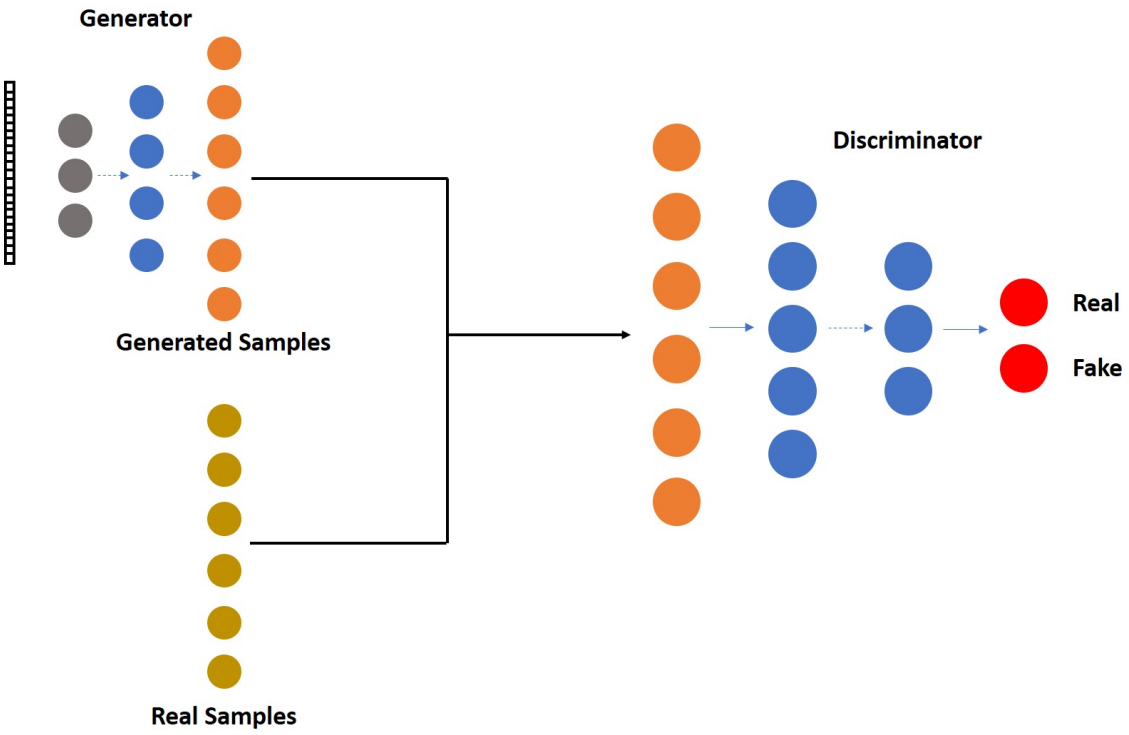

Figure 2.5: **Schematic of a generative adversarial network**. Generator network produces an output sample from a prior noise distribution and the discriminator network outputs a probability to distinguish between the generated and real sample.

A GAN consists of two networks, the generator network which produces a sample $x = g(z)$ given a prior noise distribution $p(z)$, and the discriminator network which tries to distinguish between samples drawn from the training data and the generated

data. The discriminator network outputs a probability value $d(x)$ which indicates if x belongs to the training data or generated data (Fig. 2.5). The discriminator network is trained to maximize the probability of assigning the correct label to both the classes of training data and generated data whereas the generator network is trained to minimize $\log(1-d(g(z)))$. The training can be thought of as a two-player min-max game with a cost function $C(g,d)$,

$$\min_g \max_d C(g,d) = \mathbb{E}_{x\sim p_{data}(x)} \log(d(x)) + \mathbb{E}_{z\sim p_z(z)} \log(1-d(g(z))) \tag{2.47}$$

In practice, the training of a GAN is implemented in an iterative manner. For each iteration, first the discriminator network is updated (Eq. 2.48a) and then the generator network is updated (Eq. 2.48b).

$$U_d = \Delta_{w_d} \frac{1}{m} \sum_{i=1}^{m} [\log(d(x_i)) + \log(1-d(g(z_i)))] \tag{2.48a}$$

$$U_g = \Delta_{w_g} \frac{1}{m} \sum_{i=1}^{m} \log(1-d(g(z_i))) \tag{2.48b}$$

here, w_g and w_d are the parameters of the generator and discriminator networks respectively, m denotes the number of training datapoints for a mini-batch training. The training of GAN can also understood as a zero-sum game with the cost function being the payoff to both the networks. After the successful training, the generator data become identical to training data and hence the discriminator network provides an output probability of $\frac{1}{2}$ for all the possibilities. The cases where $\max_d C(g,d)$ is not convex, the training of a GAN may prove to be difficult and lead to underfitting of data.

An interesting capability of a GAN is that rather than learning the specific distribution, the generator network can learn the transformation between a random noisy distribution to the training data. The extraordinary capabilities of GANs have made them a popular topic for research in machine learning. M. Mirza et al. proposed conditional GANs [126] which modified the cost function to have conditional output with respect to an extra auxiliary information (y) from the generator

and discriminator networks. The cost function was modified as:

$$\min_{g}\max_{d} C(g,d) = \mathbb{E}_{x\sim p_{data}(x)} \log(d(x|y)) + \mathbb{E}_{z\sim p_z(z)} \log(1 - d(g(z|y))) \tag{2.49}$$

In another study, A. Radford et al. proposed deep convolutional generative adversarial network (DCGAN) [127] which allowed for the generation of images directly from the random noise. X. Chen proposed InfoGAN [128] to learn disentangled representations in completely unsupervised manner. H. Zhang et al. proposed StackGAN [129] which generated images based from text descriptions and also proposed a conditioning augmentation technique to improve and stabilize the training process of conditional GAN. To improve the stability of learning and reduce the problems of mode collapse, M. Arjovsky et al. proposed Wasserstein GAN [130] for which the loss function was modified to include the Wasserstein distance as a metric. DiscoGAN [131], was proposed by T. Kim et al., which learned to discover relations between different domains; they showed that after learning these domains the network could transfer styles from one domain to another while preserving the key attributes. Another variant, namely cycle GAN [132] was proposed by J. Zhu et al. They demonstrated that by using adversarial training for learning the inverse mapping in a cycle consistent manner results in an excellent image mapping.

For their excellent learning capabilities, the GANs have recently been very popular in the research of optics and photonics. They have found applications, to enhance resolution in scanning electron microscopy, for cross modality super-resolution in fluorescence microscopy, for accelerating the frame rate and reducing independent localizations for super-resolution microscopy, for segmenting bright field microscopic images of cells and X-ray computed tomography of metallic nanowire meshes, to achieve high resolution images similar to confocal scans with 64 fold improvement in imaging speed, to deconvolve and improve the quality of blurred and noisy deep depth fluorescence microscopy images, to learn the mapping between a partial image and a full image for reconstructing full images from their much smaller subareas, to generate volumetric holographic images of using bright-field microscopy contrast, to use unpaired images for coherent noise reduction in optical diffraction tomography, to generate synthetic ultrasound images with crisp speckle patterns without the knowledge of underlying physical model, to effectively reduce speckle noise in optical

coherence tomography images and videos while maintaining spatial and temporal resolutions, and to produce high-efficiency, topologically complex diffraction free devices with a broad operation range of deflection angles and wavelengths [133–144].

2.5 Conclusion

In conclusion this chapter laid the foundation for this thesis by providing an introduction to the fundamentals of Machine Learning.

Different approaches of ML, including supervised learning, unsupervised learning and reinforcement learning are discussed in the context of different types of fundamental problems. The discussion for supervised learning has been explained with respect to the fundamental problems of regression and classification. Unsupervised learning is discussed with respect to the problems concerning density estimation, clustering and dimensionality reduction. Finally, the fundamental of reinforcement learning are discussed where a brief overview of finite markov processes, dynamic programming, Mote Carlo methods and temporal difference learning are provided.

The next section provides a mathematical insight into the popular algorithms of ML. Additionally, the applications of these techniques in the context of optics and photonics are also discussed. Beginning with an overview of eigen analysis based principal component analysis, this section spans the fundamentals of linear discriminant analysis, k-means algorithm, support vector machines, t-distributed stochastic neighbor embedding and artificial neural networks.

Further, this chapter explains the working and fundamental of Deep Learning, the most recent advancement of ML. Mathematical description for the working of various deep learning based neural networks is provided and their recent applications in the field of optics and photonics are discussed. The feedforward neural networks are discussed which encompasses their ability to form chain functional form and applicability to different approaches of ML. Finally, a few of their applications are discussed. Following the consideration of feed forward networks, convolutional neural networks are explained where a heuristic description of the convolution operation, the function of strides, padding and pooling layers is given; this is followed by a discussion of their applications in various field of optics including microscopy,

holography, spectroscopy, optical trapping and wavefront shaping. Subsequently, recurrent neural networks are examined with a brief overview of forward propagation and back-propagation in time is explored; finally their applicability is discussed. The successive discussion has been dedicated to generative adversarial networks which includes a brief description of their training process. After this different variants of GAN are discussed and finally their applicability is discussed in the field of optics.

The information in this chapter is intended to provide a foundation of ML based approaches implemented throughout the thesis. The specific applicability and reasoning with respect to a given study shall be detailed in the respective chapters.

Chapter 3

Label-free characterization of the immune cells enhanced by machine learning

Parts of this chapter are adapted from the article "Label – free optical hemogram of granulocytes enhanced by artificial neural networks" [145], published in Optics Express (2019).

3.1 Introduction

Identifying and quantifying the various types of cells of the immune system in the blood stream is an important process in both the clinical and lab environment. Alterations in both number and type can indicate not only infection and inflammation, but the presence of abnormal cells associated with blood cancers such as leukemia and lymphoma. Currently, attaching fluorescent-tagged antibodies to the cells for analysis on a flow cytometer, or simply manual microscopic analysis of a blood smear on a glass slide are routinely used techniques. Both suffer from either being time consuming or requiring expensive antibody reagents. The aim is to develop a novel, and completely label-free method to characterize cells of the immune system which may lead to rapid, inexpensive, reagent-free laser light based system for use in the clinic and laboratory from very small blood or cell samples.

This chapter provides a brief introduction to the fundamentals of Immunology

and label-free optical techniques, namely, Raman spectroscopy and digital holographic microscopy (DHM). Following which two studies are presented where the application of machine learning in combination with optical techniques is demonstrated to enhance the characterization of human immune cells in a label-free manner.

3.1.1 Immunology

Immunology is the subject in biology to study the physiological mechanisms that humans and other animals use to protect their body from invasion by different types of microbes, toxins and environmental extremities. The discipline grew out of the observation that the people who survived the wreckage from the epidemic diseases were immune to the disease when infected again. Infections are caused by the presence of large population of microorganisms inside the humans/animals. The infected body responses to the infection by producing the cells dedicated to defend themselves, the collection of these defensive cells form the immune system of the body [146].

The immune system provides protection against pathogens, including bacteria and viruses for the human body. Without the presence of a functioning immune system, even the most minor of the infections can damage the body severely. However, despite having a fully functioning immune system, humans suffer from infections. This happens because of the time taken by the immune system to build up the strongest response to an invading microorganism. During the process of the battle between the immune system and infection, the number and types of immune cells change hence providing important information to the modern medicine about the type and level of attack.

Immunity is defined as the resistance exhibited by the host towards injury caused by the microbes and their products. To defend the host body from the any foreign exposures or pathogens, the immune system has three lines of defense that operate at different timelines with respect to the progress of infection. The first line of defense is the innate immunity, also known as the invariable or hereditary response, which is always ready for action and functions from the very beginning of an infection. The second line of defense is the induced response to the infection, activated when the

immune system detects the presence of infection and turn on the gene expression and protein synthesis necessary to make the response. The third line of defense is the adaptive immunity, the resistance that an organism acquires during the lifetime, it is capable of recognizing and selectively eliminating the infectious agents.

The first and second line of immune responses are affiliated to the innate immune system. The first line of defense comprises of the skin and the mucosal epithelia that lines the digestive, respiratory and urogenital tracts. These tissues provide effective physical and chemical barriers that prevent pathogens from gaining access to the internal tissues and organs. The low pH and presence of fatty acid makes the environment inhospitable for pathogens. The mucous membrane form a less formidable barrier. The host body is protected from the pathogens as they are trapped into the mucus which is swept away by the cilia of the respiratory mucosa or the villi in the intestine; particles are swallowed and coughed out by cough reflex. This flushing effect in the body reduces the chances of bacterial growth.

The defense mechanism of the innate immune systems is activated as soon as the pathogens pass though the epithelial walls. The first weapon that is fired by the system is a combination of soluble proteins made constitutively by the liver and present in the blood, lymph, and extracellular fluids. These plasma proteins are collectively known as the complement system. Any infection triggers the complement activation which proceeds by a series of highly specific enzyme reactions, in which each protease cleaves and activates the next protease in the pathway. These enzymes then attack the pathogens via various pathways to destroy their growth and their presence from the host body.

When the first line of defense fails due to congenital or acquired defects, then the deeper tissues are at risk and next line of defense is activated. This phase of the innate immune system involves soluble and cellular receptors that detect the presence of infecting organisms and then recruit leukocytes to fight the invasion. These genetically programmed receptors provide the initial level of defense by recognizing their structural features of microbes that are not found in the host. The engagement of these pattern recognizing receptors lead to the activation of host cells and their secretion of antimicrobial substances. The receptors are expressed by macrophages, natural killer (NK) cells and other innate immune system cells. These cells of the

immune system are responsible for the phagocytosis. The macrophages resident in the infected region are responsible for orchestrating the induced innate response to the infection, they are responsible for the secretion of inflammatory cytokines and chemokines that actively recruit neutrophils, monocytes, and NK cells. Neutrophils are the dedicated phagocytes and the first effector cells recruited to the sites of infection, they are recruited by the inflammatory cytokines and are the potent killers of pathogens. Neutrophils are themselves programmed to die after the process is complete. NK cells are the main circulating lymphocytes that contribute to the innate immune response with two of its populations in the blood and tissues. NK cell cytotoxicity is activated at sites of virus infection.

Adaptive immunity is the type of immune response which a host body acquires during its lifetime. Once the system is developed for an infection, it is long lasting and there is no lag or latent time for response. Body responses to the infection promptly and powerfully. This kind of immune response clears the body of extracellular pathogens and their toxins by means of antibodies, the secreted form of the B-cell receptor for antigen. These antibodies are produced by the effector B lymphocytes, or plasma cells, of the immune system in response to infection. They circulate as a major component of the plasma in blood and lymph and are always present at the mucosal surfaces. Antibodies are very variable proteins which can recognize all types of biological macromolecules. They are the best source of protective immunity and the most successful vaccines protect through stimulating the production of high quality antibodies.

To understand the working of immune system, it is important to shine some light on the cells and tissues of the immune system. The complete structure containing the cells and tissues arranged in an organized manner for an effective functionality is called as the lymphoid system. The cells of the immune system are principally the white blood cells (WBCs) also known as leukocytes. These cells are continuously generated by the body by the process called hematopoiesis. As is depicted in Fig. 3.1, the leukocytes derive from a common progenitor called pluripotent hematopoietic stem cell, which is also the source of red blood cells (RBCs) – known as erythrocytes, and megakaryocytes – the source of platelets. All the cell types together are called the hematopoietic cells.

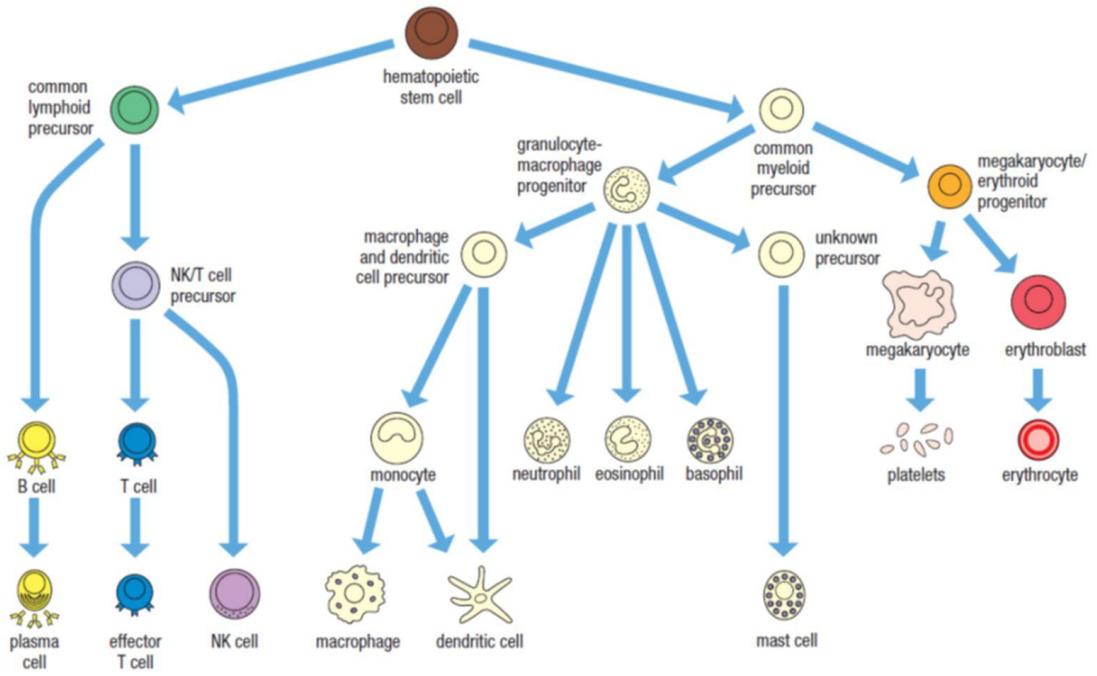

Figure 3.1: **Schematic of origination of blood cells from a common hematopoietic stem cell**. The pluripotent stem cell divides into lymphoid and myeloid lineages of blood cells. These can then further be divided into B cells, T cells, NK cells, macrophage cells, dentritic cells and mast cells. Adopted from [3]

The hematopoietic progenitor gives rise to the myeloid and lymphoid precursors which in turn provide origin to their individual lineages. The myeloid lineage consists of two groups, one of them being granulocytes and the other being megakaryocyte. The granulocytes, having the prominent cytoplasmic granules, contain reactive substances which kill microbes and enhance inflammation. Major cells of this group are neutrophils which specialize in engulfing, capturing and killing of microbes; eosinophils which specialize in defending against helminth worms and other intestinal parasites; and basophils which are least abundant but they also respond to the parasites. The other group of myeloid lineage consists of monocytes, macrophages and dendritic cells. These cells also get involved in immune response of the body and are well equipped for phagocytosis. Neutrophils are programmed to be short lived whereas the macrophages are long lived and provide the warning to other cells and orchestrate the local response to the infection. The lymphoid lineage gives rise to two morphologically distinguished lymphocytes being NK/T cells and B cells. Natural Killer cells are the large granular lymphocytic effector cells of the innate immunity, are very important during the viral infections. B Cells have typically 10^5

molecules of membrane bound antibody per cell, which exhibit the same antigenic specificity and interact directly with antigens. T cells bear $CD4^+$ or $CD8^+$ proteins which are involved in the recognition of class I and class II MHC molecules. These small lymphocytes are responsible for recognizing the pathogens and start the process of selection, growth, and differentiation to induce a powerful adaptive immune response.

The host body generates different types of blood cells to fight against the pathogens. The density and level of different types varies as per the degree of infection that the host is facing. This makes it medically relevant to study, identify and distinguish the subsets in blood for the implication in both diagnosis and treatment. Currently, this is achieved using a variety of techniques; Flow cytometry – often referred to as Fluorescence activated cell sorting (FACS), Immunofluorescence (IF), and chemical-based hemogram being a part of everyday routine in every hospital to assist medical diagnosis [147]. FACS is a specialized type of flow cytometry which provides the method for sorting a heterogeneous mixture of biological cells [148]. It helps to sort large number of cells with a high rate. IF is a fluorescence microscopy which uses the specificity of antibodies to their antigens to target specific biomolecule targets within a cell. It allows visualization of the target molecule through the sample, compared to FACS, it is a slower sorting technique with about 100 cells per second [149]. The routine hemogram records number of the different corpuscular components, which can also give an information on size, morphology, and granularity [147].

3.1.2 Label-free optical techniques for cell identification

The techniques explained above are very useful and quite efficient to study the number and type of cells, but they often result in death of the cells or potential alteration in cellular behavior. For example, many of CD antigens recognized by antibodies are cell surface proteins involved in cell activation, recognition etc., thus binding antibodies to these molecules runs the risk of activating the cells, which may alter their behaviour in subsequent experimental studies. In addition, these tests can be cumbersome, require significant user intervention and require large number of cells for differentiation. These disadvantages call for an important development of a technique which can be non-invasive, label-free and passive. Optical spectroscopic

and microscopic profiling may help to monitor the differentiation process while preserving cell integrity and functionality with requirement of fewer cells [150].

Vibrational Spectroscopy in combination with digital holography provide to be the non-invasive and label-free viable alternative to differentiate between the cellular structures which provide the spectral and morphological signature with high selectivity and specificity. Raman spectroscopy is the type of vibrational spectroscopy which has been widely employed to study and discriminate various types of immune system cells. It can provide molecular information by inelastic scattering of light that is detected as a spectrum which in turn provides a chemical signature of the sample. It has also been applied in combination with optical tweezers and light sheet experiments for various applications [151–154].

Raman Spectroscopy

Raman Spectroscopy is a spectroscopic technique used to observe vibrational, rotational and other low-frequency modes in a system [155]. It is based on inelastic scattering of monochromatic light. When the light from laser beam interacts with a material, the photons are absorbed and re-emitted with a frequency above or below the original photon frequency. This effect is called the *Raman effect* [156].

The incident light is absorbed by the molecule to get excited to a virtual energy state and then it relaxes to come back to the ground state with three possibilities:

- The molecule ejects the photon with the same energy as that of the original photon. This kind of interaction is known as **Rayleigh Scattering**.

- The molecule returns to its ground state with part of the original photon energy being transferred to Raman active mode to give off the photon with a frequency less than the original frequency. This kind of interaction is known as **Stokes Scattering**.

- The third possibility is when the Raman-active molecule, which is already in the excited state absorbs the photonic energy and then returns to the basic ground state to give off the photon with a frequency of more than the original. This kind of scattering is called **Anti-Stokes Scattering**.

An illustration of the above explained scattering phenomenon is depicted in Figure 3.2, showing the vibrational ground states and virtual energy states of a given sample.

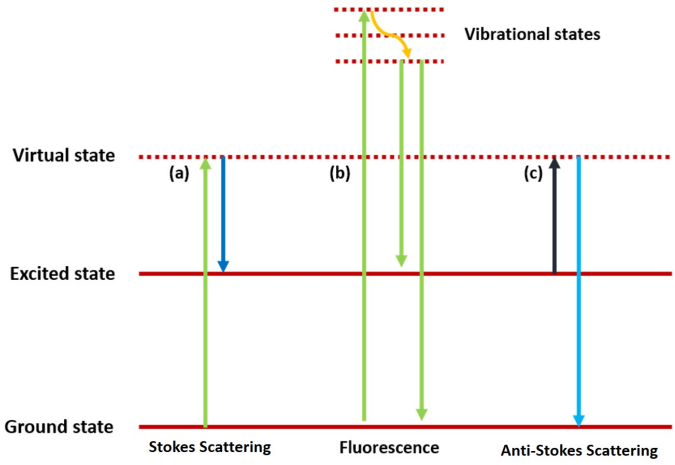

Figure 3.2: **Possibilities of light scattering.** (a) Stokes scattering, (b) Fluorescence (c) Anti-Stokes scattering

The instrumentation of a typical Raman system consists of two components; the illumination unit and the collection unit. As shown in Fig. 3.3, the laser light illuminates the sample with an incident wavelength, then the fluorescence from the sample is collected by the objective and this beam is passed through various filters to the spectrometer connected to a CCD camera to finally obtain the Raman spectra.

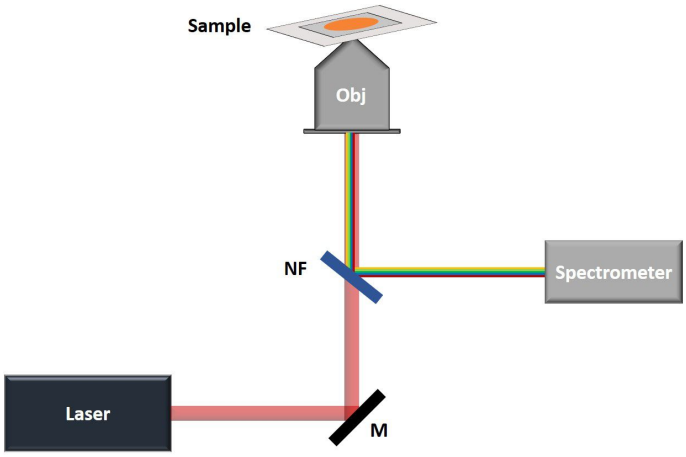

Figure 3.3: **Schematic of a Raman spectrometer.** Laser light illuminates the sample with an incident wavelength then the fluorescence is collected by the objective and is passed onto the spectrometer. Here, NF means notch filter, M means mirror and Obj means objective.

Raman shift is calculated by the following equation:

$$\bar{\nu} = \frac{1}{\lambda_{in}} - \frac{1}{\lambda_{sc}} \tag{3.1}$$

Here $\bar{\nu}$ is the Raman shift and λ_{in} is the incident laser beam wavelength and λ_{sc} is the scattered wavelength detected by the spectrometer.

Wavelength Modulated Raman Spectroscopy

Compared to the intense background fluorescence and stray light, the Raman signal is very weak and it makes the detection of noise free Raman signal to be very difficult. To overcome this problem, various optical notch filters are employed to cutoff the most prominent stray light present in the spectral range close to the incident laser beam wavelength. For practical biomedical applications, the fluorescence background should be removed to improve the signal to noise ratio, selectivity, and specificity. Various experimental methods have been employed to improve the raw data and extract the Raman signal from the spectra. Some of the techniques are time resolved Raman spectroscopy, polarization modulation technique, use of annular excitation beam profile, phase sensitive detection scheme, [157–162] and many others. One of the most innovative and efficient ways to detect and improve the signal to noise ratio was Wavelength modulated Raman Spectroscopy (WMRS) developed by Dholakia group. This technique is based on the principal that the fluorescence background does not change whereas the Raman scattering is shifted by the change in excitation wavelength as can be understood by the Raman equation stated before. [163]

Figure 3.4 illustrates the improvement in signal to noise ratio of the Raman spectrum accumulated for 1 μm polystyrene beads. The fluorescence background is completely removed for Fig. 3.4 (b) for the sample under identical environmental conditions. The excitation laser beam had a central wavelength of 785 nm with a power of 120 mW on the sample plane. For WMRS, the wavelength was modulated between 784.5 nm and 785.5 nm with a variation of 0.1 nm and each spectrum was accumulated with an exposure time of 1 second.

Since its onset, WMRS has found several applications in the label free identification of subsets of the cells of the immune system to identify T lymphocyte subsets, Natural Killer cells and Dendritic cells [164]. This system has also been to identify

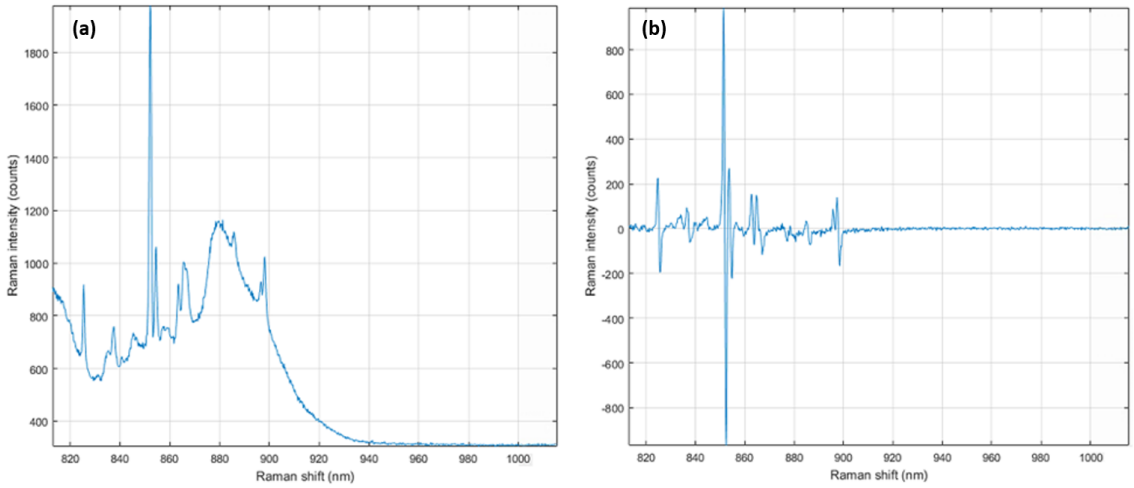

Figure 3.4: **Comparison between Raman spectroscopy and WMRS** (a) Raman Signal for a 1 μm polystyrene bead; (b) WMRS signal for the same. The Raman signal give an intrinsic biochemical information stored in the beads. Raman spectrum represents the signature as the peaks signify specific vibrational frequency exhibited by these 1 μm beads.

and detection of pharmaceuticals with the combination of paper microfluidics [165] and has been optimized for high throughput cell screening [166].

3.1.3 Digital holographic microscopy

Raman spectroscopy displays a huge potential for discriminating the biological samples, however, this technique lacks the aspect of throughput rate. Hence to improve on this aspect and generate more accurate and specific ways to discriminate between different cell types, another dimension of microscopy can be considered. In addition to the specific chemical information stored in each cell, they also exhibit morphological differences which play an important role in sorting cell types. The state of the art digital holographic microscopy technique makes it possible to acquire the morphological information stored in the cellular structures. It would be rather important to understand the working of a digital holographic microscope and how the data can be analyzed for the same.

Digital holographic microscopy (DHM) is ultimately digital holography applied to microscopy. Compared to the conventional microscopy methods in which the projected image is caught, DHM captures the light wavefront as a digital hologram. It provides quantitative information on the phase shifts induced by the sample which can therefore be used to determine the morphological information for identification

and diagnosis of diseases. [167–169]

The working technique of DHM is based on simple hologram generation. To create a hologram, the sample must be illuminated by a coherent monochromatic laser source. As shown in Fig. 3.5, this source is split into two arms, an object arm and a reference arm. The object beam is collected though the microscopic objective and is combined with the reference arm to create a holographic interference pattern. This pattern is produced into a view-able image of the object using widely available numerical reconstruction algorithms.

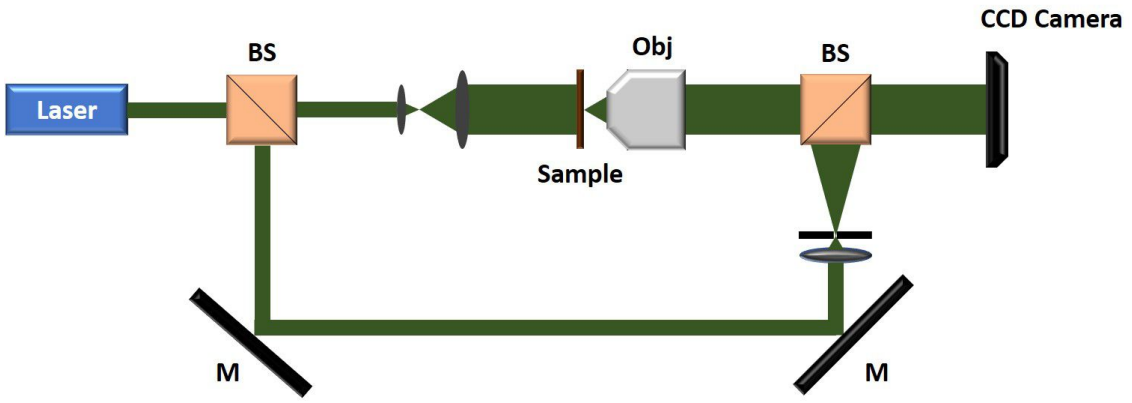

Figure 3.5: **Schematic of an off-axis digital holographic microscopy system.** The laser beam is split into object and reference arm by a Beam splitter (BS) and then the object beam is collected by objective lens (Obj) to combine with reference beam at the second beam splitter, the combined interference pattern is collected by a camera. Here M are mirrors.

The combination of spectroscopic and microscopic techniques provides a multi-modal approach for detecting the molecular and morphological properties of the biological samples. Major interest has been displayed in the field of biophotonics to incorporate the multi-modal systems. These systems serve as a very useful tool to overcome the limitations displayed a single technique and give a more useful and complete information about the sample. The combination of Raman with OCT, and Raman with DHM have been studied in the past to characterize the tissues or cancer cells and to discriminate different cells of the immune system, where the chemical information was studied using Raman and structural and morphological information was studied using the OCT and DHM systems [170, 171]. A multi-modal system (implemented in [171]) combining Raman spectroscopy and DHM is illustrated in Fig. 3.6. The common sample is illuminated by a bright green

Spectra Millennia Vs laser source (∼532 nm) for DHM; and a 785 nm Solstis laser beam with tunable wavelength was illuminated on the sample for taking the Raman spectra. For the DHM setup, the laser beam was passed through a beam splitter, out of which one of the arms were the imaging arm passed through a long working distance 10X objective lens and other was the reference arm for the generating the interference pattern. The two beams were combined using a cube beam splitter to achieve an off-axis hologram on a CMOS detector. For the Raman setup, sample was illuminated using a 60X objective and the signal was collected by a spectrometer. The mentioned objective was considered for both Raman and DHM system to achive an optimal optical resolution (lateral: 0.31 μm; axial: 1.47 μm) for analysing the immune cells.

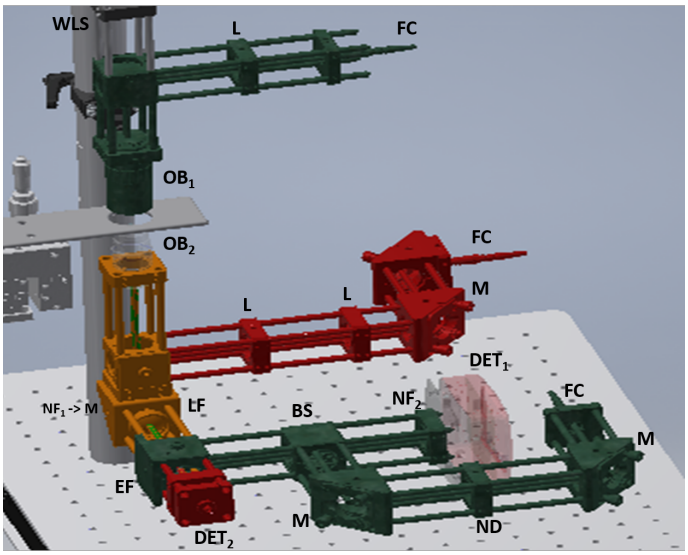

Figure 3.6: **Multimodal Raman and DHM setup.** (referred from [171]) Here OB_i is objective lens, FC are the fiber coupling, L are the Lenses, NF_i are the notch filters, M are the mirrors, BS_i are the beam splitters, ND is the neutral density filter, DET_i are the detectors and WLS is the white light source for microscope.

Mathematical description of phase reconstruction

To get a numerical understanding of the phase map reconstruction using DHM images, it would be useful to understand the mathematical description of this problem. As shown in Fig. 3.5, the object wave is reconstructed by illuminating the hologram with the reference beam. The reconstructed image exhibits all the information about the variations in morphology and refractive index that may be present in the sample. The object wave can be described as:

$$O(x,y) = o(x,y)e^{i\Phi_O(x,y)} \tag{3.2}$$

Here, $O(x,y)$ is the complex amplitude of the object wave described by the real amplitude $o(x,y)$ and the complex phase $\Phi_O(x,y)$. Similarly, the reference wave can be described as:

$$R(x,y) = r(x,y)e^{i\Phi_R(x,y)} \tag{3.3}$$

$R(x,y)$ describes the complex amplitude of the reference wave, described by the real amplitude $r(x,y)$ and complex phase $\Phi_R(x,y)$. After interference, the intensity detected by the camera is given by:

$$\begin{aligned} I(x,y) &= |O(x,y) + R(x,y)|^2 \\ &= O(x, y + R(x,y))(O(x,y) + R(x,y))^* \\ &= R(x,y)R^*(x,y) + O(x,y)R^*(x,y) + O^*(x,y)R(x,y) + O(x,y)O^*(x,y) \end{aligned} \tag{3.4}$$

where $*$ denotes the complex conjugate. Considering the background noise and exposure time of the camera, final detection amplitude can be written as:

$$D(x,y) = d_{back}(x,y) + \beta\tau I(x,y) \tag{3.5}$$

Here, d_{back} is the background shot noise generated due to the lack quantum efficiency of any general detector, β is the proportionality constant and τ is the exposure time.

For numerical reconstruction, the detection complex amplitude has to be multiplied with the complex amplitude of reference wave giving:

$$R(x,y)D(x,y) = [d_{back} + \beta\tau(r^2 + o^2)]R(x,y) + \beta\tau r^2 O(x,y) + \beta\tau R^2(x,y)O^*(x,y) \tag{3.6}$$

From equation 3.6 the real part of image can be easily reconstructed by considering the zero order non-diffracted wave. The imaginary part of the image consists

information about the changes in optical path due to the deformations in sample.

Computationally, this is achieved by using fast Fourier transformation (FFT) technique and inverse Fourier transform. As shown in Fig. 3.7, zeroth and first order of the Fourier space image are selected and are multiplied element-wise to form the amplitude and wrapped phase images. Finally, the phase is unwrapped using a 2D phase unwrapping algorithm [172,173]. The calculated phase and amplitude images can be used for 3D reconstruction of any holographic image.

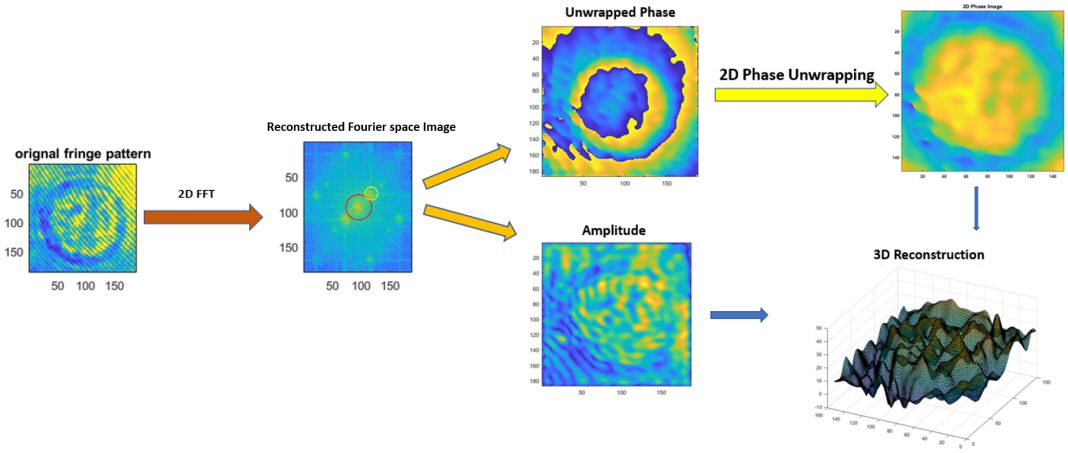

Figure 3.7: **Image reconstruction using DHM**. The highlighted region in reconstructed Fourier image represents the zeroth and first order of the initial interference pattern, as accumulated by the CCD camera. Here, red Circle denotes zeroth order or amplitude of the image, whereas yellow circle denoted the first order or complex phase.

Technique	Advantage	Disadvantage
Raman Spectroscopy	1. Label-Free 2. Non-invasive 3. Non-destructive 4. Chemical information	1. No morphological information 2. Hampered by long acquisition time
DHM	1. Label-free 2. Non-invasive 3. Non-destructive 4. Rapid acquisition time 5. Morphological information	No chemical information

Table 3.1: Table to summarize the advantages and disadvantages of label-free optical techniques which are used to discriminate and study the different cells.

3.2 Classification of neutrophils and eosinophils

3.2.1 Methods

Cell Isolation

In order to perform the cell isolations on whole human blood, an ethical permission was acquired from the School of Medicine at the University of St. Andrews. After obtaining the informed written consent from the three healthy donors, the isolations were performed on the whole blood samples. These samples were collected using heparinised tubes for the quantities of 20-30 ml per donor. Eosinophils were isolated by following a two step mechanism. For the first step, whole blood was separated over Histopaque (Sigma, Poole UK). As the second step, the untouched eosinophils were isolated using a MACsxpress eosinophil isolation kit (Mitenyi Biotec, UK, cat:130-092-010) by following manufacturer's instructions. Neutrophils were isolated directly from the whole blood by using MACSxpress neutrophil isolation kit (Miltenyi Biotec, UK, cat: 130-104-434) following manufacturer's instructions. The isolated neutrophil solution also contained a small amount of erythrocytes. These small number of erythrocytes were removed using an erythrocyte lysis kit (Milteny Biotec, UK cat: 130-094-183) following manufacturer's instructions. The isolated neutrophils were cultured in the plasma from which they were isolated in the kit whereas the eosinophils were cultured in RPMI supplemented with 5 % Foetal Bovine Serum (FBS, both Invitrogen, UK).

To chemically confirm and benchmark the isolation of these cells, flow cytometric analysis was conducted. Purified populations of both cell types were washed into Phosphate Buffer Saline (PBS) with 0.5 % FBS and blocked with 10 % human plasma for 10 mins on ice prior to labelling with antibodies. Anti-CD3-FITC (eBioscience, clone OKT3, cat: 11-0037-41) was used as a negative control for both cell types. Purified neutrophils were stained with anti-CD15-FITC (eBioscience, clone H198, cat: 11-0159-41), and eosinophils with anti-CD66b-FITC (eBioscience, clone G10FS, cat: 11-0666-42). Flow cytometry was performed on a Guava 8HT (Merck-Millipore, UK).

In succession to flow cytometry, the isolated cells were prepared for optical analysis. First these cells were centrifugated at 1200 rpm for 5 minuntes and then

resuspended in PBS with 0.5% FBS solution to avoid aggregation. Cell suspension with a volume of 20 μl was prepared and transferred to the center of a quartz slide (25.4 mm x 25.4 mm x 1 mm). This slide had a chamber which was formed by the use of an 80 μm thick vinyl spacer. The whole chamber was covered from top by another thin quartz slide (25.4 mm x 25.4 mm, 0.11 mm - 0.15 mm thickness). The whole assembly was inverted and left for ∼30 minutes such that the cells could settle down and no cell motion would be present during measurement.

Raman data acquisition, processing and classification

Single cell WMR spectra with a resolution of 2.4 cm^{-1} were collected by placing the prepared sample on a confocal Raman system [174, 175] which was pumped by tuneable Ti:Sapphire laser (M squared Solstis) at a central wavelength of 785 nm. To reject noisy wavelength variations from the laser beam, the incident path was guarded by a line filter at 785 nm. After the line filter, the laser line was passed through a 40X oil immersion objective (Nikon, NA 0.9) such that it hit the sample plane at a power of 130 mW. The back scattered Raman florescence was collected by the same objective. An edge filter (Semrock LPD02-785RU) was placed in the acquisition path to separate the fluorescence from the incident signal and hence allowing only Raman signal to pass through. The Raman photons were then fed into a monochromator (Semrock SR-303i, Andor technologies) coupled with a notch filter (Semrock NF03-785E) with 400 lines/mm grating, blazed at 914 nm and a deep depletion, back illuminated and thermoelectrically cooled CCD camera (Newton, Andor Technology). To confirm that the laser was not causing any photo-damage or denaturation of cells, a 5 second single acquisition for 5 minutes was continuously accumulated and no variation in the collected Raman spectra confirmed it. The laser wavelength was modulated over a range of 1 nm with a step size of 0.2 nm to acquire 5 Raman single cell spectra over a period of 5 seconds each. Raman spectra in the region of 934 cm^{-1} to 1774 cm^{-1} were subsequently analysed for the classification.

After the accumulation of Raman spectra, the data was processed for classification using an MLP. As the very first step of processing, single-cell Raman spectra were normalized with respect to the total spectral intensity which allowed for cancel-

lation of any noise generated in the signal due to the laser power fluctuations. The second step was to calibrate the Raman shift which was implemented by linearly calibrating these shifts with respect to the characteristic 1001.4 cm^{-1} Raman peak of the polystyrene bead.

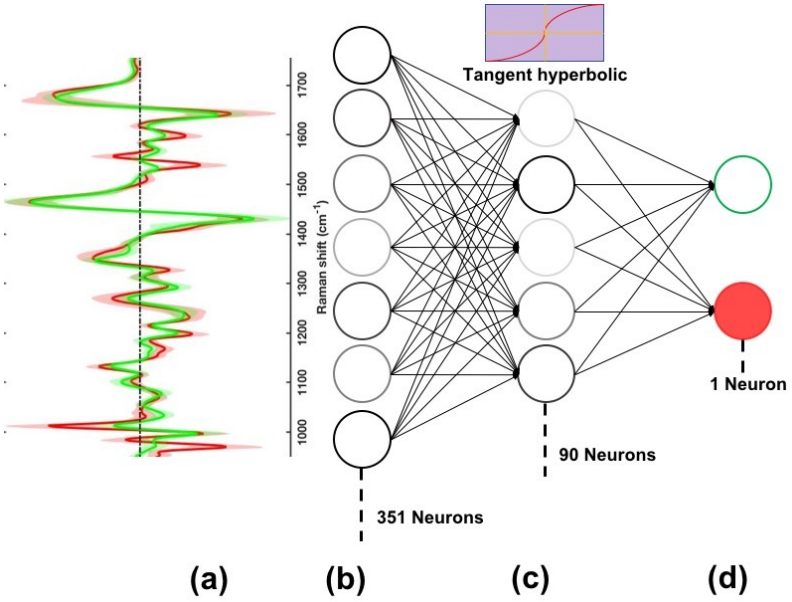

Figure 3.8: **Schematic of the Multi-layered perceptron for the classification of WMRS.** The network takes a WMR spectrum as the input (layer with 351 neurons) and processes it though a hidden layer (90 neurons) and finally connects to a sigmoid neuron for the binary classification. (a) Representation of mean WMR spectra for the two cell lines, (b) input Layer with nodes equal to the number of data points in each spectrum, (c) hidden layer with 90 neurons and tan hyperbolic as the activation function, (d) representation of classification layer with a single logistic sigmoid neuron for binary classification, here red neuron represents the Eosinophil class whereas the green neuron represents the Neutrophil class.

A modulated Raman spectrum was calculated for each cell as the first principal component using PCA (using the script A.4) on the normalized five Raman spectra by considering each excitation wavelength step as a parameter.

Each WMR spectra for a given biochemical structure is a signature representative of its chemical ingredients, hence the classification of the one dimensional WMRS dataset can be considered as a pattern recognition problem for a machine. To classify these 1D patterns, an MLP architecture (patternNet, MATLAB 2017b) was implemented. The MLP comprised of an input layer with 351 neurons, a hidden layer having 90 neurons with a tan hyperbolic activation function and a final output layer of one neuron with logistic sigmoid as the activation function. A schematic of

the MLP is represented in Fig. 3.8. This MLP model was chosen by considering its classification accuracy over the validation dataset. Before considering the final geometry, multiple geometries were considered by changing the number of hidden layers from one to three, and changing the number of neurons in each layer from 50 to 200 over the step of 10 neurons. The final selected MLP model was optimized using using stochastic gradient descent with momentum and adaptive learning rate [176] with momentum as 5×10^{-1} and L2 regularization as 1×10^{-4}. Maximum epochs were set to 2000 with validation patience of 5 iterations and a mini-batch size of 20 spectra was considered for training the model. An initial learning rate of 1×10^{-3} was considered which was reduced by a factor of 10 each time the validation loss plateaued.

The classification results obtained using MLP were compared with a previously known and applied method of non-parametric nearest neighbor algorithm based on PCA [164]. The PCA based model was optimized by changing the number of principal components and observing the effect on the classification accuracy of LOOCV over the training and validation dataset. First 8 principal components corresponding to 87.1 % of the total variance were considered since increasing the number of PCs only contributed to the noise in the spectra. After identifying the correct number of PCs, a classification was performed using LOOCV based on the nearest neighbour algorithm.

DHM data acquisition, phase image calculation and classification using CNN

The imaging data was acquired using a Mach-Zehnder interferometer based off-axis DHM (green and orange parts of Fig. 3.6). The setup of the mentioned DHM consisted of an incident laser (at wavelength of 532 nm, Spectra-Physics Millenia Vs). The light beam coming from the laser was passed through a beam splitter such that both the signal (power = 25.5 μW) and reference (power = 30.1 μW) arms were illuminated. The reference arm consisted of a 2X afocal telescopic system of lenses (f = 2.5 cm and 5.0 cm). The signal arm, on the other hand, consisted of a long working distance objective lens (Mitutoyo UK, M Plan Apo 10x NA = 0.23) coupled with a 60X microscopic objective (NA = 0.85, Nikon). The sample stage

was placed in the signal arm between the two objectives. The signal arm light beam passing through the sample was interfered with the light from the reference arm at the surface of the CCD camera which was set at 40 fps with an exposure time of 33.3 ms.

The acquired imaging data consisted of bright field and fringe images of the cells placed under the microscope. After the acquisition of these images, wide field phase images were calculated from the fringe images by calculating 2D Fourier transform followed by a 2D phase unwrapping algorithm [171]. Following the above, single cell phase images of the neutrophils and eosinophils with a size of 170 px $\times$ 170 px were manually cropped and further downsized to 150 px $\times$ 150 px using bi-cubic interpolation method [177]. These single cell phase images capture the refractive index variation across the cytoskeleton of a cell which acts as a morphological fingerprint for the specific cell type.

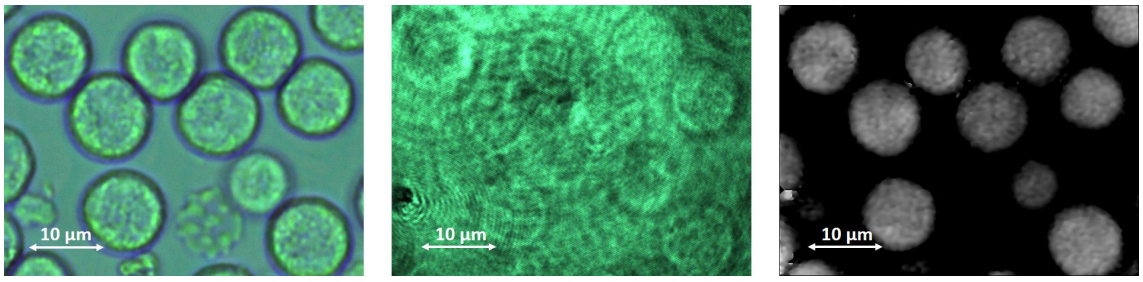

Figure 3.9: **Demonstration of phase image calculation.** (a) A subsection of the accumulated bright field image using our microscope, (b) fringe image for the same section, (c) phase image calculated from (b). Scale bar: 10 μm (image)

Following the calculation and augmentation of single cell phase images, a supervised classification of these phase images was performed. The classification was performed by using a deep learning based CNN (following the script A.1) and to benchmark the results another approach of calculating the intensity histograms and applying PCA and LOOCV was also considered.

The phase images can be considered as an input with a binary label $k \in \{0, 1\}$ indicating the two classes of cells, hence a supervised mechanism can be easily be constructed. This classification was performed using a deep learning based 17 layered CNN as represented in figure 3.10. The CNN processes the input phase images by passing through three convolution blocks (CBs), each containing a convolution layer, batch normalization layer [178] and a recitied linear unit layer. The final

block of the CNN was connected with a max pooling layer whose output was fully connected to the classification layer containing one neuron with logistic sigmoid as the activation function. The mentioned CNN geometry was chosen for achieving maximum classification accuracy over the validation dataset. Before considering the final CNN geometry, a range of CNNs were developed and tested over the validation dataset. These CNN were developed by changing the number of CBs from 2 to 10, changing the number of convolution layers from one to three and chaning the filter sizes in each convolution layer from 2 px × 2 px to 5 px × 5 px. Table 3.2.1 summarizes the complete CNN architecture and the total number of learnable parameters implemented.

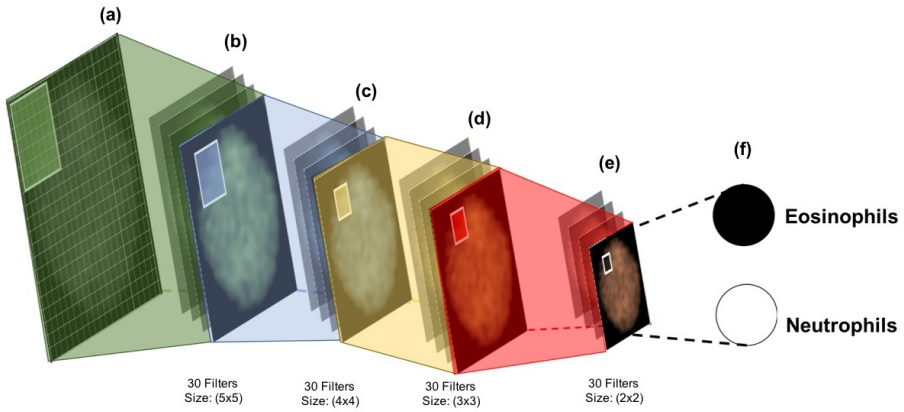

Figure 3.10: **Schematic of the optimized CNN for phase image classification.** The network takes phase images as its input and processes these though four convolution blocks and finally connects to a sigmoid neuron for the binary classification. Here we see the (a) Input phase image, (b) the convolution layer containing 30 5×5 filters with 1×1 stride and 1×1 padding, and (c) the convolution layer containing 30 4×4 filters with 1×1 stride and 1×1 padding, (d) the convolution layer containing 30 3×3 filters with 1×1 stride and 1×1 padding,(e) the convolution layer with 30 2×2 filters with 5×5 stride and 1×1 padding. Each convolution layer is followed by a batch normalization and ReLu Layer. The final convolution block has a max-pooling layer with 2×2 filter size and 1×1 padding and (f) shows the Binary representation of output classification layer with a single sigmoid neuron.

The training set for the CNN consisted of 235 single cell phase imaged of size 150 px × 150 px for each class. As explained in section 2.3.6, the binary cross entropy cost function (Eq. 3.7) was considered and optimized for each training example.

$$L(j,k) = \begin{cases} -A_1 \cdot k \cdot log(p(k|j)), & \text{for } k = 1 \\ -A_2 \cdot (1-k) \cdot log(p(k|j)), & \text{for } k = 0 \end{cases} \tag{3.7}$$

In the above equation, $p(k|j)$ is the probability that the network assigns to the label j with $A_1 = |N_1|/(|N_1| + |N_2|)$ and $A_2 = |N_2|/(|N_1| + |N_2|)$ with N_1 and N_2 being the total number of cases for each class. The weights and biases of the network were initialized by seeding the random environment. The network was trained end-to-end using stochastic gradient descent with momentum as 4×10^{-1} and L2 regularization as 5×10^{-6}. Maximum epochs were set to 100 with validation patience of 5 iterations and a mini batch size of 40 images was considered for training the model with a learning rate of 5×10^{-5}. Before the training process, a zero center normalization was implemented on the dataset.

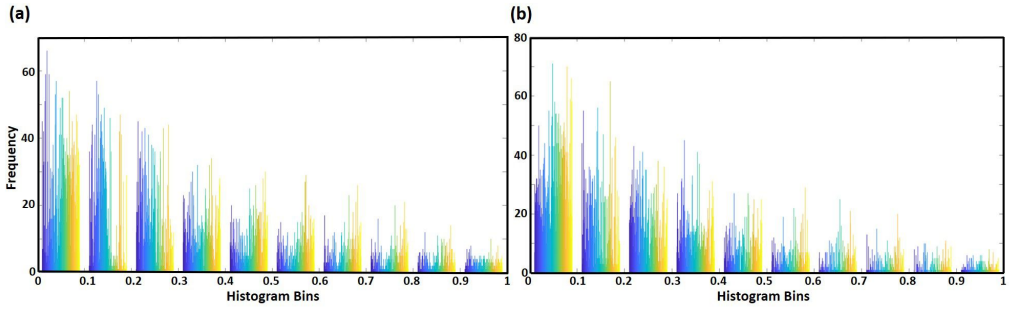

Figure 3.11: **Intensity histograms of the cellular phase images.** Representation of intensity histograms, calculated to represent the size of image, maximum OPD, and total OPD of (a) Neutrophils and (b) Eosinophils.

As a benchmark, the results obtained using the CNN were compared against PCA (explained in section 2.3.1 and following the script detailed in section A.4) and LOOCV based intensity histogram approach. As shown in Fig. 3.11, intensity histograms were calculated to represent the information about the size of image, maximum optical path difference (OPD), and total OPD over the entire cellular phase image. These histograms were then used as the one dimensional descriptor vectors for the datasets and PCA was conducted on the new dataset. The number of principal components were optimized to gain maximum classification accuracy over the training and validation datasets.

3.2.2 Results

Cell isolation

The isolation of the neutrohils and eosinophils was obtained by removing all the cells form blood which were now of interest. This was performed by using cell-lineage

S. No.	Layer Name	Type	Activations	Parameters	Total Parameters
1	Input	Image Input	150×150×1	-	0
2	C1	Convolution	148×148×30	Weights 5×5×1×30 Bias 1×1×30	780
3	B1	Batch Normalization	148×148×30	offset 1×1×30 scale 1×1×30	60
4	R1	ReLU	148×148×30	-	0
5	C2	Convolution	147×147×30	weights 4×4×30×30 bias 1×1×30	14430
6	B2	Batch Normalization	147×147×30	offset 1×1×30 scale 1×1×30	60
7	R2	ReLU	147×147×30	-	0
8	C3	Convolution	147×147×30	weights 3×3×30×30 bias 1×1×30	8130
9	B3	Batch Normalization	147×147×30	offset 1×1×30 scale 1×1×30	60
10	R3	ReLU	147×147×30	-	0
11	C4	Convolution	30×30×30	weights 2×2×30×30 bias 1×1×30	3630
12	B4	Batch Normalization	30×30×30	offset 1×1×30 scale 1×1×30	60
13	R4	ReLU	30×30×30	-	0
14	Maxpool	Max Pooling	31×31×30	-	0
15	FC	Fully Connected	1×1×2	weights 2×28830 bias 2×1	57662
16	Softmax	Softmax	1×1×2	-	0
17	Output	Classification output	-	-	0
	Total CNN learnable Parameters			**84,872**	

Table 3.2: Table to summarize the optimal CNN architecture and the total number of learnable parameters.

specific antibodies. To chemically confirm the isolation these cell types, flow cytometry was used to confirmed the isolation of monodisperse cell populations by forward and side scatter profiles. These profiles were negative for the T lymphocyte marker CD3, however, they stained 100 % for CD66b for eosinophils and 100% positive for CD15 for neutrophils(Fig. 3.12). The negative isolation of cells allowed for completely label-free (antibodies or beads) untouched cells and therefore represented as far as possible cells *in situ*.

Digital holographic microscopy

Bright field and holographic fringe images were accumulated at a rate of 40 frames per second. These images consisted a field of view of 50 μm × 40 μm area (Fig. 3.9) which covered a maximum of nine cells in a single field of view. This enabled the high-throughput capabilities of the system of more than 100 cells per second.

The phase images were calculated using the fringe images and single cell images (170 px × 170 px) were cropped manually for the two cell types. Following this, the

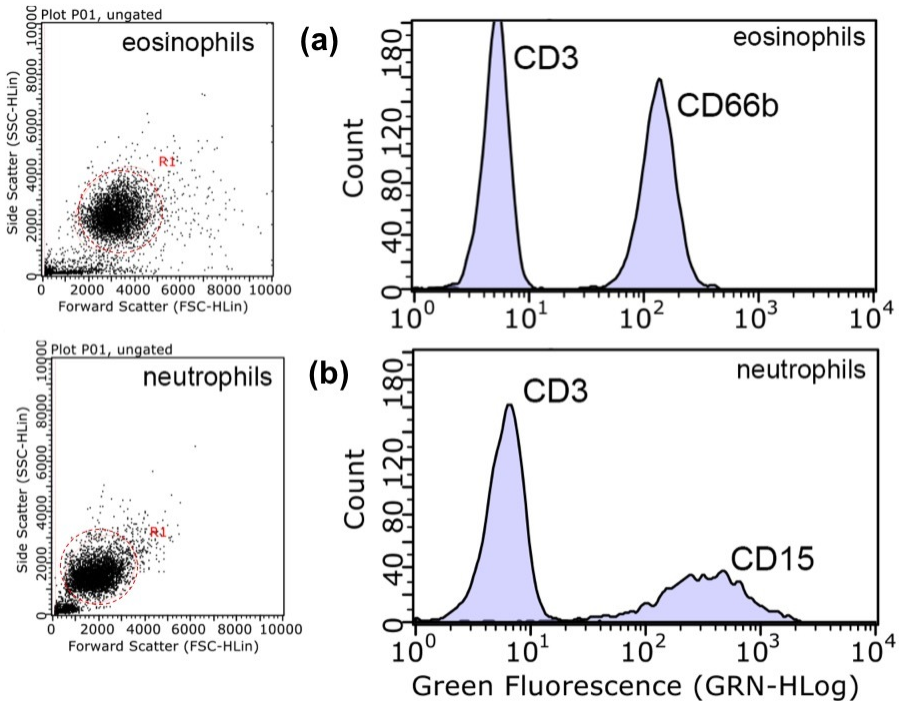

Figure 3.12: **Chemical quantification of granulocyte isolation.** Flow cytometric analysis of purified untouched eosinophils (a) and neutrophils (b) demonstrating forward and side scatter profiles of purified cells and antibody staining with anti-CD3-FITC (negative control) and anti-CD66b-FITC for eosinophils, and anti-CD3-FITC (negative control) and anti-CD15-FITC for neutrophils.

single cell images were downsized to 150 px × 150 px for augmentation purposes. The single cell phase images (Fig. 3.13) acquired for the two cell types signified morphological similarities with ~ 10 μm diameter possessing the granularity of cells as represented by the presence of differences across the central cellular structure.

The phase image dataset comprising the images of neutrophils and eosinophils were accumulated from three different donors. This was done to consider the variability that may arise due to blood drawn from different donors. The dataset was divided such that the training set solely consisted of images corresponding to donor 1, validation set consisted of images for donor 2 and test set consisted of images acquired using the blood from the donor 3.

After splitting the data, CNN geometry (Tab. 3.2.1) was optimized by training on the training/validation datasets as explained before. This optimal CNN was then implemented on the test dataset to classify the novel images from the third donor. The classification of the test set by the CNN resulted in a sensitivity of 100 % and specificity of 82.7 % with an overall classification accuracy of 91.3 %. These

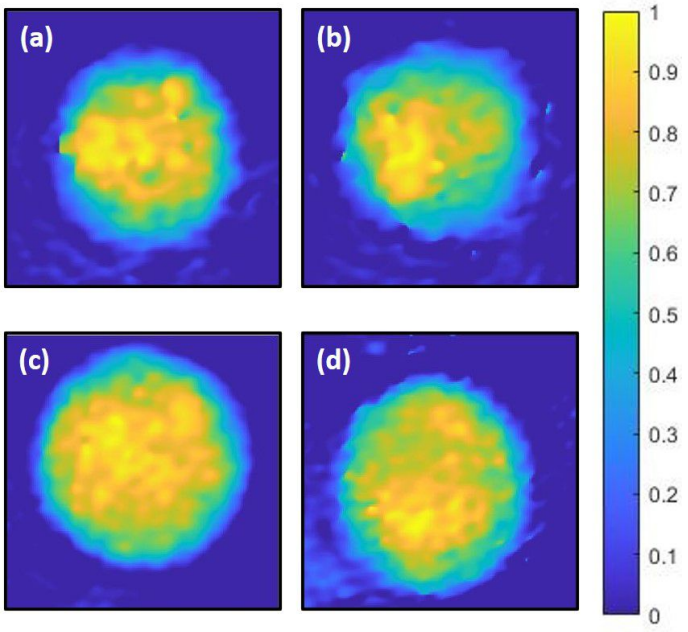

Figure 3.13: **Normalized phase images of the Granulocytes.** (a), (b) Eosinophils; (c),(d) Neutrophils. These images represent the inter-cellular structural variation in the form of refractive index map which expresses the granularity of the two cell types. Color bar represents the normalized phase difference between the signal and reference arm.

results are summarised in the form of a confusion matrix in table 3.3. As a comparison, the phase images were also classified using PCA and LOOCV method (results summarised in Tab. 3.3) described above. This method resulted in a sensitivity of 87.9 % and a specificity of 72.4 %. To understand the weak of performance by this method, the principal components were plotted as scatter plot (Fig. 3.14). These plots displayed an absence of variability among the phase images and hence confirmed the lack of sensitivity by this method to detect the morphological differences between the two cell types.

Table 3.3: Confusion matrix representing the prediction accuracy of the trained CNN and PCA/LOOCV for classification of the neutrophils and eosinophils on the test dataset. Each row of the matrix expresses the total number of phase images of the cells available for classification, whereas each column represents the predicted cell lines. Thus, the diagonal elements of the confusion matrix represent the correct predictions made by the network whereas off-diagonal terms represent the inaccurate predictions.

C.M.	**Pred. E**	**Pred. N**	**Pred. E**	**Pred. N**
	CNN		**PCA**	
Act. E	*58*	0	*51*	7
Act. N	10	*48*	16	*42*

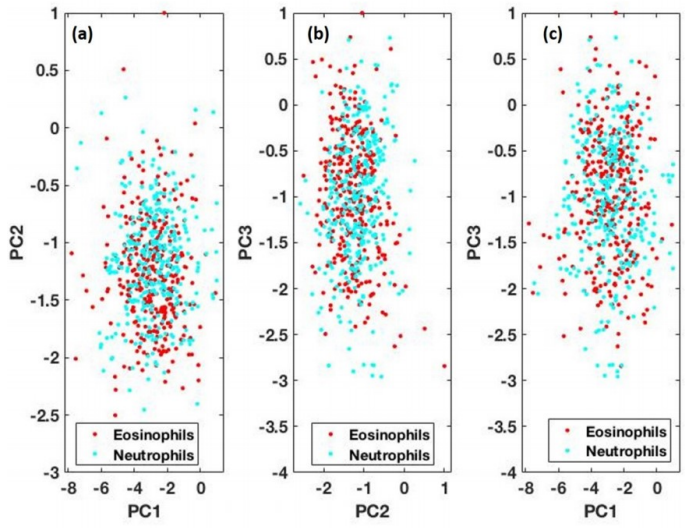

Figure 3.14: **PCA scatter plot for DHM**. (a) PC1-PC2 scatter plot for the two cell types; (b) PC2-PC3 scatter plot for the two cell types; (c) PC3-PC1 scatter plot for the two cell types; Eosinophils are represented by the data points in red whereas the neutrophils are represented by the data points in blue

Raman spectroscopy

Standard and WMR spectra were collected from the neutrophils and eosinophils. Figure 3.15 (a) and (b) depicts the Raman spectra for the granulocyte cells with a deviation of 0.1 nm in the incident laser wavelength as explained before and (b) depicts the pairwise comparison of WMR spectra for the two cell subsets.

The collected WMRS dataset was classified using the multi layered perceptron (MLP) model explained in the methods section. To consider inter-donor variability, we trained the MLP on the data considered from the two donors and tested its performance over the data accumulated from an independent donor, this approach of splitting the data resulted in low classification ability of the MLP based model (validation accuracy of 97.1 % and test accuracy of 67.0 % with 120 hidden neurons). The reason for such poor performance can be attributed to the presence of experimental variations while accumulating the data for individual donors; this in-turn contributes to the biasing for the training of MLP based model. In addition, the absence of inter-donor variability can be verified by considering the PCA and t-SNE (Fig. 3.16) analysis of the WMRS data. Each method gave a clear clustering of the WMR spectra of the two cell types over all the data collected from all the three donors. Hence to overcome the effects of experimental bias [179] in the

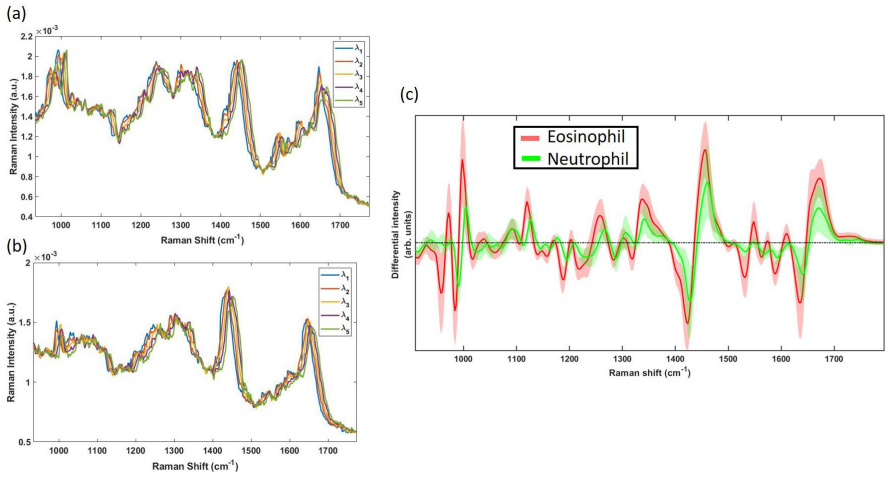

Figure 3.15: **Evaluation of WMRS for the granulocyte**. Raman spectra accumulated with a wavelength deviaiton of 0.1 nm for the (a) Neutrophils (b) Eosinophils. (c) Wavelength modulated Raman spectra illustrating pairwise comparison between eosinophils and neutrophils. Solid lines show the mean spectrum for each cell subset and shadowed regions represent the standard deviation.

dataset, the complete dataset was randomly divided into a training set (70 % of the total spectral dataset), a validation set (15 % of the total spectral dataset) and a test dataset (15 % of the total spectral dataset) [180–182]. The MLP showed a sensitivity of 96.7 % and a specificity of 100 % on the test dataset as depicted by the confusion matrix presented in table 3.4.

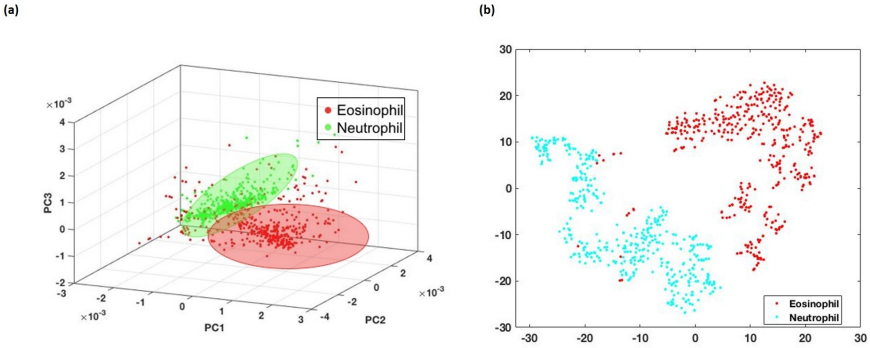

Figure 3.16: **Demonstration of high molecular sensitivity using PCA and t-SNE.** (a) 3D PC scatter plot showing the clustering of the neutrophils and eosinophils. The red points on the scatter plot correspond to the Eosinophil WMR spectra whereas the green points on the scatter plot correspond to the neutrophil WMR spectra. (b) t-SNE scatter plot in 2D showing a clear clustering of the complete dataset. The red points in the scatter plot correspond to Eosinophils WMR spectra whereas blue points correspond to Neutrophil WMR spectra.

PCA in combination with LOOCV was also performed on the collected Raman dataset. First eight PCs which accounted for 87.1 % of the total variance, were

Table 3.4: Confusion matrix representing the prediction accuracy of the trained MLP and PCA/LOOCV for classification of the neutrophils and eosinophils on the test dataset. Each row of the matrix expresses the total number of WMRS of the cells available for classification, whereas each column represents the predicted cell lines. Thus, the diagonal elements of the confusion matrix represent the correct predictions made by the network whereas off-diagonal terms represent the inaccurate predictions.

C.M.	**Pred. E**	**Pred. N**	**Pred. E**	**Pred. N**
	MLP		**PCA**	
Act. E	*60*	2	*58*	4
Act. N	0	*59*	1	*58*

considered. The PC scatter plot, shown in Fig. 3.16 (a), of the analyzed data points displayed their clustering corresponding to individual classes in the PC space. The classification accuracy of the analysis was calculated using LOOCV. The PCA based analysis resulted in a sensitivity of 93.5 % and specificity of 98.3 % as calculated from the confusion matrix depicted in Table 3.4. Additionally, to further demonstrate the chemical specificity of the Raman data, t-SNE (detailed in section 2.3.5) based approach was also implemented; this has been displayed in Fig. 3.16 (b) demonstrating the formation of two distinct clusters for the two cell types.

The network classified the input images at a rate of 435 frames per second which was more than 3 times the single-cell image acquisition rate. This indicates that CNN complements the high-throughput capabilities of the DHM system and can be used in real time for the classification purpose.

The above represented results confirm that digital holographic microscopy when combined with deep learning can be used to create a highly-accurate hemogram device with high-throughput for the label-free classification of the cells of the immune system. Hence to identify the limits of a DHM based hemogram system, the next study was conducted where the T-cell subsets namely $CD4^+$ and $CD8^+$ cells were considered for the classification purposes.

3.3 Determination of throughput limit of DHM based hemogram to classify the T-cell subsets

3.3.1 Methods

Cell Isolation

Cell isolations were performed, after ethical review from by the School of Medicine at the University of St. Andrews, using cells isolated from buffy coats of six different healthy donors obtained from NHS UK. For the isolation of $CD4^+$ and $CD8^+$ T cells, PBMC were isolated from the buffy coats by centrifugation at 1,200 rpm for 20 mins at room temperature on Ficoll-Paque at density 1.077 g/ml (Thermofisher, UK). $CD4^+$ and $CD8^+$ T cell populations were isolated by negative depletion (Dynabeads CD4 T cells, 11346D and Dynabeads CD8 T cells, 11348D, Thermofisher UK). After the isolation, the purified cells were cultured in RPMI 1640 supplemented with 5% Foetal Bovine Serum (both Thermofisher, UK).

To confirm the purity of the cell samples of each were stained with combinations of the following antibodies (CD3-PE and -FITC, clone HIT3a, eBioscience UK, CD4-PE and -FITC, clone OKT4, eBioscience UK, CD8-PE and -AF488, clones SK1, eBioscience UK, and FAB1509G, R&D UK). Cells were analysed on a Guava 8HT cytometer (Merck Millipore, UK).

For the optical analysis, centrifugation was performed to isolate the cells. These cells were then resuspended in Phosphate Buffer Saline (PBS) with 0.5% FBS solution to avoid aggregation. The optical measurements were performed on an assembly of quartz slides and 20 μl of prepared cell suspension. The cell suspension was transferred to the center of a clean quartz slide (25.4 mm $\times$ 25.4 mm $\times$ 1 mm) chamber - formed by use of an 100 μm thick vinyl spacer. This chamber was covered from the top using a thin quartz slide (25.4 mm $\times$ 25.4 mm, 0.11 mm - 0.15 mm thickness). This assembly was inverted and left for 20-30 minutes to avoid cellular motion.

Digital holographic microscopy

A Mach-Zehnder interferometer-based off-axis digital holographic microscope [171] was modified to capture the holographic images of the cells at various field of views

(Fig. 3.17). Details of setup are summarised in section 3.2.1. For this study, three different microscopic objectives were considered, a 20X microscopic objective (NA = 0.4, Nikon Japan 130314), a 60X microscopic objective (NA = 0.85, Nikon) and a 100X microscopic objective (NA = 0.9, Nikon Japan 230538). The sample was placed between the two objectives and the image was interfered with the light from the reference arm at the surface of the CCD camera (Ximea XiQ MQ013MG-E2). This camera was set to accumulate 16 bit images with a frame rate of 60 fps.

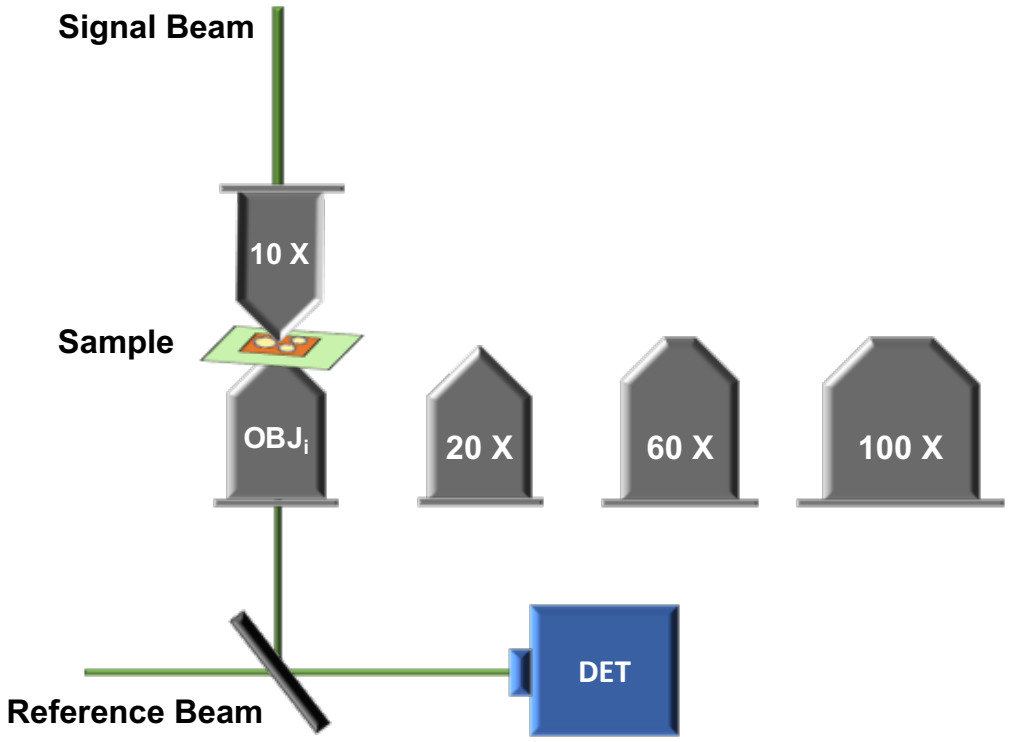

Figure 3.17: **Schematic of optical modifications made to digital holographic microscope**. Schematic of modifications made to the the DHM system [171] implemented for the data acquisition. The microscopic objectives with magnifications of 20X, 60X and 100X were used as shown.

The data for the two cell types was acquired separately by using the three objectives mentioned earlier. The use of 20X objective resulted in the images with largest field of view (FOV) but with the worst resolution (lateral: 0.66 μm; axial: 6.65 μm). The images captured using the 60X objective showed moderate size of FOV with a reasonable resolution (lateral: 0.31 μm; axial: 1.47 μm) and the images acquired by replacing the 100X objective had the smallest FOV with highest resolution (lateral: 0.29 μm; axial: 1.31 μm). Using the 20X objective, a total of 901 wide-field images were captured from three different donors whereas a total of 1447 images from three

different donors were captured using the 60X objective and 2272 images from four different donors were captured using the 100X objective.

Haugh transform based detection of cells

As mentioned, the three objectives provide a different FOV which result in a variable radii of cells for each case. To automate the detection of cells and overcome the problem of variable radii, a Haugh transform based method was implemented. A prewritten matlab script [183] was considered whereby the input variables controlling the search for *range of radii*, *gradient threshold* and *radius of filter* (to be used in the search for local minima in the accumulation array) were optimized. The code was run using MATLAB 2019b. After isolating the single cell images, the total images considered for each optical configuration are presented in table 3.5.

S.No.	**20X**		**60X**		**100X**	
	CD4	**CD8**	**CD4**	**CD8**	**CD4**	**CD8**
Train/Val	2385	2056	1066	971	704	704
Test	344	323	84	77	104	96

Table 3.5: Table summarizing the total number of single cells phase images considered for different optical configurations.

Optimization of the CNN geometry

The optimization of the CNN geometry poses a difficult problem invoking a multitude of parameters such as the selection of layers, number of filters specific to the layers, sizes of those filters, strides, padding, dropout ratio and the number of neurons in the fully connected layers. The layers have to be selected from the set of convolution layer, rectified linear unit (reLu) layer, batch normalization layer, dropout layer, max pooling layer, and fully connected layer. All these parameters have to be optimized in order to gain maximum classification accuracy over the validation dataset.

To solve this problem, a particle swarm optimization (PSO) based approach was implemented (using code detailed in section A.2). PSO is a type of swarm intelligence method for global optimization where each individual (named particle) of the population (called swarm) adjust their trajectory towards the previous best

position attained by any member of their topological neighborhood. This approach is used to minimize the error output of an objective function.

For the optimization of CNN geometry, an objective function was developed as a training instance. This training instance consisted of the layers to be identified for the network geometry. To identify the layers of the CNN, the layers namely convolution layers with the filter sizes ranging from 1 to 5 with an increment of 1, a batch normalization layer, a rectified linear unit layer, a max pooling layer with filter size of 2 and the fully connected layers were considered. The input layer was set at image input layer and the output layer was set at fully connected layer with 2 neurons (representing each class) followed by softmax layer and a classification layer. The training instance gave an output of cost function described as the mean of sensitivity (Eq. 3.8a) and specificity (Eq. 3.8b) for the validation dataset (calculated as 3.8c) calculated by testing the trained network's classification ability over the validation dataset. The objective function took input as: the selection of layers (layer index), number of neurons in fully connected layers, number of filters for the convolution layers, padding to be provided to the layers and the dropout ratio. Each parameter was restricted and conditionally valued in the objective function. The restrictions posed over the individual parameters were to restrict the layer index between 2 and 9 and the index value was placed as 0 when crossing either limit. To keep the network geometry with the bounds of the GPU's memory, the number of convolution filters were also restricted to be greater than zero and less than 75. The number of fully connected neurons and dropout layers were restricted to be more than zero.

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{3.8a}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{3.8b}$$

$$\text{Cost} = 1 - \frac{\text{Sensitivity} + \text{Specificity}}{2} \tag{3.8c}$$

Here, TP is true positive, TN is ture negative, FP is false positive and FN is false negative. A total of 40 particles were considered for the PSO algorithm. Initializing the particles' position and velocity randomly, the algorithm evaluated the cost of the objective function for each particle. After the cost evaluation, the particle with

minimum evaluated least cost was considered as the leader (x_L) and all the particles' trajectory and velocity were relatively updated from the leader and preceding velocities using the Equations 3.9a and 3.9b.

$$\vec{v_i} = (1 - \frac{\sqrt{k \times fi}}{2})V_i^p + kf_i(x_L - x^p) \tag{3.9a}$$

$$\vec{x_i} = x_p + \gamma(1 - \frac{\sqrt{k \times fi}}{2}\vec{v_i} + (1 - kf_i)(x_L - x^p)) \tag{3.9b}$$

Here, k represents the weight of stochastic element, f_i represents the random weight of the stochastic element and γ represents the weight of the position updation. For the experiments, k was chosen as 2 whereas γ was chosen as 2.5 and f_i was randomly generated for each particle at each iteration. These values were considered after varying the values between 1 to 5 with the step of 0.5 for getting the minimum cost with least number of iterations.

Cycle generative adversarial training for image transformation

The images acquired from three optical configurations allow for different throughput. With decreasing magnification the throughput increases however the details associated with respect to individual cell types are lost. Hence, it would be beneficial to identify a numerical method where the images accumulated from a lower magnification system may be transformed into a higher resolution image. To solve this problem, single image super-resolution methods of machine learning were considered. Generally, a lower resolution image y can be modelled from a higher resolution image x as:

$$y = (x \otimes k) + n \tag{3.10}$$

Here, $x \otimes k$ is the convolution between a kernel k and unknown higher resolution image x and n is a stochastic noise term. The convolution between x and k results in a downsampled image. Problem posed by Eq. 3.10, is ill posed since the inverse solution may result into multiple higher resolution images. Deep learning has proven to be a very effective solution for this problem [184]. Using deep learning, paired image super-resolution has been vividly applied in photonics [185–187]. However,

applying the super-resolution to unpaired image sets is still quite unexplored.

Using the DHM system explained above, it is very challenging to identify the same cells using two different magnifications. Hence considering this problem, a cycle GAN type training (using code detailed in section A.3) was considered to train a deep generative network (schematic represented in Fig. 3.18). Cycle GAN type training module has been popularly known to be applied for the transformation of unpaired images [132]. Two deep convolutional networks were developed by such that the size of input image could be down-sampled and then up-sampled to get the required size at the output. For the transformation of phase images captured using 20X optical configuration to 100X optical configuration, a 54 layered CNN ($\mathcal{G}_{20X\rightarrow 100X} \coloneqq \mathcal{G}_a$) was developed such that the network filters were changed in the step of 8 units with respect to the network performance on the validation dataset. Similarly, the inverse network (100X to 20X) with 34 layers ($\mathcal{G}_{100X\rightarrow 20X} \coloneqq \mathcal{G}_b$) was also optimized. As a part of the cycle GAN model, two discriminator models with 23 layers ($\mathcal{D}_{100X} \coloneqq \mathcal{D}_a$) and 15 layers ($\mathcal{D}_{20X} \coloneqq \mathcal{D}_b$) were also developed. To train the deep models, 300 phase images each of $CD4^+$ and $CD8^+$ T cells accumulated using both the 20X and 100X optical configurations were randomly selected. Out of these, 225 images were considered for training and 75 images were considered for validation.

The cycle-GAN model was trained by using the proposed cycle consistency loss coupled with the GAN loss. The GAN loss function is computed as:

$$\mathcal{L}_{\text{GAN}}(\mathcal{G}_a, \mathcal{D}_a, X, Y) = \mathbb{E}_{y\sim p_{data}(y)}[\log \mathcal{D}_a(y)] + \mathbb{E}_{x\sim p_{data}(x)}[\log(1-\mathcal{D}_a(\mathcal{G}_a(x)))] \quad (3.11)$$

The cycle consistency loss $\mathcal{L}_{\text{cyc}}(\mathcal{G}_a, \mathcal{G}_b)$ is computed, to satisfy the condition $x \rightarrow \mathcal{G}_a(x) \rightarrow \mathcal{G}_b(\mathcal{G}_a(x)) \approx x$, as:

$$\mathcal{L}_{\text{cyc}}(\mathcal{G}_a, \mathcal{G}_b) = \mathbb{E}_{x\sim p_{data}(x)}[||\mathcal{G}_b(\mathcal{G}_a(x)) - x||_1] + \mathbb{E}_{y\sim p_{data}(y)}[||\mathcal{G}_a(\mathcal{G}_b(y)) - y||_1] \quad (3.12)$$

Here, the variables x and y represent the phase images obtained using 20X and 100X

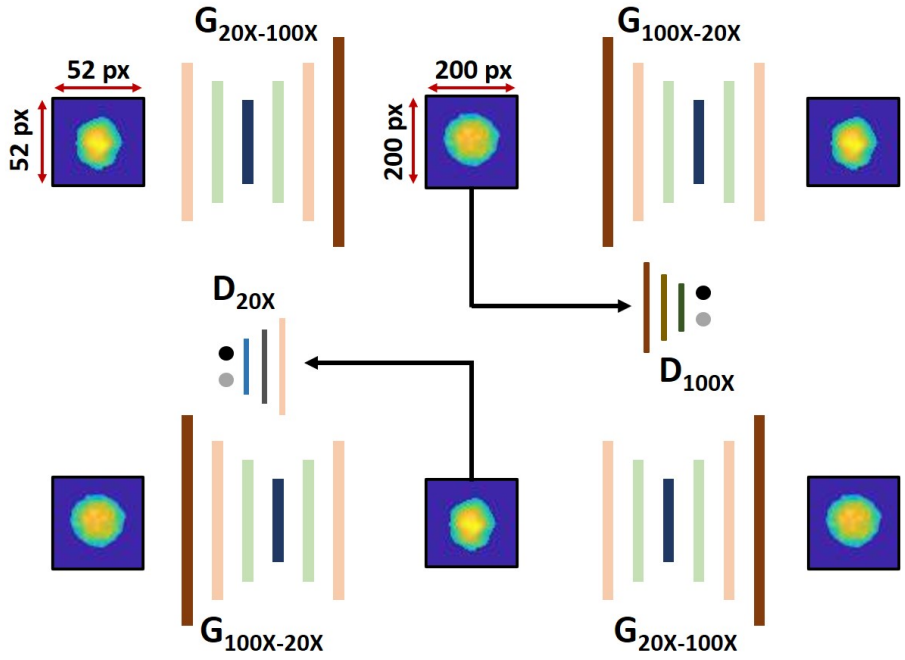

Figure 3.18: **Schematic of cycle GAN model applied for super-resolving the phase images**. The generative models $\mathcal{G}_{20X\rightarrow 100X}$ and $\mathcal{G}_{100X\rightarrow 20X}$ are trained with in a cycle consistent manner such that the inverse transformation of the images is conserved.

optical configuration respectively. The combined GAN loss can be written as:

$$\begin{aligned}\mathcal{L}(\mathcal{G}_a, \mathcal{G}_b, \mathcal{D}_a, \mathcal{D}_b) = \mathcal{L}_{\text{GAN}}(\mathcal{G}_a, \mathcal{D}_a, X, Y) \\ +\mathcal{L}_{\text{GAN}}(\mathcal{G}_b, \mathcal{D}_b, Y, X) \\ +\lambda\mathcal{L}_{\text{cyc}}(\mathcal{G}_a, \mathcal{G}_b)\end{aligned} \tag{3.13}$$

Here, λ is a hyperparameter which was chosen as 10 for this application. During the training, the objective is to minimize the combined loss for the generator networks while maximizing the loss for the discriminator networks:

$$\mathcal{G}_a^*, \mathcal{G}_b^* = \arg\min_{\mathcal{G}_a,\mathcal{G}_b}\max_{\mathcal{D}_a,\mathcal{D}_b}\mathcal{L}(\mathcal{G}_a, \mathcal{G}_b, \mathcal{D}_a, \mathcal{D}_b) \tag{3.14}$$

To achieve this training, the images were considered in the mini batches of 45 images. An Adam optimizer was considered with a learning rate of 2×10^{-4}, gradient descent factor of 0.5 and a squared gradient descent factor of 0.99. The network performance was validated after 25 iterations using 25 randomly selected images belonging to both $CD4^+$ and $CD8^+$ T cell types. The training was continued

for a total of 5000 epochs.

k-means segmentation

To identify the degree of granularity exhibited by the phase images of the two cell types, a method of k-means clustering (detailed in section 2.3.3) based image segmentation [188] was implemented, using MATLAB 2019b. The phase images corresponding to each objective were considered individually and the number of segmentation classes were increased by one step until the algorithm returned a saturation solution of discontinuous boundaries. In this case, a class represents the distribution of refractive index across the cellular structure representing the variation of granularity for the distinct FOV's.

3.3.2 Results

Cell isolation

Untouched human blood $CD4^+$ and $CD8^+$ T cells were obtained by negative depletion, in which other cells not of interest were removed using cell-lineage specific antibodies. Flow cytometry (Fig. 3.19) confirmed the purity of the cell populations in line with previous studies [164, 171], with $CD4^+$ cells isolated at an average of 89% (n=3) and $CD8^+$ cells at an average of 86% (n=3).

Automated detection of cells and phase image calculation

Bright field and fringe images were captured using the DHM system by replacing the objectives as mentioned before. The images captured using different objectives demonstrate different properties in terms of resolution and overall field of view. The acquired fringe images were considered to calculate the phase images. After the calculation of phase images for all the different FOVs, an automated cellular detection of cells was performed using circular Haugh transform [183] on the bright field images.

To detect the cells using circular Haugh transform, the mentioned parameters were optimized for each optical configuration. Table 3.6, summarizes the optimal parameters used for the detection of cells for each optical configuration. These

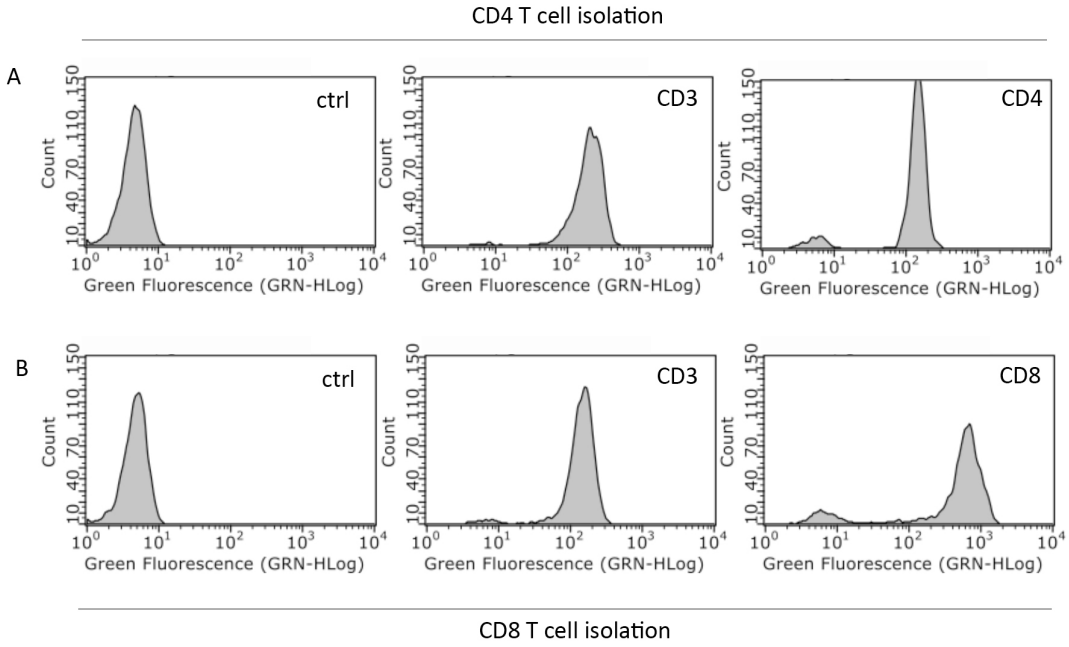

Figure 3.19: **Representative flow cytometric plots of CD4$^+$ and CD8$^+$ T cells purified by negative depletion**. Purified cell samples were stained with anti-CD3, -CD4 or –CD8-FITC or AF488 coupled antibodies and analysed by flow cytometry for (A) CD4 T cells and (B) CD8 T cells. Average purity of three separate purifications is reported in the main text.

parameters namely - range of radii, gradient threshold and radius of search filter were optimized with respect to the size of cells, the mean magnitude of gradient for the empty space and the radius of cells respectively for the particular optical configuration.

Optical Configuration	Range of radii (px)	Gradient threshold	Radius of search filter (px)
20X	15 - 20	10	12
60X	25 - 50	8	43
100X	60 - 100	5	51

Table 3.6: Table summarising the optimal parameters identified for the automated detection of cells using different optical configurations.

The images accumulated using the three optical configuration exhibit varying FOVs and resolutions. Fig. 3.20 demonstrate the automatic cellular detection for various FOVs. As summarised in Table 3.7, the FOV achieved by using the 20X objective was greatest at 100 μm $\times$ 130 μm (which may allow imaging a maximum of 1300 cells in one snapshot permitting the maximum possible throughput of 78,000 cells per second), however, the resolution of the accumulated images was poor. For imaging using the 60X objective, a smaller FOV of 68 μm $\times$ 64 μm was achieved

(allowing a maximum of 36 cells resulting in the highest possible throughput of 2,160 cells per second) with a reasonable resolution. Imaging using the 100X objective resulted in a much smaller FOV of 40 μm $\times$ 32 μm (enclosing a maximum of 12 cells and allowing a highest possible throughput of 720 cells per second) with the highest resolution.

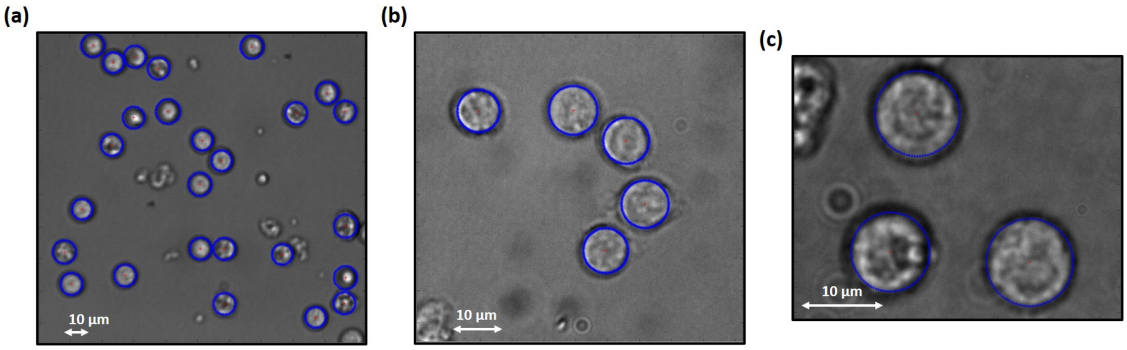

Figure 3.20: **Automatic detection of cells using Haugh transform** Subsection of Bright field images recovered from (a) 20X Objective (100 μm $\times$ 130 μm) (b) 60X objective (68 μm $\times$ 64 μm) and (c) 100X objective (40 μm $\times$ 32 μm). Blue highlighted regions represent the automatic detection of cells for three FOV's using Haugh transform circular detection.

Optical Configuration	Field of View	Maximum Throughput
20X	100 $\times$ 130 μm	78000 cells/s
60X	68 $\times$ 64 μm	2160 cells/s
100X	40 $\times$ 32 μm	720 cells/s

Table 3.7: Summary of field of views and maximum allowed throughput for the three optical configurations

With respect to the numerical aperture of the microscopic objectives, the retrieved phase images show the differences in resolution. The phase image recovered using 20X objective with a numerical aperture (NA) of 0.4, displays the least resolution for both the cell lines (Fig. 3.21 (a),(d)). The application of 60X objective with the NA of 0.8 results in moderately resolved phase images (Fig. 3.21 (b),(e)) and the phase images calculated from the fringe images captured using the 100X objective (with the NA of 0.9) were highly resolved.

Quantification of granularity

After the extraction of single cell phase images, a quantification of granularity or resolution probed by different optical configurations was implemented. This was

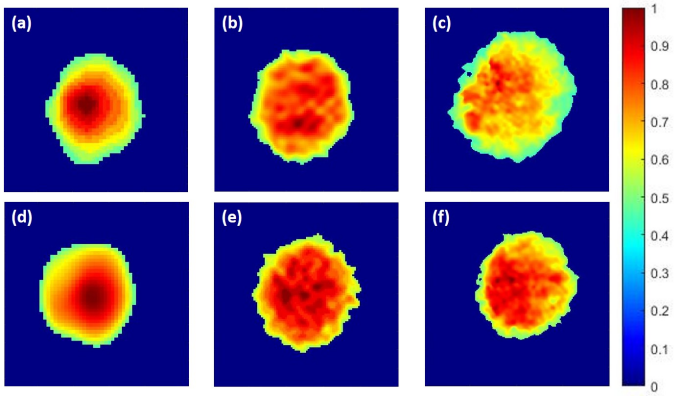

Figure 3.21: **Normalized phase images of T Cells.** Single cell normalized phase images of the $CD4^+$ cells retrieved using (a) 20X objective (b) 60X objective (c) 100 X Objective; Single cell normalized phase images of $CD8^+$ cells retrieved using (d) 20X objective (e) 60X objective (f) 100X objective. Colorbar represents the normalized phase gain of the signal arm with respect to the reference arm.

achieved by using image segmentation based on k-means clustering.

As anticipated, the phase images of the two cell types show a very similar variation in resolution with respect to the objectives. As shown in Fig. 3.22, the algorithm when applied over the phase images of the two cells for the 20X objective, saturated at 8 segments. For the phase images accumulated using 60X objective, the algorithm saturated at 9 segments for $CD4^+$ cells whereas it saturated at 10 segments for the $CD8^+$ cells. When the algorithm was implemented over the phase images captured using 100X objective, it saturated at 11 segments for both the cell types.

Classification of phase images

The next step for the analysis of the single phase images was to classify them with respect to the cell types. This was achieved by employing the CNNs which were optimized by implementing PSO algorithm as explained before. For gaining statistical relevance over the dataset, the optimized CNN geometries were tested on the validation and test sets for five instances.

The three optical configurations, resulted in different sizes of single cell phase images as 52 $\times$ 52 px for 20 X objective, 100 $\times$ 100 px for 60 X objective and 200 $\times$ 200 px for 100X objective. Hence the optimal CNN geometry also displayed a variation in size. For the 20X optical configuration, the optimal CNN geometry

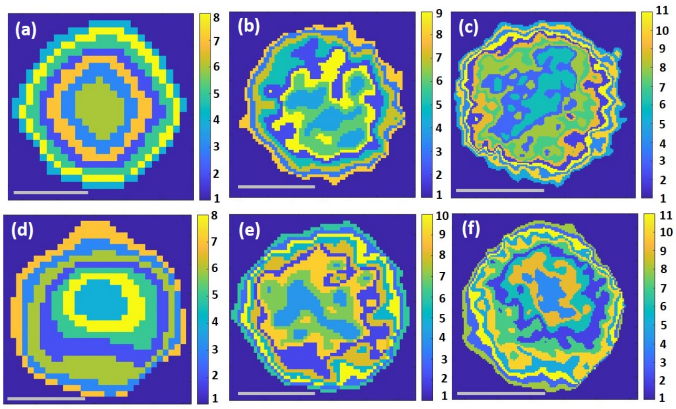

Figure 3.22: **Quantification of granularity for T-cells.** k-means clustering based image segmentation of the phase images of $CD4^+$ cells into (a) 8 sections for 20X objective (b) 9 sections for 60X objective and (c) 11 sections for 100X objective. k-means clustering based image segmentation of the phase images of $CD8^+$ cells into (a) 8 sections for 20X objective (b) 10 sections for 60X objective and (c) 11 sections for 100X objective. Scale bars represent 5 μm.

(summarized in Table 3.8) was identified with a total of six layers with 39,998 parameters. Using the validation set, the CNN returned a sensitivity of 63.13 % ± 2.23 % and specificity of 64.93% ± 5.65%, whereas when considered for the test dataset, the CNN resulted in a sensitivity of 64.07 % ± 2.64 % and a specificity of 56.83 % ± 2.36 %.

S.No.	Type	Activations	Learnable parameters	Total learnable parameters
1	Image Input	52x52x1	-	-
2	Convolution Layer	48x48x9	Weights 5x5x1x9 Bias 1x1x9	234
3	Max Pooling Layer	47x47x9	-	-
4	Fully Connected Layer	1x1x2	Weights 2x19881 Bias 2x1	39764
5	Softmax Layer	1x1x2	-	-
6	Classification Layer	-	-	-
	Total CNN learnable parameters		**39,998**	

Table 3.8: Table summarizing the optimal CNN geometry for the classification of phase images of $CD4^+$ and $CD8^+$ T-cell acquired using the microscopic objective with 20X magnification.

For the 60X optical configuration, the optimal CNN geometry (Table 3.9) was identified as a slightly longer network. This geometry comprised a total of 16 layers with 226,707 parameters. On the validation set, the classification efficiency of the CNN resulted in 70.94 % ± 2.27 % sensitivity and 65.52 % ± 2.08% specificity, whereas on the test set the trained CNN resulted in a specificity of 69.92 % ± 3.91% and a sensitivity of 69.59 % ± 3.10 %. Finally, the optimization routine was also implemented on the phase images acquired using the 100X optical configuration and

this resulted in a CNN geometry (Table 3.10) of 24 layers with 1,603,327 parameters. This geometry when applied over the phase images from validation dataset resulted in the specificity of 80.28 % ± 1.17 % and a sensitivity of 77.77 % ± 2.72 %. The specificity and sensitivity calculated using the test data were 82.5 % ± 3.96 % and 73.18 % ± 7.55 % respectively.

S.No.	Type	Activations	Learnable Parameters	Total Learnable Parameters
1	Image Input	100 x 100 x 1	-	-
2	Dropout	100 x 100 x 1	-	-
3	Convolution	99 x 99 x 1	Weights 2 x 2 Bias 1 x 1	5
4	Convolution	98 x 98 x 1	Weights 2 x 2 Bias 1 x 1	5
5	Convolution	97 x 97 x 1	Weights 2 x 2 Bias 1 x 1	5
6	Convolution	96 x 96 x 1	Weight 2 x 2 Bias 1 x 1	5
7	Convolution	112 x 112 x 7	Weights 3 x 3 x 1 x 7 Bias 1 x 1 x 7	70
8	Convolution	112 x 112 x 9	Weights 1 x 1 x 7 x 9 Bias 1 x 1 x 9	72
9	Batch Normalization	112 x 112 x 9	Offset 1 x 1 x 9 Scale 1 x 1 x 9	18
10	Convolution	111 x 111 x 1	Weights 2 x 2 x 9 Bias 1x1	37
11	Convolution	111 x 111 x 9	Weights 1 x 1 x 9 Bias 1 x 1 x 9	18
12	Convolution	111 x 111 x 9	Weights 1 x 1 x 9 x 9 Bias 1 x 1 x 9	90
13	Convolution	127 x 127 x 7	Weights 3 x 3 x 9 x 7 Bias 1 x 1 x 7	574
14	Fully Connected	1 x 1 x 2	Weights 2 x 112903 Bias 2 x 1	225808
15	Softmax	1 x 1 x 2	-	-
16	Classification Output	-	-	-
	Total Learnables		226,707	

Table 3.9: Table summarizing the optimal CNN geometry for the classification of phase images of $CD4^+$ and $CD8^+$ T-cell acquired using the microscopic objective with 60X magnification.

The interesting aspect of this comparison is that the trend of increasing classification accuracy (Fig. 3.23) is evident. For the 20X configuration, the optimal CNN geometry resulted in a validation accuracy of 62.28 % ± 2.54 % and 59.43 % ± 1.99 % as the test accuracy, whereas this value increased for 60X optical configuration with 67.98 % ± 0.27 % validation accuracy and 69.31 % ± 2.04 % test accuracy. Whie considering the dataset accumulated using the 100X objective, the optimal CNN geometry resulted in a maximum classification accuracy with 78.91 % ± 1.57 % for validation set and 76.2 % ± 5.27 % for the test set. This shows that: increasing the magnification in a holographic system escalated its ability to classify the phase

images. However, with increasing magnification the throughput limit of the system deteriorates. Hence to keep an increased throughput limit and simultaneously improving the classification ability of the system, another deep learning based model was trained to transform the phase images acquired from 20X configuration to the phase images which may represent the acquisition from 100X optical configuration.

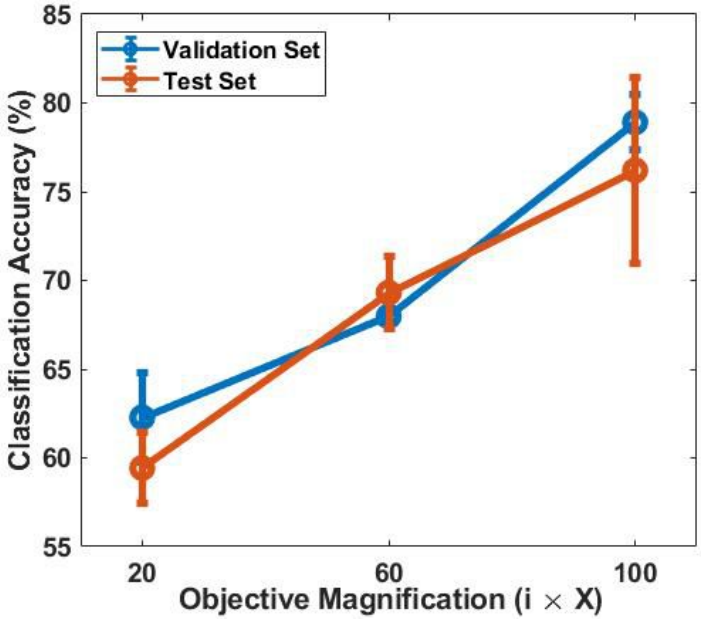

Figure 3.23: **Evaluation of classification accuracy for the three optical configurations**. Variation of classification accuracy evaluated for the three optical configurations. For the validation set, the values range from 62.28 % ± 2.54 % for 20X objective, 69.31 % ± 2.04 % for 60X objective and 78.91 % ± 1.57 % for 100X objective. For the test set, a classification accuracy of 59.43 % ± 1.99 % for 20X objective, 69.31 % ± 2.04 % for 60X objective and 76.2 % ± 5.27 % for 100X objective was obtained. Here the curve in blue represents values for validation set and curve in red represents the values for test set.

Image transformation and classification

The single image super resolution transformation was implemented on the phase images acquired using 20X configuration to convert them into the phase images acquired using 100X configuration. To achieve this the DL models were trained using the cycle GAN training method as explained before.

The trained generative models resulted in astounding transformations of the phase images. As shown in Fig. 3.24, the trained deep generative model transformed the phase images with a high speed of 6.89 milliseconds per transformation which

is limited by the processing power of the computer and can further be improved. In the mentioned figure, frame (I) represents all the phase images acquired using the 20X optical configuration where (a), (b) and (c) represent the phase images for $CD4^+$ T cells and (d), (e) and (f) represent the phase imaged for $CD8^+$ T cells. Frame (II) represents all the transformed phase images vis, (a) to (g), (b) to (h), (c) to (i), (d) to (j), (e) to (k) and (f) to (l). An interesting aspect which is visible on the transformed images is that the deep generative models automatically learned to draw an outline around the periphery of the cells. Additionally, it is quite evident from the images that as a result of transformation, the shape of these cells may also change. This can be explained by the mathematical inversion of the cycle GAN training module. The CNN learn to transform the images from one domain to another and simultaneously learn to inverse transform these images. This in turn results in a statistically coherent transformation model which works on increasing the size of images and redeveloping the shape of cells such that the inversion condition is met.

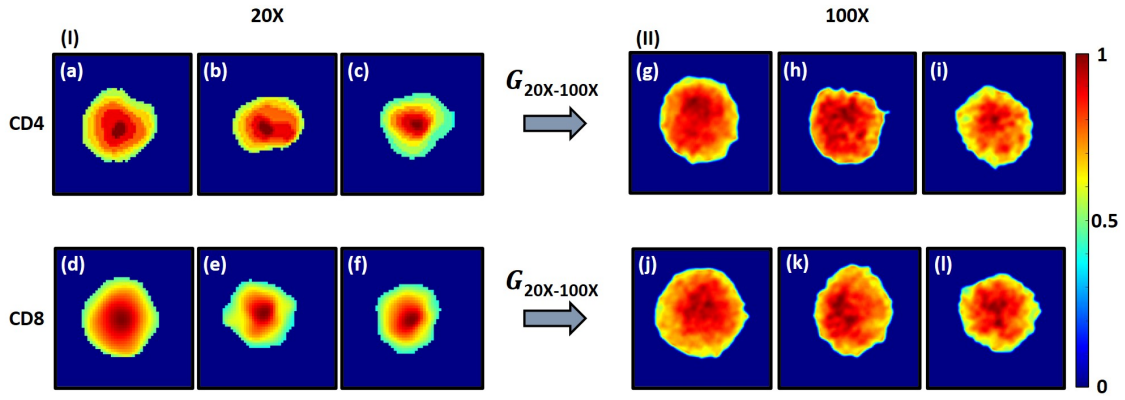

Figure 3.24: **Demonstration of image transformation using the trained deep generative model**. Part (I) of the figure represents the phase images acquired from the microscopic objective with 20X magnification with (a), (b), (c) representing the $CD4^+$ cells and (d), (e) and (f) representing the $CD8^+$ cells. Part (II) represents the transformed images of (I) where (g), (h), (i) represents $CD4^+$ cells and (j), (k), (l) represent $CD8^+$ cells.

After the transformation of the phase images, a classification was performed using the pre-trained optimal CNN geometry. This resulted in the classification accuracy of 45.93 % $\pm$ 2.46 % for the validation set and 47.91 % $\pm$ 2.05 % for the test set. These values were below the expectation and may be explained due to the presence of boundaries and overall shape orientation of the cells. Hence, to overcome

these problems, the previously trained networks were re-trained on the transformed images. This resulted in satisfactory results with sensitivity of 81.55 % ± 0.81 % and a specificity of 84.72 % ± 1.16 % for the validation set. For the test set, the sensitivity and specificity were evaluated as 79.77 % ± 3.32 % and 81.77 % ± 1.51 % respectively.

3.4 Conclusion

This chapter examines the applications of label free techniques of Raman spectroscopy and digital holographic microscopy to classify the cells of immune system. As an analytical method, the applications of artificial neural networks demonstrated an improvement in the classification accuracy of the two modalities.

This chapter starts with a brief introduction to immunology. Then it explores the applicability of label-free optical techniques for the classification of immune cells. Following this, the technique of Raman spectroscopy is discussed and wavelength modulated Raman spectroscopy (WMRS) is introduced. Subsequently, the label-free technique of digital holographic microscopy (DHM) is introduced with an insight into mathematical evaluation of the phase information extracted using this method. Following the introduction to the two techniques, two studies are explained demonstrating the applications of the mentioned techniques and their enhancement using artificial neural networks.

For the first study, the modalities of WMRS and DHM were explored to classify the morphologically similar granulocytes, namely neutrophils and eosinophils. The application of artificial neural networks rendered with an average improvement of 6.8% in the classification ability for both WMRS and DHM when compared to PCA. This approach further indicated that the combination of DHM with CNNs can be used as a highly efficient and stand-alone system to classify the cells of immune system. Additionally, with respect to the throughut rate, the combination of CNNs with DHM proved to be 2000 times faster than WMRS. Due to the high degree of differences in the biochemical properties of the two cell types, WMRS proved to be highly efficient with more than 98% classification accuracy.

The second study was directed to improve the throughput rate of the DHM

system to classify the T - cells subsets, namely $CD4^+$ and $CD8^+$ cells. To inspect the variation in classification accuracy of DHM - CNN combination with respect to the throughput rate, this study was conducted by changing the optical configuration of the DHM system; the microscopic objectives were changed between 20X, 60X and 100X magnifications. Owing to the variation in cellular phase image size, this study further applied particle swarm optimization to optimize the CNN geometry to classify the images acquired using each configuration. Further investigation was applied to improve the throughput rate of the DHM system by implementing a cycle GAN type training module to transform the images acquired using objective with 20X magnification into the images acquired using objective with 100X magnification. The results successfully demonstrated that by applying deep learning with DHM, it is possible to increase the classification accuracy and throughput rate of the system by increasing the resolution of images.

Indeed the application of deep learning with DHM exemplify a highly precise hemogram device. As per the requirement of deep learning based CNNs, the higher the number of points in the data, the higher the classification accuracy. This could have been achieved by acquiring more data for training. Hence, the results could have been better if the phase image datasets chosen for the training of the networks would have higher number of instances. Future works may focus on acquiring more data which may help in further improvement of results.

Relevant Publications

R. K. Gupta, M. Chen, G. P. A. Malcolm, N. Hempler, K. Dholakia, & S. J. Powis, "Label-free optical hemogram of granulocytes enhanced by artificial neural networks", In : Opt. Express 27, 10, 13706-13720 15p (2019).

Contributions

Author developed the numerical analysis procedures presented and acquired the imaging and spectroscopic data. Dr. Mingzhou Chen helped in accumulating the spectroscopic data. The cell isolations were performed by the author and flow cytometry was performed by Dr. Simon Powis. The projects were developed by Prof. Kishan Dholakia and Dr. Simon Powis.

S.No.	Type	Activations	Learnable Parameters	Total Learnable Parameters
1	Image Input	200 x 200 x 1	-	-
2	ReLU	200 x 200 x 1	-	-
3	Convolution	204 x 204 x 1	Weights 1 x 1 Bias 1 x 1	2
4	Convolution	205 x 205 x 2	Weights 2 x 2 x 1 x 2 Bias 1 x 1 x 2	10
5	Convolution	204 x 204 x 2	Weights 2 x 2 x 2 x2 Bias 1 x 1 x 2	18
6	Batch Normalization	204 x 204 x 2	Offset 1 x 1 x 2 Scale 1 x 1 x 2	4
7	Convolution	206 x 206 x 1	Weight 1 x 1 x 2 Bias 1 x 1	3
8	Convolution	202 x 202 x 11	Weights 5 x 5 x 1 x 11 Bias 1 x 1 x 11	286
9	Map Pooling	201 x 201 x 11	-	-
10	Convolution	203 x 203 x 1	Weights 1 x 1 x 11 Bias 1 x 1	12
11	Dropout	203 x 203 x 1	-	-
12	Convolution	205 x 205 x 1	Weights 1 x 1 Bias 1x1	2
13	Convolution	205 x 205 x 1	Weights 1 x 1 Bias 1 x 1	2
14	Convolution	205 x 205 x 1	Weights 1 x 1 Bias 1 x 1	2
15	Convolution	206 x 206 x 2	Weights 2 x 2 x 1 x 2 Bias 1 x 1 x 2	10
16	Convolution	208 x 208 x 1	Weights 1 x 1 x 2 Bias 1 x 1	3
17	Convolution	210 x 210 x 1	Weights 1 x 1 Bias 1 x 1	2
18	Convolution	212 x 212 x 1	Weights 1 x 1 Bias 1 x 1	2
19	Convolution	214 x 214 x 1	Weights 1 x 1 Bias 1 x 1	2
20	Fully Connected	1 x 1 x 35	Weights 35 x 45796 Bias 35 x 1	1,602,895
21	Tan Hyperbolic	1 1 35	-	-
22	Fully Connected	1 1 2	Weights 2 35 Bias 2 1	72
23	Softmax	1 1 2	-	-
24	Classification Output	-	-	-
	Total Learnables		1,603,327	

Table 3.10: Table summarizing the optimal CNN geometry for the classification of phase images of $CD4^+$ and $CD8^+$ T-cell acquired using the microscopic objective with 100X magnification.

Chapter 4

Application of deep learning to study speckle metrology

Parts of this chapter are adapted from the article "Deep learning enabled laser speckle wavemeter with a high dynamic range" [189], published in Laser Photonics & Reviews (2020).

4.1 Introduction

Previous chapter has demonstrated that supervised deep learning based approach using the convolutional neural networks (CNNs) proves to be a very useful tool for the classification of cell images with identical morphology. To demonstrate a broader applicability of deep learning, an application for the classification of speckle patterns as a function of incident laser wavelength was considered. This section will introduce the origins of speckle and its applicability for the calculation of incident laser wavelength, which can be improved by the application of deep learning.

The speckle pattern, a random granular intensity distribution pattern which is formed when a coherent light interacts with a rough surface or a medium with random refractive index variations, has been a part of research discussions since the late nineteenth century when Exner suggested the formation of speckling phenomenon. Following which it was also suggested by eminent scientists such as Von Laue, de Haas, Raman, Buchwald, and Ramachandran in the context of their respective research [190]. In late 1962, Langmuir [191] described it as a curious phenomenon of

presenting the pattern of random dark and light spots. Then in late 1963, Goodman introduced the laser sparkle patterns [192] and discussed their statistical properties as a part of a technical report. Nevertheless, the research efforts in late 1960s and early 1970s laid the foundations for speckle based optical metrology in the form of electronic speckle pattern interferometry.

Generally, the statistical properties of these patterns depend on the spatial and temporal characteristics of incident light and the microscopic properties of the random medium. The origin of speckle can be recognised as the "random" roughness of the scattering medium or the "random" variation of refractive index in a diffusive medium. For a rough scattering medium, the microscopic features of surface change the phase of observed field which interferes to form speckle. For a diffusive medium, the variation in refractive index leads to the variation in optical path length of various light rays which in turn results in an interference and formation of speckle. Evidently, speckle phenomenon appears frequently in optics and is generally deemed as noise for the given system. However, from a physical perspective, they have also been recognized to be rich in information [193].

Recently, speckle has also been recognized to measure wavelength with high resolution or broad operating range [194–201]. This has been possible because they overcome the limitations of standard diffraction grating based wavemeters by multiplexing spatial-to-spectral mapping in a compact system [202]. The critical step for the measurement of wavelength using speckle relies on the interpretation of variation in speckle patterns as a function of wavelength. A data-driven approach, where a statistical model can be trained over the speckle patterns for known wavelengths, seems to be a reasonable option. Previous techniques based upon the calculation of transmission matrix or principal component analysis presented limitations with respect to the sensitivity range of the camera or limited range of operation [203–205]. While these results seem to be impressive, a single step algorithm which may result in the wavelength measurements of high resolution and broadband range would be desirable.

The speckle patterns generated using a non-varying disordered medium present rich information about the wavelength/s of the incident light source [206]. These patterns also demonstrate a presence of noise which may be caused due to environmental

variations or instrumental fluctuations. The presence of these noisy factors invokes variation in the captured speckle patterns which makes them an ideal candidate for the training of a deep learning based model for classification type problems [207].

In this chapter, the application of deep learning for the development of speckle based wavemeter and spectrometer shall be explored. The chapter starts with a discussion of methods for data acquisition and optimization of a CNN architecture. Then the study for the development of a wavelength modulated speckle wavemeter is discussed which is followed by the study resulting in a speckle based wavemeter with high resolution and broadband range. Consequently, the study demonstrating the development of a speckle based binary spectrometer is presented.

4.2 Methods

4.2.1 Data Acquisition

The speckle data corresponding to the incident laser wavelength was accumulated using a 1.5 inch diameter, spectralon integrating sphere. The incident light source was chosen as an external cavity diode laser (Topica DL-100 / LD-0785-P220) which was stabilized to the ^{87}Rb D2 line ($F = 2 \rightarrow F' = 2 \times 3$ crossover at $\sim$ 780 nm) using saturated absorption spectroscopy and top-of-fringe locking. The wavelength of the incident light, coming from the laser, was controlled using an acousto-optic modulator (AOM) (Crystal Technologies 3110-120). This was achieved by passing it through the AOM in a cat-eye double pass configuration. The incident light was linearly polarised using a polarising beam splitting cube to avoid any variations that may occur due to polarisation [208]. To annul the effects of transverse mode profile of the beam [209–211],the light was coupled into an angle cleaved single-mode fibre (SMF) (Thorlabs P5-780M-FC-10). This SMF was then connected to the integrating sphere input-port via an FC/PC connector without collimation optics to produce a diverging fundamental Gaussian mode within the integrating sphere. A large optical path difference allowing high resolution for the system was achieved due to the highly Lambertian diffusive coating and multiple reflections inside the sphere. Passing through the sphere, the light then propagated for a fixed distance of 20 cm and hence impinged the CMOS camera (Mikrotron EoSens 4CXP). This distance

was chosen to prevent sub-Nyquist sampling and associated aliasing effect as the mean grain size of ∼3 pixels was achieved using this configuration.

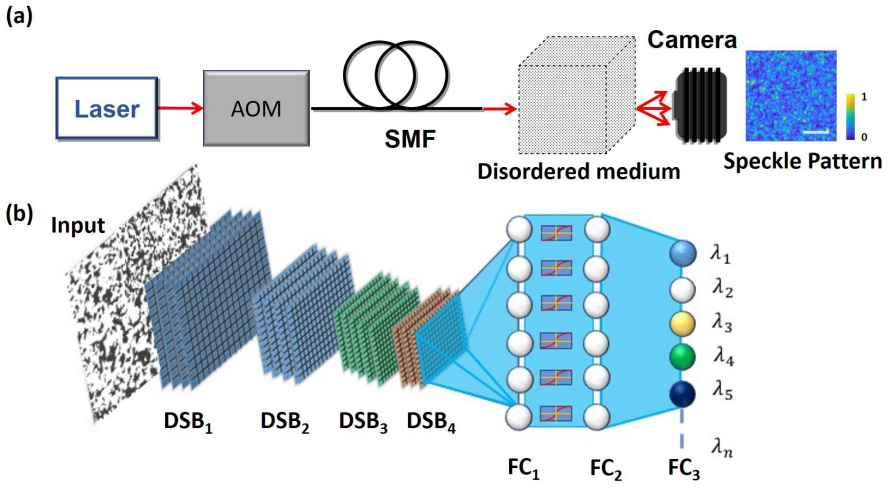

Figure 4.1: **Speckle wavemeter assembly and CNN geometry**. (a) The experimental assembly for a speckle wavemeter. The laser wavelength is set using an acousto-optic modulator (AOM) and injected into the disordered medium via a single mode fibre (SMF). The output speckle pattern is captured by the camera. (b) The convolutional neural network (CNN) used to classify the speckle images with respect to the incident laser wavelengths. The CNN consists of an input layer, multiple down-sampling blocks (DSB_i) and three fully connected layers (FC_i). Here λ_i denotes the output wavelength class. The white scale bar on the representative speckle pattern represents 224 μm, while the intensity is normalized as shown in the adjacent color bar.

4.2.2 Deep learning model geometry, training and calibration

To extract the wavelength dependence of the accumulated speckle images a supervised deep leaning based convolutional neural network (CNN) (depicted in Fig. 4.1(b)) was implemented. The CNN architecture was optimised for each experiment. Incidentally, the geometry of the convolution network was found to be optimum, for all the mentioned experiments in this chapter, with 4 down-sampling blocks (DSB). Each block consisted of 3 convolution layers with 30 filters. Each convolution layer was followed by a batch normalization layer and a ReLU activation function layer. Each DSB was connected with a max-pooling layer with a filter size of 2 px × 2px. The filter sizes of convolution layers vary as 5 px × 5 px, 4 px × 4 px and 3 px × 3px respectively with a stride and padding of 1 px × 1 px. The DSBs were followed

by two fully connected (FC) layers with leaky ReLU [212] as the activation function. Each FC layer was followed by a dropout layer [213]. These layers were then fully connected to the output layer having n neurons with softmax activation function, here n denotes the number of wavelength classes. In order to attain the maximum classification accuracy over the validation dataset, the CNN geometry was chosen after optimizing: the number of DSBs on the range 1 to 10; the number of convolution layers between 1 and 5; the filter sizes from 1 px $\times$ 1 px to 8 px $\times$ 8 px; and the number of neurons in FC layers from 8 to 512 by doubling the neurons at each step.

The CNN geometry was optimized by considering the training dataset for each experiment, this set comprised of speckle images corresponding to n different wavelengths. The training was implemented in Matlab 2018a over Nvidia Quadro P5000 GPU following a general training script mentioned in section A.1. To remove any intensity dependent fluctuations, all the speckle images were zero-center normalized. The CNN was trained to minimize the cross entropy cost function (also discussed in section 2.3.6):

$$\text{Cost} = -\frac{1}{k}\Sigma_x[y \times \log a + (1-y) \times \log(1-a)], \tag{4.1}$$

for 10 epochs in the mini batches of 128 images using an ADAM optimizer [214], where Σ_x represents training over all the input images x, k is the total number of training data points, y is the target output and a is the network output. Here y and a are the one hot vectors representing the category of the input image. Initial learning rate was set at 1×10^{-6} and L2 regularization at 2×10^{-4}. The training process was validated after every 100 iterations.

Using the common methods described above, the next sections (including a few modifications in the methods) will discuss about the development of laser speckle based wavemeters and spectrometer using various techniques of machine learning. Starting with an initial attempt to develop a wavelength modulated speckle wavemeter, the following section will discuss about the successful development of a wavemeter with a high dynamic range. Then a discussion about the development of a speckle spectrometer is presented.

4.3 Study to develop a wavelength modulated wavemeter

4.3.1 Methods

Data acquisition

The principle of using speckle patterns as the wavelength signature of light is very appealing. To achieve this, the optical system depicted in Figure 4.1 (a) was considered. As explained in section 4.2.1, the wavelength dependent speckle data was accumulated using the mentioned laser beam which was shown onto the acousto-optic modulator (AOM) and passed through a single mode fibre (SMF). The light coming from the SMF was passed into an integrating sphere which dispersed it to form a speckle pattern over the camera.

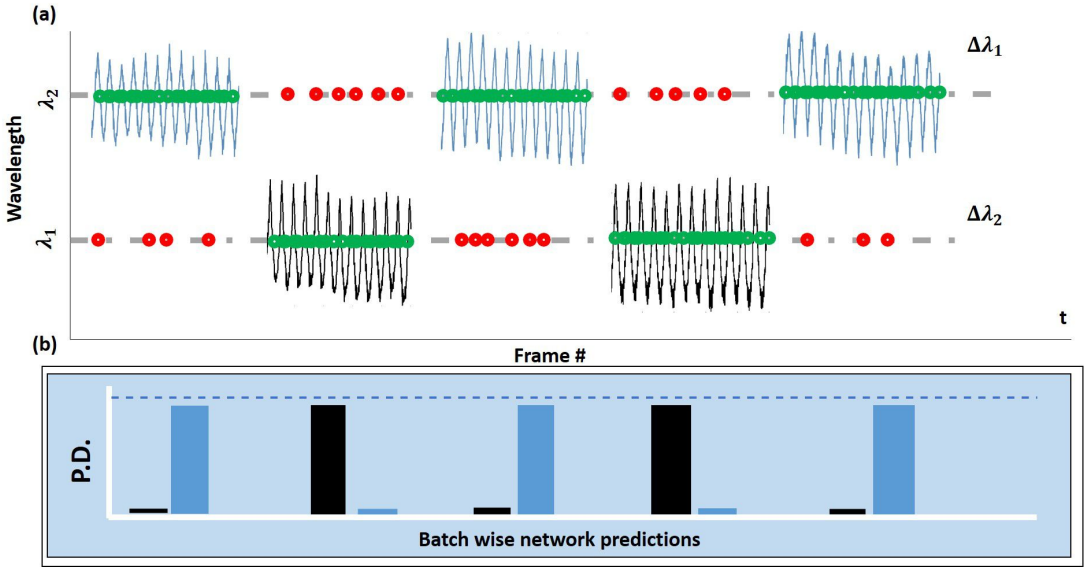

Figure 4.2: **Binary speckle wavemeter based on wavelength modulation.** (a) Real-time wavemeter for wavelengths at λ_1 and λ_2 with the corresponding modulations at $\Delta\lambda_1$ and $\Delta\lambda_2$. (b) Schematic of real-time softmax prediction by the CNN for the speckle patterns belonging to a given waveform.

With an initial understanding [205], the experiments were designed such that the speckle patterns were accumulated by modulating the incident laser wavelength. This resulted in the formation of speckle waveforms having a central wavelength with a peak to peak modulation. A model depiction of the proposed wavemeter is displayed in Fig. 4.2. Part (a) of this figure exhibits the working principal of the

binary wavemeter. Here, the real-time working of the wavemeter is displayed with central wavelengths at λ_1 (waveform colored in black) and λ_2 (waveform colored in blue) with corresponding modulations of $\Delta\lambda_1$ and $\Delta\lambda_2$. The dots present over the timeline represent the prediction of central wavelength made by the trained CNN. Here, green color represents the correct prediction whereas red color represent the wrong prediction. Fig. 4.2 (b) indicates the schematic of real-time softmax output from the trained CNN for a given category which eventually leads to a prediction.

During the data accumulation, a continuous train of 160,000 128×128 pixel speckle images for each waveform at a frame rate of 1 kHz with an exposure of 998 μs were recorded. Variable modulations with a period of 0.2 seconds were applied on the input light beam.

Optimization of the Deep learning model

A CNN was employed to classify the speckle waveforms of a central wavelength with modulation. The geometry of CNN was optimized by following the methodology explained in section 4.2.2. With respect to the performance gained over the validation dataset, the convolution network with 4 DSBs was found to be optimal whereas 256 neurons were identified as the best combination for the number of neurons in both FC layers.

4.3.2 Results

A few initial studies were conducted to optimize and train a CNN geometry to classify the speckle waveforms. The modulation of wavelength in the train of speckle patterns was identified by applying PCA (following the general script in section A.4). Figure 4.3, represents the first principal component (PC) calculated using the train of speckle images. As shown in the figure, the first PC signifies wavelength modulation for the triangular and sinusoidal waveforms. For this experiment, the speckle patterns were accumulated using a triangular wave modulation at the central wavelength of 30 fm with a peak to peak modulation of 20 fm (Fig. 4.3 (a)), another traingular modulation at a central wavelength of 52 fm with a peak to peak modulation of 20 fm (Fig. 4.3 (b)) and a sinusoidal modulation at the central wavelength of 30 fm with a peak to peak modulation of 20 fm (Fig. 4.3 (c)).

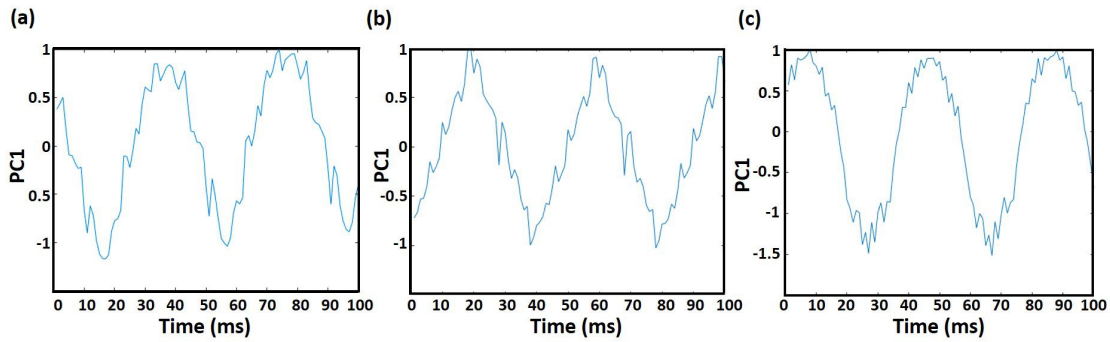

Figure 4.3: **First principal component of the wavelength modulated speckle waveforms** (a) First principal component (PC1) of the speckle waveform with triangular modulation of 20 fm with a central wavelength at 30 fm from ^{87}Rb D2 line reference. (b) PC1 of speckle waveform with central wavelength at 52 fm from reference with triangular wavelength modulation of 20 fm. (c) PC1 of speckle wavefrom with sinusoidal wavelength modulation of 20 fm at the central wavelength of 30 fm from reference.

	Act. Wave 1	**Act. Wave 2**	**Act. Wave 3**
Pred. Wave1	*30174*	2487	29586
Pred. Wave2	1450	*29118*	2370
Pred. Wave3	376	395	*44*

Table 4.1: Confusion matrix to classify the speckle images corresponding to three waveforms.

The CNN was trained on the mentioned dataset to classify the images following the routine mentioned in section 4.2.2. After the training, the trained CNN geometry was tested over the speckle dataset comprising 32000 images for each waveform. As the training and test sets consisted of images corresponding to same central wavelengths (first and third case), to no surprise, the CNN classified all the images with a low classification accuracy of 61.8%. Interestingly, as can be understood from the confusion matrix (Tab. 4.1), the trained CNN mixed up the predictions between the triangular and sinusoidal waveform for the same central wavelength and evidently classified 92.4% speckle images from the sinusoidal group as the images from triangular modulation group.

To have a better understanding, outputs from the softmax layer of the trained CNN were plotted as a histogram plot for three waveforms. Figure 4.4 shows the predictions made by the CNN when the input data was changed between waveform one, two and three. As can be appreciated from Fig. 4.4 (a), the CNN confused the prediction of sinusoidal waveform as the traingular waveform, whereas the second

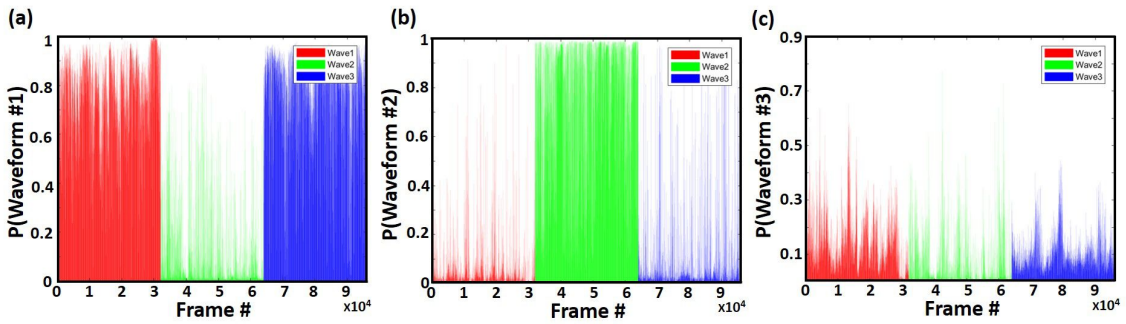

Figure 4.4: **Histogram of softmax output from the trained CNN.** Probability of predicting the speckle image to be belonging to (a) first (b) second and (c) third waveform by the trained CNN when the input images were varied for each class in the intervals of 32,000 images. The red color denotes images corresponding to first waveform, green color denotes speckle images from second waveform and blue color denotes speckle images from third waveform.

waveform was correctly predicted (Fig. 4.4 (b)). The sinusoidal waveform was instead confused (Fig. 4.4 (c)) among the three classes with only 0.13% correct predictions.

To consider the temporal stability of the CNN enabled wavemeter, previously trained CNN was further tested over the data accumulated over the first two waveforms accumulated after two days. This dataset consisted of a total of 48000 images for each waveform. The CNN predicted each waveform with a classification accuracy of 86.4%. As can be inferred from table 4.2, first waveform was correctly predicted with an accuracy of 99.43% whereas the second waveform was predicted with an accuracy of 73.3%. The CNN also predicted the two waveforms as the sinusiodal waveform for 1.05% times of total images.

	Act. Wave 1	**Act. Wave 2**	**Act. Wave 3**
Pred. Wave1	*47727*	12062	0
Pred. Wave2	4	*35197*	0
Pred. Wave3	269	741	0

Table 4.2: Confusion matrix evaluated for the speckle images corresponding to waveform one and two for the data accumulated after two days.

The above mentioned results signify that the CNN once trained can be implemented with high accuracy. The CNN, after training over the three waveforms, got mixed up while predicting between sinusoidal and triangular waveforms for the same central wavelength and modulation whereby the triangular waveform was favored. The results strongly depicted that the CNN classified speckle patterns with respect

	Act. Wave 1	**Act. Wave 2**	**Act. Wave 3**	**Act. Wave 4**	**Act. Wave 5**
Pred. Wave 1	*31996*	19013	50	0	0
Pred. Wave 2	4	*10288*	1119	11	0
Pred. Wave3	0	2699	*30505*	400	0
Pred. Wave 4	0	0	326	*30206*	25
Pred. Wave 5	0	0	0	1383	*31975*

Table 4.3: Confusion matrix of test dataset for the speckle images corresponding to five different waveforms.

to the incident wavelength and generalized the waveform type to the simplest mathematical description. Additionally, the CNN proved to be considerably stable while making predictions on a dataset accumulated after two days.

Limitations

After achieving encouraging results mentioned above, the next set of experiments were designed to train the optimized CNN geometry with more than two speckle waveforms – five distinct classes.

The speckle waveforms with a constant peak to peak wavelength modulations of 8 fm were accumulated at the central wavelengths ($\Delta\lambda = \lambda - \lambda_0$, here λ is the modulated wavelength and λ_0 is the reference ^{87}Rb D2 line) on -20 fm, -10 fm, 0 fm, 10 fm and 20 fm. The CNN was trained and validated over 128,000 images per class and tested over 32,000 images for each class. After the training process, the CNN resulted in a very low classification accuracy for the test dataset.

Table 4.3 represents the confusion matrix for the performance of the trained CNN over the five speckle waveforms. As can be calculated, the total classification accuracy of 84.54% was obtained. The network classified waveforms two, three and four as being a part of the neighbouring group for 67.85%, 4.67% and 5.60% times respectively. The highest number of wrong predictions were made for waveform 2 with the classification accuracy of 32.15%.

To firmly understand low classification efficiency of the CNN, another analysis was conducted over the output (FC_1) of the trained CNN. The analysis was done by considering a speckle image from the test dataset and passing it through the CNN. Parsing the image through the CNN allows systematic down-sampling of images from the size of 128 $\times$ 128 px to 256 spatial features. This output from the FC_1 was plotted and Fourier transform was applied on it. As these neurons represent the

spatial distribution of the speckle patterns, the Fourier transform would represent the temporal variations.

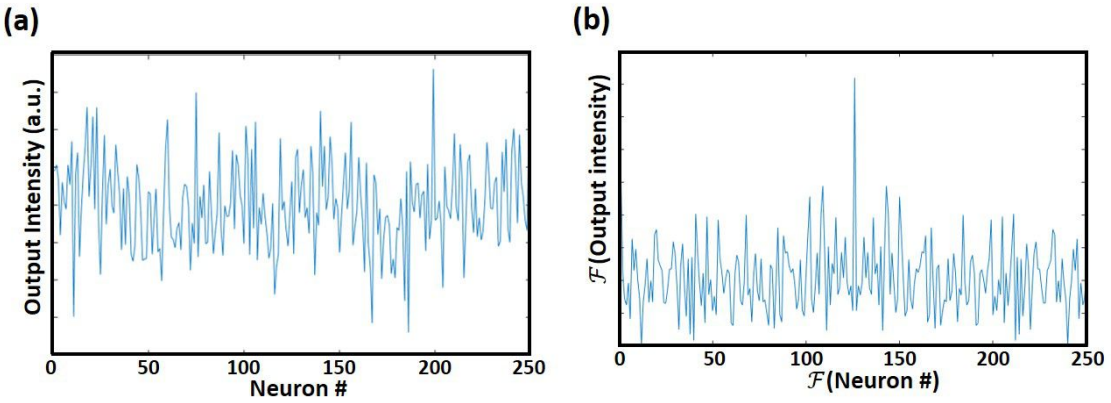

Figure 4.5: **Analysis of the output of CNN when trained over wavelength modulated data.** (a) Noisy output from the FC_1 of trained CNN hiding an approximated waveform. (b) Fourier transform of (a) showing the presence of a single peak.

Figure 4.5 represents the variations present in the output of FC_1 layer and its Fourier transform. Here, part (a) clearly illustrates a noisy varying output which hides an approximated waveform. Part (b) of the same figure represents the Fourier transform of (a) which clearly portrays the presence of a single peak. This peak directly points in the direction that during the training phase, the CNN learned a generalized wavelength distribution of the speckle waveforms instead of individual patterns. It approximated the waveforms with the presence of huge noise. This resulted in miss-classification of various waveforms discussed above.

The above analysis pointed in the direction that the CNN could be applied to enable a speckle wavemeter when the speckle patterns were accumulated without modulating the incident laser wavelength. Considering this hypothesis, a deep learning enabled speckle wavemeter was constructed with a high dynamic range. This is explained in the next section.

4.4 Deep learning enabled speckle wavemeter with high resolution and broadband range

This section provides the details about the successful study to develop a speckle based wavemeter using an integrated sphere. The study explored the noise rejection capabilities of the CNNs and identified the resolution of two attometres as the limit of the experimental instrument. The generalization capabilities of deep learning were also explored by testing the wavemeter with speckle patterns generated using a ground glass.

4.4.1 Methods

Data Acquisition and Deep learning geometry

The principle of approach for measuring wavelength is outlined in Fig. 4.1. As mentioned in section 4.2.1, the speckle patterns were produced by scattering laser light from a disordered medium, and were recorded on a camera (Fig. 4.1 (a)). Unless stated otherwise, a tunable diode laser, wavelength-locked to a rubidium reference ($\sim$ 780 nm), was used as the source of laser light, an acousto-optic modulator to apply controlled wavelength variations, and an integrating sphere to scatter light.

A continuous train of 10,000 128 $\times$ 128 pixel speckle images corresponding to each wavelength were recorded. The frame rate of the recording was set at 1kHz with an exposure time of 998 μs. The whole data acquisition corresponding to each wavelength was completed in 10 seconds with a typical time difference of 0.5 seconds between different wavelength classes. The dataset was randomly sampled into 70% training, 15% validation and 15% testing images corresponding to each wavelength.

A CNN was implemented to extract the wavelength dependence of the speckle images (depicted in Fig. 4.1 (b)). The optimal geometry of the CNN was identified using the optimization routine mentioned in section 4.2.2. This resulted in a CNN geometry with 4 DSB's and 128 neurons in both the fully connected layers.

Broader range of optical spectrum, using additional lasers, were also considered to test the wavemeter capabilities. Additional lasers at wavelengths of 488 nm (M-Squared frequency-doubled SolsTis Ti:Sapphire), 532 nm (Oxxius single-longitudinal mode diode-pumped solid state laser), 671 nm (Thorlabs HL6756MG Diode Laser)

and 976 nm (M-Squared SolsTis Ti:Sapphire) were used. Generalization capabilities of the deep learning enabled wavemeter were tested by replacing the integrating sphere with a ground glass diffuser (Thorlabs ED1-S20).

t-SNE analysis

During the training process, the CNN learns to generalize the wavelength-dependent variations of the speckle patterns and hence classify them. The CNN geometry can be understood as a composition of two ANNs – a down-sampling convolution network (input layer to FC_1) and a classification network (FC_2 and FC_3). The convolution network basically works to down-sample the input 2D images into a 1D descriptor vector by filtering the irrelevant / noisy features. Following the down-sampling from convolution network, the classification network classifies the input 1D descriptor vectors. Hence, for the case of a CNN trained to classify speckle images (128×128 px) with respect to the incident laser wavelength, the convolution network would produce a 1D descriptor vector (128 px) representing the wavelength of input speckle image at a given time instant. In principle, the speckle field remains constant for a given time instant which means that the so produced 1D descriptor vector would be free from any environmental fluctuations. Hence, the convolution output of a trained CNN can be directly considered to classify a broader range of optical spectrum.

The segmentation capabilities of the convolution network for multiple datasets were analysed using t-SNE, which has been explained in section 2.3.5. This method reduced the 128 dimensional 1D vectors into a 2 dimensional latent space by preserving their metric properties. In this study, a perplexity of 30 was considered to apply t-SNE (implemented using MATLAB 2018b).

4.4.2 Results

CNN optimization

The first step for the analysis of speckle patterns was to optimize and calibrate a CNN geometry. This was achieved by considering a dataset in which speckle patterns were generated by varying the incident laser in 2 fm steps over a range of 60 fm. This dataset consisted of 10,000 images which were randomly divided into training, validation and test sets in the ratio of 70%, 15% and 15% respectively.

The training of the CNN is conducted by considering a batch of randomly sampled images from each class of the training set. During the process, an error representing the wrong classification by the network is computed for each batch and is backpropagated. This backpropagation of error through the network trains it to identify the features with respect to each class. The classification efficiency of the trained network is then tested over the validation set which provides a confirmation that the CNN geometry and training are optimal.

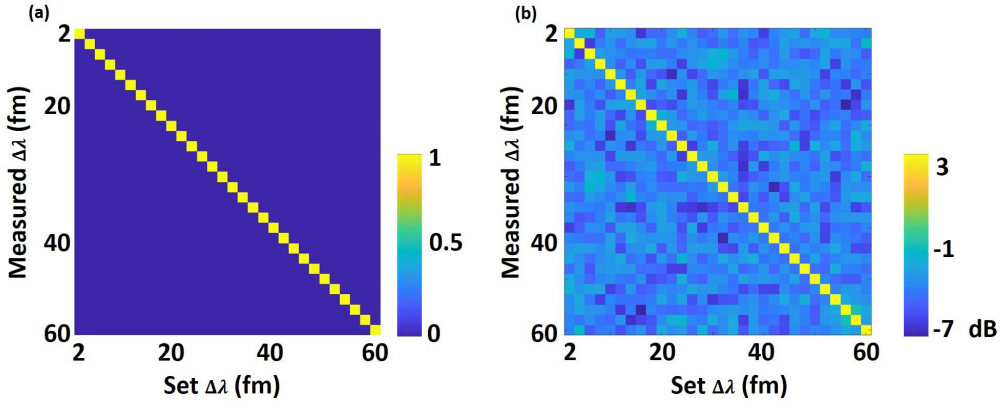

Figure 4.6: **Demonstration of high-accuracy discrimination of femtometre-resolved wavelength changes**. (a) Confusion matrix for the output of the CNN in classifying speckle patterns corresponding to wavelength separations on the femtometre-scale. The color bar represents the normalised value between 0 and 1. (b) Confusion matrix plotted on $\log_{10}$ scale. The color bar represents the decibel values.

Following the complete training and optimization of the CNN, it was tested against the novel test set of images from each class. This set consisted of a total of 45,000 images across all wavelengths. The optimization routine being precise, it resulted in a very accurate one-hot classification with 100% classification accuracy. However, to understand and quantify the classification errors of the CNN, a probabilistic classification error (PCE) was calculated by considering the softmax output of the FC_3 layer for the test dataset. This error was calculated by taking the sum of all incorrect classification values for each image and then taking the average of this summed value over all the images. This resulted in a PCE of 2.2×10^{-6}. A demonstration of accurate classification is presented in the form of a confusion matrix in Fig. 4.6 (a), and to further understand the PCE a $\log_{10}$ of the same matrix is calculated and present in Fig. 4.6 (b).

Deep learning based CNN when operated on speckle patterns prove to be excel-

lent candidate for wavelength classification. The inherent variability in the speckle patterns in turn provide to be the ideal candidate for the training of CNNs. The combination of speckle with CNNs achieves a remarkable classification accuracy since speckle patterns represent an ideal candidate for the training of the CNN. If the disordered medium and the laser wavelength are kept constant then, ideally, the resulting speckle pattern should not change. However, the environmental fluctuations or fluctuations due to instrumental circuitry cause the speckle patterns to change with time. Therefore, the next step was to understand this variability in the input speckle patterns and the processed output of the CNN.

CNN noise rejection capabilities

To understand on how the trained CNN allowed for such a small PCE and 100% classification accuracy, an analysis was implemented on the raw speckle images and their processed 1D vectors. In order to identify the presence of variance, PCA was implemented on both the input and output of the CNN.

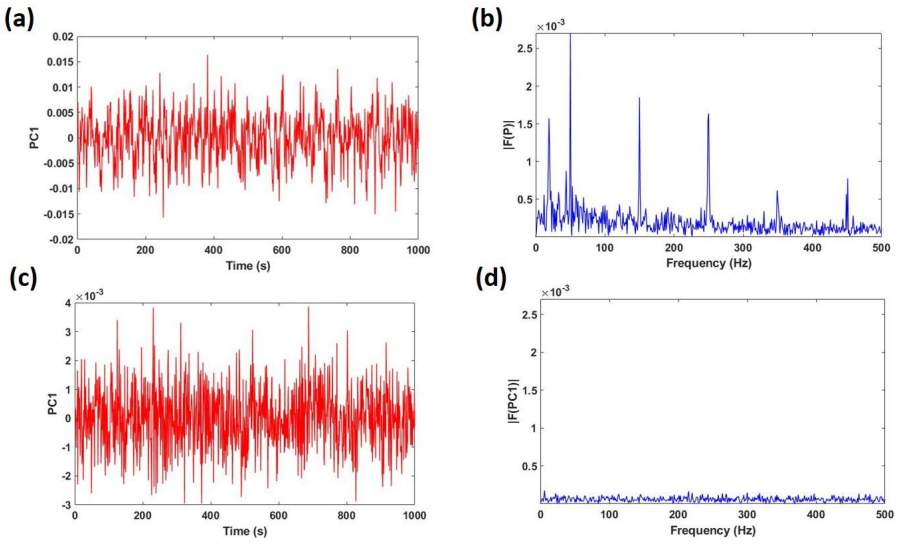

Figure 4.7: **CNN-enabled noise rejection**. The CNN learns, through training, to reject instrumental noise from the wavelength measurement. For an example dataset comprising 1000 images sampled over 1 second at a fixed wavelength, (a) shows the PC1 of the input raw speckle images. (b) Fourier transform of (a) highlighting the presence of noisy variations; (c) PC1 of the output from the trained CNN for full train of speckle images, (d) the Fourier transform of (c) highlight the absence of any variations present in the output of the CNN. Here, PC1 denotes the first principal component.

As shown in Fig 4.7, a train of speckle patterns accumulated over 1 second was

considered. To identify the maximal variation, the first principle component of both the raw images and their 1D vectors (Fig 4.7 (a) and (c)) were further analysed by estimating the smallest detectable shift in wavelength. This was estimated as three times the standard deviation (σ) from the mean position. For the raw speckle images, the 3σ value was evaluated as 0.014, whereas a value of 0.003 was calculated for the first principal component of the output 1D vectors of the trained CNN. An improvement by the factor of 4.66 quantitatively demonstrated the suppression of noise in the speckle images when processed by the CNN. Further analysis of the first principal components was implemented by the virtue of Fourier transform (Fig. 4.7 (b) and (d)). As can be identified from the figure, the PC1 of raw speckle images at a fixed wavelength show several periodic noise components. However, the PC1 of the output of convolution network (FC_1) for these speckle images showed the absence of any temporal noise. Analogous to a previous study [104], the CNN once trained to classify the speckle images with respect to wavelengths can filter the input speckle images of any environmental or instrumental circuitry noise. Hence the output 1D vector from the CNN can be thought of as representing a single wavelength.

Figure 4.7, further provides with an explanation that the sturcture of CNN and backpropagation training drives it to learn and filter the input images. This allows the CNN to filter the input speckle images, with environmental or instrumental fluctuations, into noise-less output vectors.

Attometre precision

The results presented above in combination with the universal functional approximator capabilities [69] of the ANNs orient in the direction that the CNNs can be further trained to recognise the incident laser wavelength with a precision below the instrumental circuitry noise. Hence as the next step of analysis, the speckle patterns were accumulated with attometre scale deviations.

Specifically, the speckle patterns were captured with an deviation of two attometres by detuning the acousto-optic modulator across five distinct wavelengths. To ensure that the wavelength drifts between measurements could be smaller than the drift within a single measurement period, the the dataset was accumulated in 10 seconds for a single wavelength with a typical time difference of 0.5 seconds between

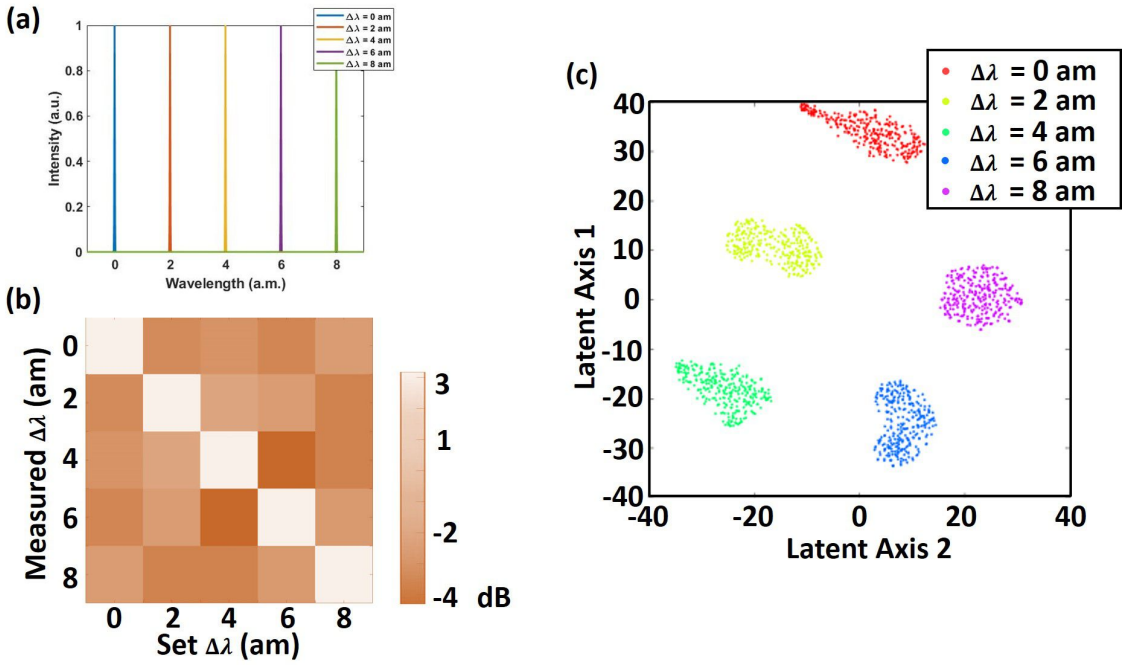

Figure 4.8: **CNN classification and segmentation capabilities of attometre-resolved speckle data**. (a) Visualization of an attometre-resolved wavemeter. (b) Confusion matrix on the log_{10} scale depicting the classification abilities of the CNN for wavelengths separated by 2 am. Here $\Delta\lambda = 0$ am depicts a detuning of $\Delta\lambda = 30.652$ fm from the rubidium crossover and the other values are relative to it. The color bar represents the decibel values. (c) t-SNE visualization for the output of the FC_1 from the CNN trained over femtometre-resolved speckle data, applied to speckle data separated by two attometres. The speckle images at each wavelength form a distinct cluster, showing that the CNN can be retrained simply by using a single speckle image at a known wavelength.

different wavelength classes.

Since the dataset was changed and the accumulated speckle patterns show completely different variation across the image, the classification abilities of the CNN were re-tuned. This re-tuning was implemented by the virtue of transfer learning, whereby the number of neurons in the output layer (FC_3) were changed. The training process was implemented by considering 7000 images per wavelength for training/validation and 3000 images for testing. Again, as anticipated, the CNN resulted in a 100% one-hot classification accuracy with a PCE of 3.8×10^{-5}. The accuracy of classification is emphasised in Fig. 4.8 (b) where the confusion matrix on $\log_{10}$ scale is presented. These results confirm that the CNN can be trained to classify the speckle patterns with respect to the wavelength with a precision of as low as 2 am. This precision of 2 am is **not** a fundamental limit but is limited by

the precision of AOM which is used to control the wavelength deviations.

Further, as demonstrated in Fig. 4.7, the output of a trained CNN represents a single wavelength. Hence to eliminate the process of retraining the CNN the speckle images were also processed using the CNN trained to classify the fm-resolved data. To proceed with this, the output 1D vectors of the CNN were analysed using t-SNE and the segmentation capabilities were visualised. Figure 4.8 (c) demonstrates the clustering of speckle images with respect to the incident light wavelength. In this figure, the output of the FC_1 layer are downsampled into a two dimensional latent vector space such that the metric properties are conserved. Using this method, further training of the CNN is not required to achieve attometre-scale precision and only a single image (in contrast to 7000 images) can be used for further classification. The resolving power of a wavemeter is defined as:

$$R = \frac{\lambda_0}{\delta\lambda} \tag{4.2}$$

here λ_0 is the absolute wavelength and $\delta\lambda$ is the minimum detected deviation from it. Using Eq. 4.2, the resolving power for the CNN based wavemeter can be calculated as $> 10^{11}$ for a central wavelength at 780 nm with a least deviation of 2 attometre.

This demonstration of high resolving power and attometre scale precision to classify the speckle patterns with respect to incident wavelength can be directly attributed to the automated noise rejection capabilities of the CNN.

Broadband operation and high dynamic range capabilities

After detecting the limit of precision for the CNN based wavemeter, the next step was to identify the range for which this could work. Hence to test the broadband capabilities of the CNN, the speckle patterns were accumulated over two wavelength ranges: from 770 nm to 790 nm in 5 nm increments and separately at 488 nm, 532 nm, 785 nm and 976 nm.

This data was analysed using the fm-trained CNN in combination with t-SNE. As demonstrated in Fig. 4.9 (a) and (b), t-SNE and CNN based analysis resulted in independent clustering of speckle patterns with respect to individual wavelength. These results show that without retraining the CNN, it is possible to classify the speckle patterns by just considering a single known image per cluster.

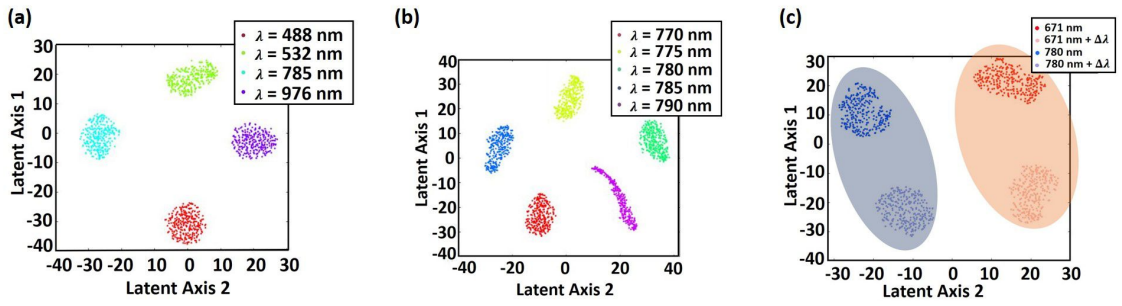

Figure 4.9: **Segmentation capabilities of the CNN over a broadband range of data**. (a) t-SNE scatter plot of the output of 1D descriptor vector for the wavelength deviations corresponding to 488 nm, 532 nm, 785 nm and 976 nm. (b) t-SNE scatter plot of the 1D descriptor vector corresponding to 770 nm, 775 nm, 780 nm, 785 nm and 790 nm. (b) 488 nm, 532 nm, 785 nm and 976 nm. (c) t-SNE scatter plot for 1D descriptor vectors of speckle patterns at 780 nm, 780 nm + $\Delta\lambda$ (Highlighted in blue), 671 nm and 671 nm + $\Delta\lambda$ (Highlighted in orange). Here $\Delta\lambda = 2$ am.

The fractional bandwidth of the wavemeter is:

$$B = \frac{\lambda_{\max} - \lambda_{\min}}{\frac{1}{2}(\lambda_{\max} + \lambda_{\min})} \tag{4.3}$$

here $\lambda_{\max}$ is the maximum detected wavelength and $\lambda_{\min}$ is the minimum detected wavelength in the broadband operation range. Using the Eq. 4.3, fractional bandwidth can be evalued to be $B = 0.66$ for the speckle wavemeter presented here with $\lambda_{\min} = 488$ nm and $\lambda_{\max} = 976$ nm.

Dynamic range of a wavemeter can be defined as the product $B \times R$. This results in the high dynamic range capabilities of the CNN based speckle wavemeter which can identify the differences in speckle patterns with a precision of a few attometres over a range of 100s of nanometres, resulting in a dynamic range value of 3.25×10^{11}. To further enunciate the high dynamic range of the wavemeter, another set of data was accumulated and analysed using the combination of CNN and t-SNE. The data was accumulated using diode lasers locked to the D2 lines of ^{87}Rb ($\sim$ 780 nm) and ^{7}Li ($\sim$ 671 nm). For each laser, the AOM was used to generate two set of speckle patterns, with a wavelength separation of 2 am between the two sets. This resulted in a broadband wavelength measurement with a precision of 2 am. As can be inferred from Fig. 4.9 (c), t-SNE plot shows the presence of four distinct clusters corresponding to the four mentioned wavelengths.

These results in combination with interpolative estimation show that once the CNN is trained, it can be employed for wavelength based classification of speckle images in the range between 488 nm to 976 nm regardless of the variations in the incident laser wavelength.

Generalization capabilities of the CNN

After testing the dynamic range capabilities of the CNN, the next step was to test its generalization capabilities. To investigate this, the trained CNN was tested against the data accumulated using a ground glass assembly instead of the integrated sphere assembly. As a known fact that the speckle patterns represent the spatial correlation

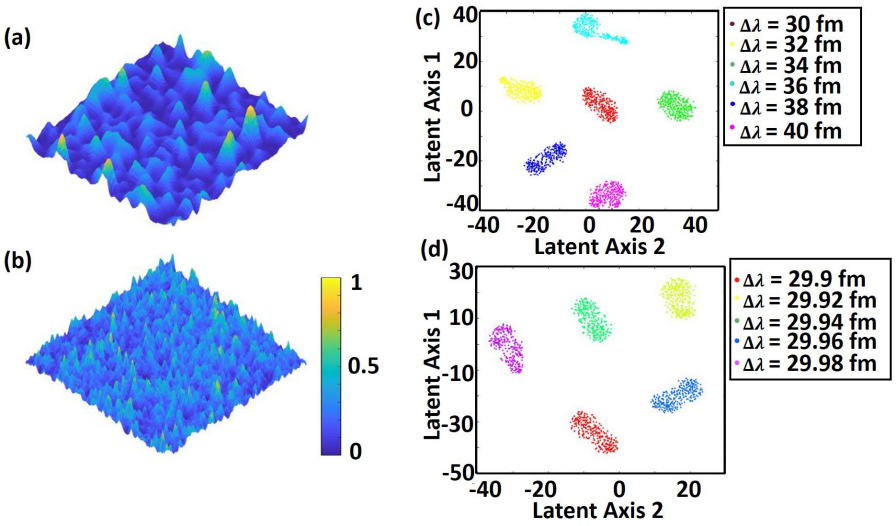

Figure 4.10: **Generalising wavelength classification to a different scattering medium**. Speckle pattern generated using an (a) ground glass and (b) integrated sphere assembly. The color bar represents normalised intensity. Segmentation results using ground glass assembly for (c) femtometre-resolved and (d) attometre-resolved incident laser wavelength modulations.

function of a given disordered medium and the incident light beam; meaning that for a different scattering medium the generated speckle patterns would maintain the wavelength dependent variations [215]. Hence, with respect to the generalization capability, an artificial neural network trained to decrrelate the speckle patterns with respect to the wavelength would also be able to segment the speckle images generated from any random disordered medium. Hence, another speckle dataset was accumulated using a ground glass diffuser (Thorlabs ED1-S20). Fig. 4.10 (a) and Fig. 4.10 (b), demonstrate the speckle patterns generated from the integrating

sphere and the ground glass. It is clearly visible that the two patterns display completely different characteristic features.

To appreciate conservation of dynamic range and generalization capabilities, two sub-datasets were accumulated. For the first dataset, the speckle images were accumulated by varying the incident wavelength with an increment of 2 fm and for the second set speckle patterns were imaged with an increment of 20 am in the incident light wavelength. The segmentation capability of the CNN, trained over femtometre resolved data, was tested by processing the speckle images generated from each class. The output from FC_1 was analysed using t-SNE and the results are presented in Fig. 4.10. As shown in Fig. 4.10 (c) and (d), the CNN segments and clusters each of the speckle images into their individual class. The results confirm that once trained, the CNN can generalise the wavlength dependent variations in the speckle patterns and when combined with a clustering algorithm can cluster them with respect tot he incident light wavelength, even when the images display completely different spatial variations.

The results presented for the high dynamic range (Fig. 4.9) and generalization capabilities (Fig. 4.10) can also be identified as a corollary from the noise cancellation property demonstrated in Fig. 4.7 combined with universal function approximator property of the ANNs. These prove that once a CNN model is optimized and trained, it can be implemented, to segment the wavelength-dependent speckle patterns generated from any disordered medium and with any wavelength deviation.

The speckle patterns exhibit the complete information about the incident light. Hence, the possibilities of detecting multiple wavelengths were explored. The next section provides the details of the study conducted to develop a speckle based binary spectrometer by using the integrating sphere assembly.

4.5 Development of a speckle spectrometer

This section provides the details of the study conducted to develop a speckle based spectrometer. The data was acquired by using two polarised light sources with different wavelengths. The fully developed speckle pattern over the camera was

considered for analysis by employing feedforward neural networks.

4.5.1 Methods

Data Acquisition

The speckle data was acquired in two steps. First step was to accumulate the speckle patterns with respect to individual wavelengths for 30 wavelength classes each using a single laser source by following the procedure mentioned in section 4.2.1. The second step was to accumulate the spectra developed using the two sources as demonstrated in Fig. 4.12 where the laser source was split into co-polarised components each with a separate AOM controlling the wavelength deviation. The spectral components considered are summarised in table 4.4. The speckle patterns coming out of the sphere can be linearly decorrelated into the constituting spectral components. The spectral components being developed as a result of equal incident laser power may contribute equally to the probabilistic modeling of the binary speckle pattern.

S No	**Source 1** ($\Delta\lambda$)	**Source 2** ($\Delta\lambda$)
1	10 fm	30 fm
2	20 fm	30 fm
3	24 fm	30 fm
4	28 fm	30 fm
5	30 fm	40 fm
6	40 fm	50 fm

Table 4.4: Table summarizing the accumulated wavelengths using two laser sources for forming the binary spectral speckle patterns

A continuous train of 128×128 pixel speckle images depicting the binary spectra were captured using the CMOS camera (Mikrotron EoSens 4CXP). The frame rate for the recording was set at 1000 frames per second with an exposure time of 998 μs.

Artificial neural network optimization

To decorrelate the speckle patterns with respect to the given spectra, a stacking of MLP models was performed. As an initial step, first MLP was trained to classify the speckle patterns with respect to individual wavelengths from one laser source. After this training was concluded the speckle patterns formed as a result of spectral

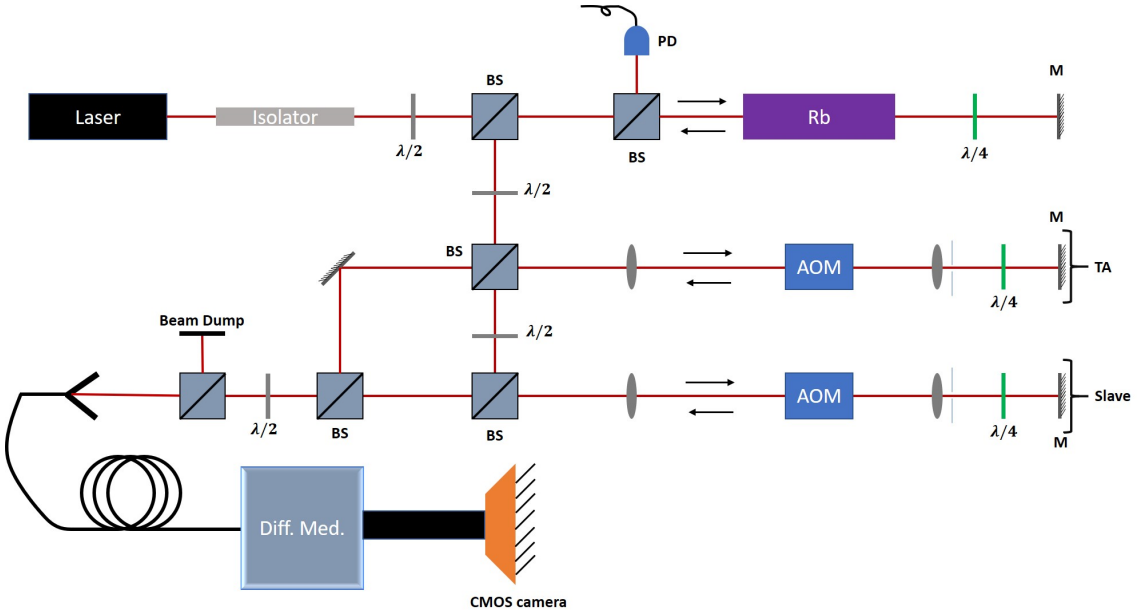

Figure 4.11: **Ray diagram of the setup used for accumulating the speckle based binary spectral data**. Here, $\lambda/2$ and $\lambda/4$ are half and quarter waveplates. BS is beam splitter, PD is photo diode, M is mirror, AOM is the acousto optic modulator. Diff. Med. is the diffusive medium.

input, were processed from the trained MLP to produce one dimensional probabilistic vectors. These probabilistic vectors were then supplied to second MLP which was then trained on a regression mechanism to identify the spectral components.

The training of an MLP requires one-dimensional vectors, hence for the first phase of training the accumulated speckle images were flattened. However, the size of the accumulated speckle patterns was 128×128 px which when flattened would result in a one dimensional vector with 16,384 units. This size of input vector would result in huge memory consumption. Since reducing the size of the speckle symmetrically does not affect its dependence on wavelength, the initial speckle speckle image was cropped symmetrically (depicted in Fig. 4.13 (b)) to achieve a final size of 64×64 px. This also helped in the reduction of memory consumption with faster computation. The training data containing 10,000 cropped and flattened images per class. These were divided randomly into training, validation and testing with a ratio of 70 %, 15% and 15 % resulting in 8500 images for the training and validation set and 1500 images for test set in each class. The training of MLP was performed by implementing cross-entropy cost function (Eq. 4.1) for 1000 epochs with scaled conjugate gradient descent training function [216] with sigma value of 5×10^{-5} and lambda value of 5×10^{-7}. An optimal MLP geometry (trained using MATLAB

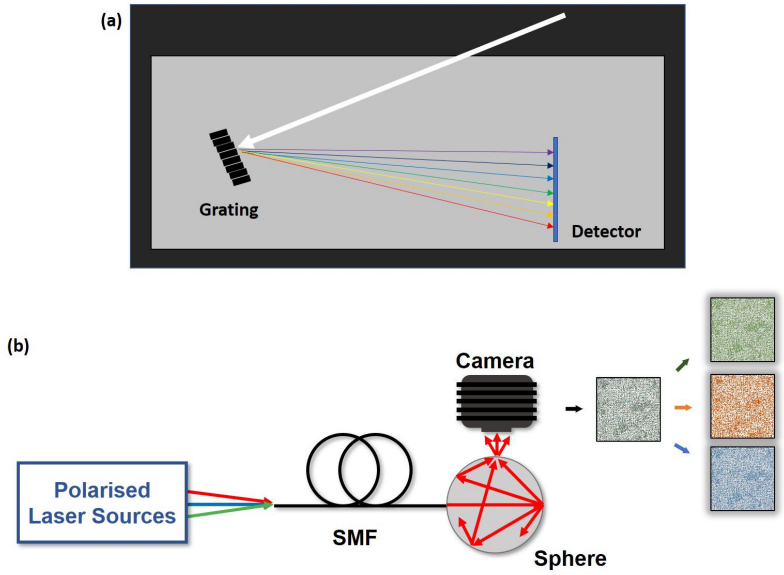

Figure 4.12: **Comparative schematic between the grating based spectrometer and a speckle spectrometer**. (a) A grating based spectrometer which works on the principal that the incident light is dispersed into constituent colors and these colors are detected using a detector. (b) A speckle based spectrometer where the incident polarised laser sources direct their light into a single mode fibre which then injects the light into integrating sphere. The sphere ejects a spectral speckle pattern onto the camera.

2020a over Nvidia Quadro P5000 GPU) with 15 hidden neurons for one layer was identified by changing the hidden neurons from 5 to 200 over a step of 5 neurons and changing the hidden layer from 1 to 5 over a step of 1 layer. A validation patience of 6 was considered for a minimum gradient of 1×10^{-6}. The final training stooped at 1.23×10^{-6} for reaching best validation performance.

After the first phase of training. The above trained MLP was considered to generate the probabilistic output vectors of the binary spectral data. A training dataset was prepared of one dimensional vectors with 30 units using 10,000 symmetrically cropped and flattened images per classes. This training set was divided randomly into the ratio of 70 % and 30% for training/validation set and test set respectively. The training/validation set was considered to optimize a second MLP geometry using Levenberg-Marquardt back-propagation algorithm using mean squared error for 1000 epochs with initial mu value of 1×10^{-3}, with an increase factor of 10 and decrease factor of 0.1. The validation patience was set at 6 until a minimum gradient performance of 1×10^{-7} was reached. The optimal MLP geometry was identified with 20 hidden neurons for one hidden layer by considering the performance over

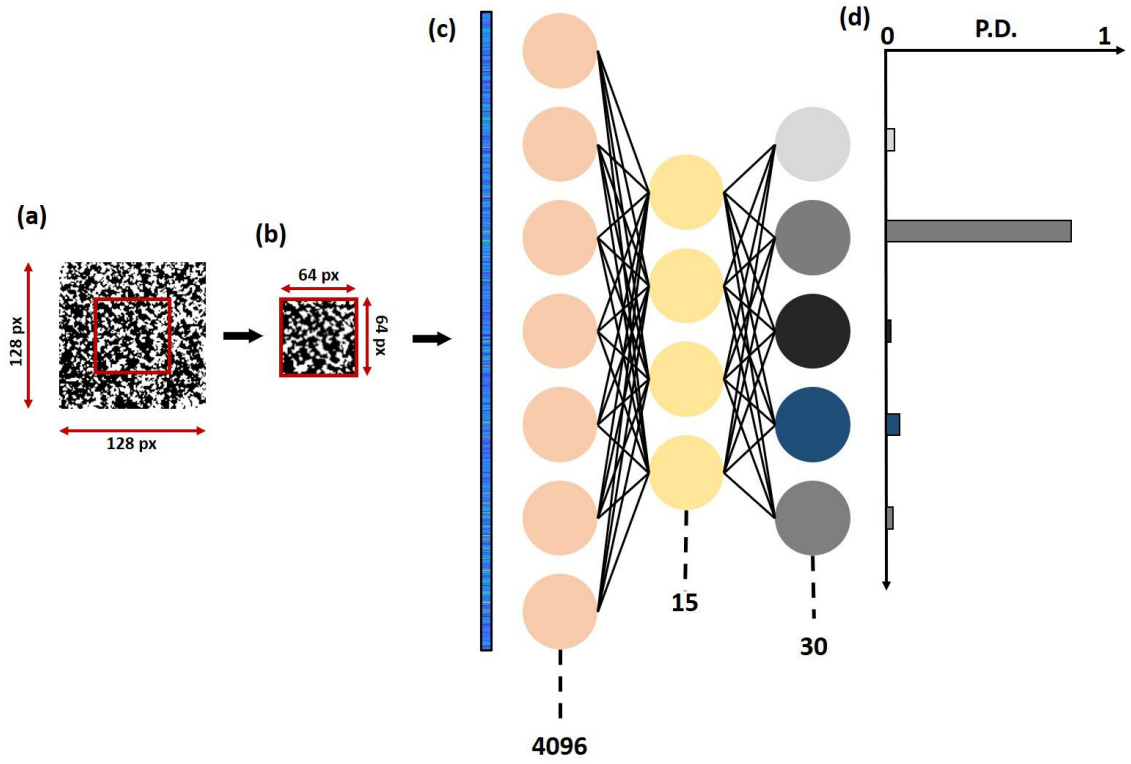

Figure 4.13: **Application of MLP for he classification of speckle images**. (a) 128 × 128 px monochromatic speckle images were cropped from the centre to get (b) reduced size speckle image of 64 × 64 px. (c) A MLP was tried to classify the speckle images, which finally resulted in a (d) probabilistic distribution vector for each training wavelength.

validation dataset. This geometry was identified by changing the hidden neurons from 5 to 200 over a step of 5 neurons and changing the number of hidden layers from 1 to 5 over a step of 1 layer. Final training was terminated at the gradient value of 8.65×10^{-8}. The trained regression based MLP resulted in a one dimensional vectors representing the intensity of individual component of the spectra.

4.5.2 Results

MLP wavemeter optimization

The first phase of training required an optimal MLP geometry which could classify the femtometre resolved dataset. As explained above, the training set was randomly split into training/validation and testing datasets in the ratio of 70 %, 15% and 15% respectively. After the training of MLP over the training set, an exceptional performance with 100 % classification accuracy was achieved for the test dataset.

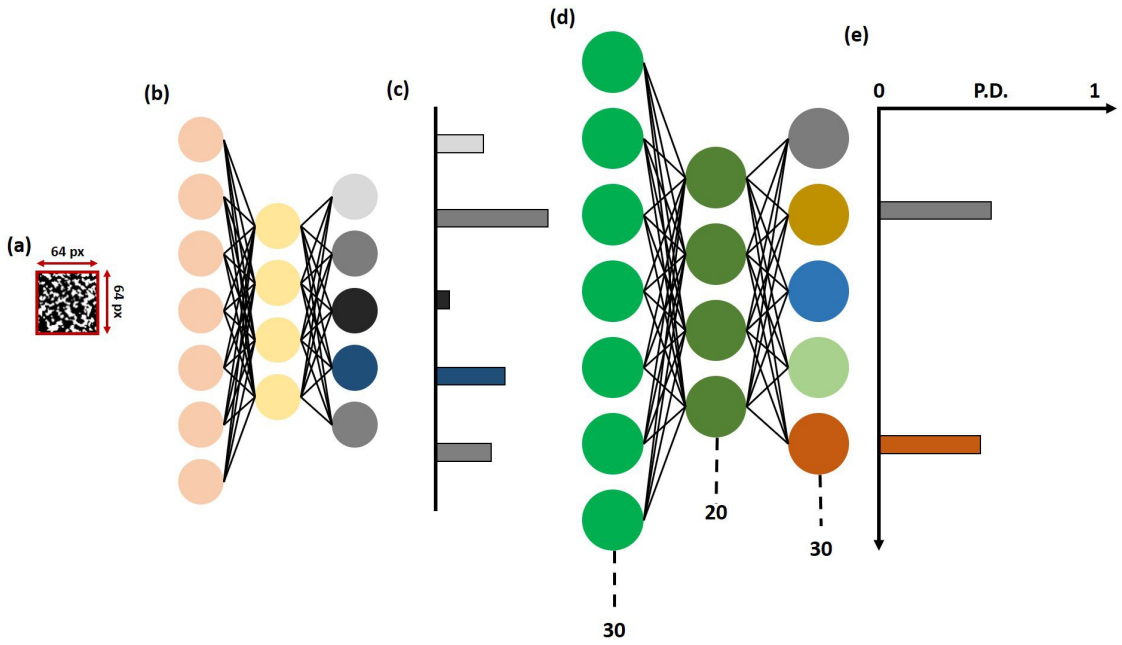

Figure 4.14: **Stacked MLP geometry for spectral regression**. (a) Input binary spectral symmetrically cropped speckle image. (b) Trained MLP for classification of wavelengths. (c) 1D probabilistic vector representing the binary spectral speckle pattern. (d) MLP trained to regress over the binary spectral response. (e) Output of regression based MLP for the binary spectral speckle pattern.

As demonstrated in Figure 4.15, the testing dataset was considered and the softmax output was evaluated for each symmetrically cropped flattened speckle image. This softmax output was then used to evaluate the confusion matrix in log scale. As shown, the confusion matrix demonstrates the high accuracy of classification. A mean softmax classification error of 1.27×10^{-6} was evaluated for the test dataset.

Stacked MLPs for binary spectral detection

After the successful training during the first phase, the trained MLP was considered with respect to the binary spectral dataset. Each images from the six classes (as shown in Tab. 4.4) of binary spectral data comprising of a total 10,000 images per class were processed using the trained MLP. As anticipated, the trained MLP resulted in very distinct probabilistic vectors for each class. As demonstrated in Figure 4.16, the bar charts for each class of binary spectra resemble mean value for each output node with an error bar. This shows that the MLP speckle images belonging to the binary wavelength dataset were not exactly identical and this difference was appreciated by the MLP as well. The output vectors were calculated for

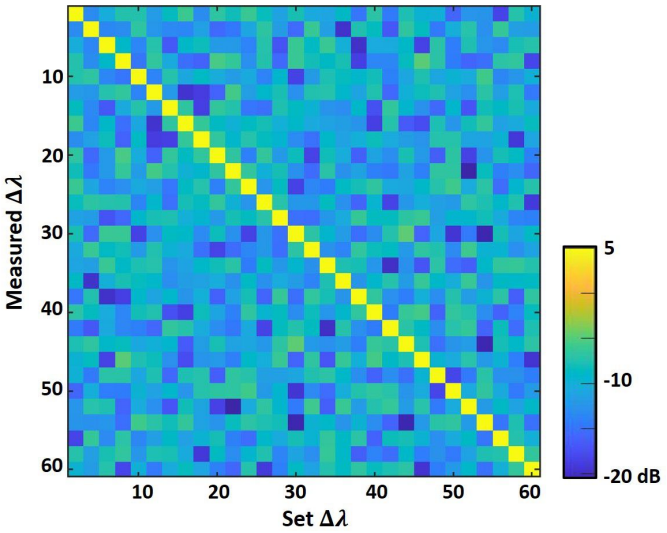

Figure 4.15: **High accuracy classification of femtometre resolved speckle patters using MLP**. Confusion matrix evaluated as the output of MLP for the classification of femtometre-resolved speckle images, plotted on $\log_{10}$ scale. The color bar represents decibel values.

all the images, and a mean and standard deviation were evaluated. The error bar in the figure shows the deviation present in the images corresponding to each class. A maximum deviation was observed for class with $\Delta\lambda_1 = 30$ fm and $\Delta\lambda_2 = 40$ fm with a variation of 0.6 units among all the nodes.

Using the one dimensional output vectors (as shown in Fig. 4.16) calculated from the trained MLP, a dataset was constructed with 10,000 patterns per spectral class. This dataset was randomly divided as mentioned above which resulted in 42000 one-dimensional training/validation vectors among all the six classes with 18000 one-dimensional testing vectors. Using this second dataset another MLP, with 20 hidden neurons and one hidden layer, was trained to evaluate the spectral components of the speckle patterns. The training of MLP based upon the mean squared cost function resulted in a stable spectrometer. Since the final spectral components of the speckle patterns can only be calculated by considering both the MLPs, together they can be called stacked MLP. The stability of the speckle based binary spectrometer can be appreciated from Figure 4.17, where the output from the stacked MLP have been presented. Each test vector was processed using the stacked MLP and a mean and standard deviation was calculated using the final outputs for each class. In this figure, the bars represent the mean value whereas the error bars represents the standard deviation for each class. The standard deviation

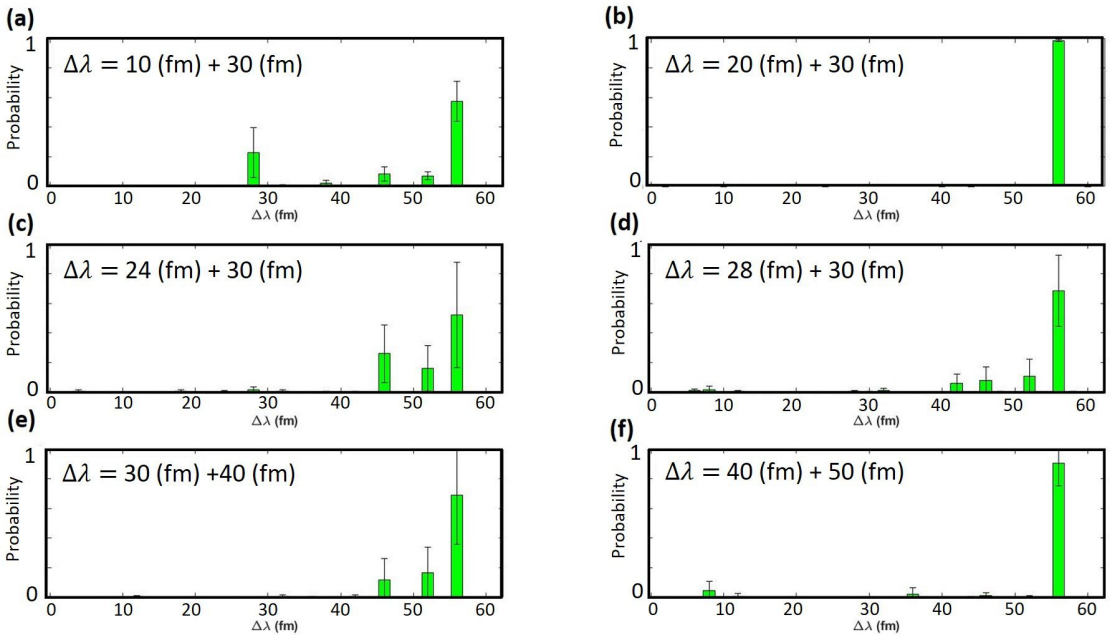

Figure 4.16: **Probabilistic output vectors for binary spectra of MLP trained over single wavelength femtrometre resolved speckle patterns**. One dimensional softmax output vector for wavelength deviations at (a) $\Delta\lambda_1 = 10$ fm and $\Delta\lambda_2 = 30$ fm, (b) $\Delta\lambda_1 = 20$ fm and $\Delta\lambda_2 = 30$ fm, (c) $\Delta\lambda_1 = 24$ fm and $\Delta\lambda_2 = 30$ fm, (d) $\Delta\lambda_1 = 28$ fm and $\Delta\lambda_2 = 30$ fm, (e) $\Delta\lambda_1 = 30$ fm and $\Delta\lambda_2 = 40$ fm, and (f) $\Delta\lambda_1 = 40$ fm and $\Delta\lambda_2 = 50$ fm.

for all the individual class as calculated as $\sim 1 \times 10^{-3}$ which signifies the presence of low variance after the speckle patterns with high variance and noise were processed using the stacked MLP. Hence, indirectly, the stacked MLP reduces the noise from the input noisy speckle images to finally evaluate the spectral components. This may be due to the presence of variations in the training data (Fig. 4.16). During the training process the MLP learns to overcome the variations in the spectral data and hence on the test set the processing by the MLP results in a much clearer and noise-less spectra.

4.6 Conclusion

In this chapter the application of Deep learning was explored to develop a speckle based wavemeter and binary spectrometer. The successful development of a speckle wavemeter with attometre resolution and a dynamic range in nanometre scale was demonstrated.

The initial methodology of developing a wavelength modulated speckle based

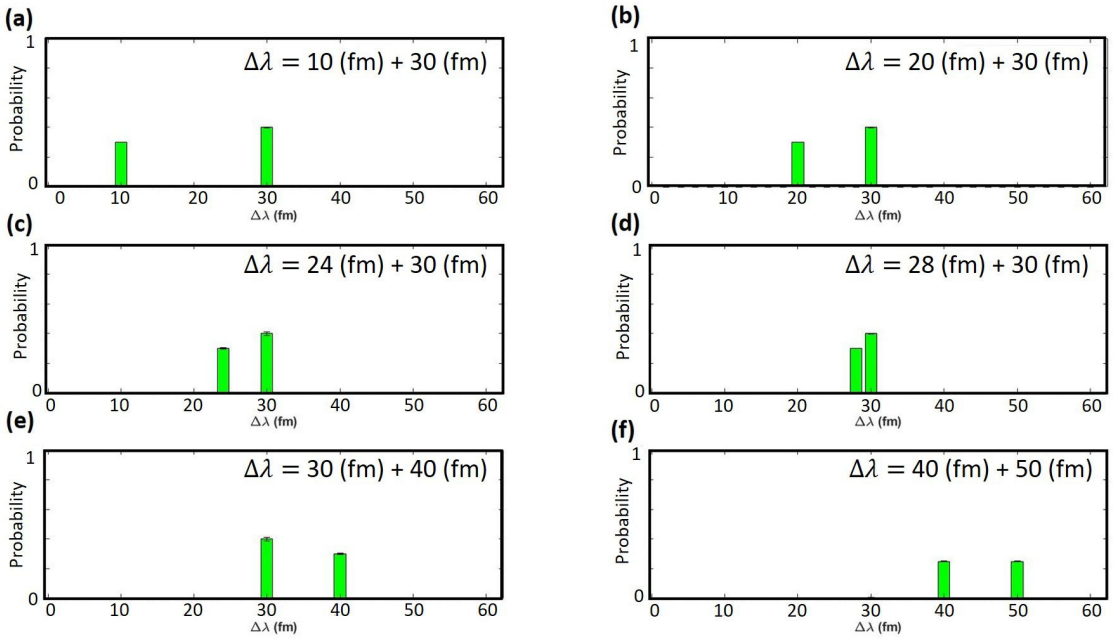

Figure 4.17: **Demonstration of a stacked MLP based speckle spectrometer** Spectral calculation using the stacked MLP for (a) $\Delta\lambda_1 = 10$ fm and $\Delta\lambda_2 = 30$ fm, (b) $\Delta\lambda_1 = 20$ fm and $\Delta\lambda_2 = 30$ fm, (c) $\Delta\lambda_1 = 24$ fm and $\Delta\lambda_2 = 30$ fm, (d) $\Delta\lambda_1 = 28$ fm and $\Delta\lambda_2 = 30$ fm, (e) $\Delta\lambda_1 = 30$ fm and $\Delta\lambda_2 = 40$ fm, and (f) $\Delta\lambda_1 = 40$ fm and $\Delta\lambda_2 = 50$ fm.

wavemeter demonstrated the ability of CNNs to find the simplest mathematical answer in terms of classification between a sinusoidal or triangular waveform. It was also demonstrated that the CNNs a very stable solution towards the classification of waveforms for their capability to correctly identify the waveforms which were accumulated after a two day interval. The failure of CNNs to classify five waveforms provided with an insight about their working which helped in the development of a wavemeter with high dynamic range.

The following study demonstrated an implementation of deep learning to classify single speckle patterns as a function of incident wavelength. This application illustrated a method to differentiate between laser wavelengths which could be separated by at least 2 am for a long range of 488 nm. As indicated, this limit was not identified to be the fundamental limit which could, if permitted by experimental instrument, be further enhanced. The study demonstrated a wavemeter with an excellent dynamic range of 3.25×10^{11} in a single step algorithm. It was also shown that the combination of CNN with speckle can also be generalised and applied to a completely different scattering medium. The highlighting factor of this study was

the demonstration of automated noise rejection which may be generated by both instrumental or environmental factors. These results could be very beneficial for automated laser stabilization and for noise rejection in multiple telecommunication applications. The two studies which involved the application of CNNs, for the classification of speckle waveform or pattern, illustrated that the fundamental convolution network geometry provided a base for both the applications and further classification was implemented by varying the number of neurons in the fully connected layers. Also for an optimal architecture, the CNN classification accuracy does not depend on the number of training classes, or the step size between them. A comparison can be made between different methods to develop a speckle based wavemeter: as shown in Fig. 4.18, the PCA based approach allows for a highly resolved wavemeter, whereas transmission matrix approach allows for a wavemeter with high bandwidth, however, the application of CNN based approach allows to construct a wavemeter with a high resolution and bandwidth, i.e., a wavemeter with a high dynamic range.

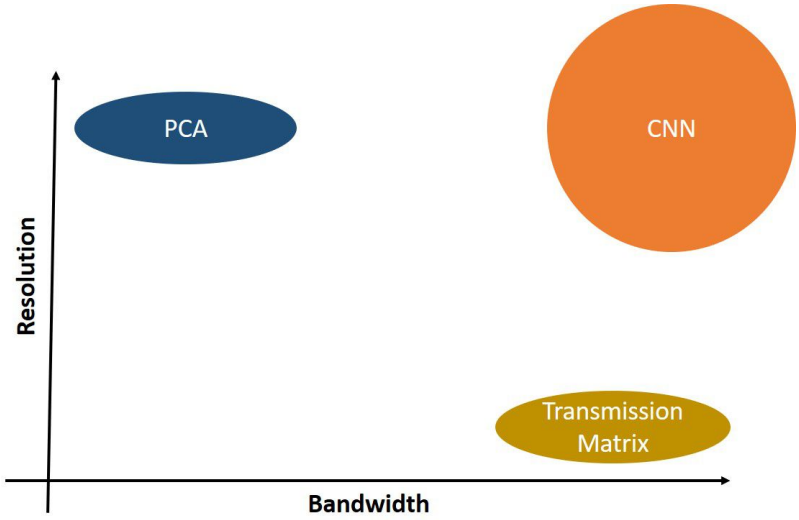

Figure 4.18: **Schematic comparison for speckle wavemeter performance with respect to the applications of PCA, transmission matrix and CNN**. Implementing PCA based approach allows for a highly resolved wavemeter, whereas the application of transmission matrix approach allows for the high bandwidth. The CNN based approach provides both, i.e., high resolution and bandwidth resulting in a wavemeter with a high dynamic range.

Finally, the application of feedforward networks was explored to develop a speckle spectrometer. This study shows that the MLPs can also be used to develop a highly accurate wavemeter with a resolution of 2 femtometre. The application of MLP trained only on one laser source was sufficient to develop the spectral 1D probabilistic

vectors for the binary spectral speckle patterns. These patterns showed the presence of high variation in terms of their 1D vector representations. However, when the second MLP was trained over these representations, the resultant binary spectra showed the absence of any variation. This also pointed towards an automated noise rejection capability of the MLP. These results show that the stacked MLPs can be implemented to calculate a binary spectra from the single image of the speckle patterns. An interesting future study could be to apply these finding to develop a Raman spectrometer. Detection of two or more Raman lines by directly observing the spectra using a cheap camera would be very fast and economical.

In contrast to the excellent capabilities demonstrated above, it would be important to discuss some limitations of the deep learning based ANNs. A major drawback of applying these networks is that they take very long time for the training process. This training time increases exponentially as the number of classes are increased. Another major drawback is that the CNN can only work with data points which belong to the training class. As an example, in the case of speckle wavelength measurement, a pre-trained ANN to classify cannot identify the intermediate wavelengths. To solve this a regression approach seems to be an appropriate alternate, but the regression based capabilities of the ANNs are known to be limited at much lower precision [217]. Hence, deep learning based ANNs do not prove to be ultimate solution for any problem but prove to be a reasonable analytic tool.

Relevant Publications

R.K. Gupta, G. D. Bruce, S. J. Powis & K. Dholakia, "Deep learning enabled laser speckle wavemeter with a high dynamic range" 02 Aug 2020, In : Laser & Photonics Reviews, Early View, 2000120 / arXiv:1910.10702 [physics.optics]

Contributions

The experiments were designed by the author, Dr. Graham Bruce and Prof. Kishan Dholakia. The author developed and designed all deep learning methods and performed the numerical analysis. Dr. Graham Bruce developed the experimental setup and performed the experiments.

Chapter 5

Application of machine learning to improve the limit of detection of a chirped guided mode resonance biosensor

5.1 Introduction

Biosensing techniques play a critical role in healthcare, environmental monitoring and the biomedical industry [218–222]. Optical biosensors offer great advantages over conventional analytical techniques due to their fast responding time, direct measuring, label-free detection, and integration ability [223,224]. The working principle of photonic biosensors is based on the interaction of an evanescent field with biorecognition elements and the detected signal is proportional to the concentration of the antigen [225]. Therefore, by engineering different optical fields, a huge number of optical sensors can be utilized, e.g. surface plasmon resonance biosensors (SPR) [226], Evanescent wave fluorescence biosensor [227], optical interferometric biosensors [228], disk resonator photonic biosensor [229], surface-enhanced Raman scattering biosensors (SERS) [230], optical microarray biosensors [231] etc. Among all these different types of bio-optic sensors, one common goal is to improve the detection accuracy in order to pursuit a lower limit of detection. By improving the detection accuracy, the noise hidden in the real biological signal can be reduced [232].

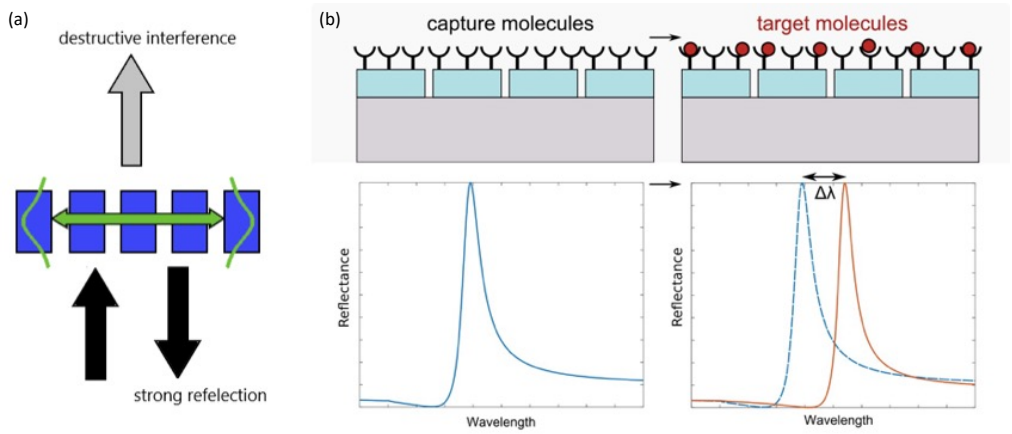

Figure 5.1: **Working principal of a guided mode biosensor**. (a) Demonstration of resonance response from chirped guided mode resonance (GMR) biosensor resulting in strong reflection (b) Demonstration of shift in peak position of the resonance response from the GMR as an effect to the capturing of target molecules on its surface.

A chirped guided mode resonance sensor is constructed by continuously altering the period of a one dimensional, sub-wavelength, dielectric grating along the grating grooves [233]. When it is illuminated by monochromatic light, a spatial resonance response in form of high reflectance from a horizontal strip can be read out (Fig. 5.1 (a)). The precise position of the resonance can be extracted by first averaging in the horizontal direction and then fitting a curve to this averaged intensity profile. As the extracted resonance curve is a result of an interplay between the Fabry-Perot resonance of the thin film and Brag resonance of the grating, it results in an asymmetrical Fano line shape which tends toward a Lorentzian line shape [234, 235]. The spatial position of this peak depends on the refractive index of the medium and is used to detect antibody binding. The capturing of target molecules at the surface of the device results in a variation in the refractive index and in-turn the peak position also shifts (Fig. 5.1 (b)). After calibrating the peak shift with respect to the refractive index, this shift helps in the determination of biological samples.

In the previous study a direct Fano fitting has been implemented in combination with the nonlinear-least-squares method [225]. This method helps in identifying a peak position of the resonance curve which in turn gives a direct indication towards the variation in the refractive index at the grating surface. However, this method

is limited to the shape of the resonance curve. Imperfections, e.g. stitching error of the grating [236], dust or defects from the sample or any instrumental noises present in the optical system could highly influence shape of the resonance, thus significantly reduce the accuracy of the peak position. To overcome these limitations, the numerical methods of machine learning can be implemented. In this chapter, the application of machine learning based methods of principal component analysis (PCA) and multi layered perceptron are explored.

5.2 PCA based improvement in the limit of detection of a chirped GMR

5.2.1 Methods

The working principal of the chirped GMR suggests that the resonance response is linearly dependent on the incident light wavelength. This means that if the incident light wavelength is tuned in fixed steps, the resonance response shall also shift proportionally. Following this understanding, the method of PCA (discussed in section 2.3.1, script mentioned in section A.4) was implemented since it outputs the degree of variations in the dataset. For this problem, the incident light wavelength was modulated linearly such that the maximal variation (first principal component) could be identified as a function of wavelength.

Principle of application

For the detection of resonance peak position, the two dimensional images were averaged in the horizontal direction. This averaging resulted in a one dimensional curve representing an intensity profile of the resonance. To remove any higher frequency noise, the signal was passed through a low pass filter with a band pass frequency of 0.5π rad/sample. To precisely measure the peak position of the calculated intensity profile, the methods of direct Fano fitting and differential Fano fitting were implemented. The differential Fano curve was calculated as the first principal component after applying PCA.

Fitting of resonance response

For direct Fano fitting, the intensity curve using a parametric Fano equation (Eq. 5.1) was used.

$$f(x) = a\left(\frac{bc + (x-d)^2}{c^2 + (x-d)^2}\right) + e \tag{5.1}$$

Here a represents the shape factor, b is the parameter for the symmetry of Fano curve, c is full width half maximum of the curve, d represents the peak position and e represents the curve offset.

After calculating a differential resonance signal using PCA, the curve was fit using a parametric differential Fano equation (Eq. 5.2).

$$f(x) = \left(\frac{a(2x - 2d + 2bc)}{c^2 + (d-x)^2} + \frac{a(2d - 2x)(x - d + bc)^2}{(c^2 + (d-x)^2)^2}\right) e + f \tag{5.2}$$

Here a represents the shape factor, b represents the symmetry of the differential Fano curve, c represents the skewness of the curve, d represents the zero crossing of the differential Fano curve, e represents the sign intensity of the curve and f represents the curve offset.

MATLAB was used to implement the fitting process using a non-linear least squares method of minimizing the least absolute residuals by following the Levenberg-Marquardt algorithm. The fitting error was determined as the uncertainty interval for the parameter d while evaluating the best fit for the experimental data.

Optimization of number of images

The dataset was accumulated by continuously varying the incidence light wavelength with a constant difference in the wavelength of 0.1 nm.

The PCA based fitting was applied over the dataset by considering different number of images (3, 5, 7, 9, 11, 13 and 15 images). The first principal component was considered to evaluate the differential signal and calculate the zero crossing. As shown in Fig. 5.2, the incident light beam was varied between 852 nm to 870 nm with a constant modulation of 0.1 nm. A deviation of peak position (PCA based differential fitting) was calculated from the mean peak position. Table 5.1 represents the deviation for the peak position (calculated using PCA based method) from the mean peak position. After doing this evaluation, it was established that

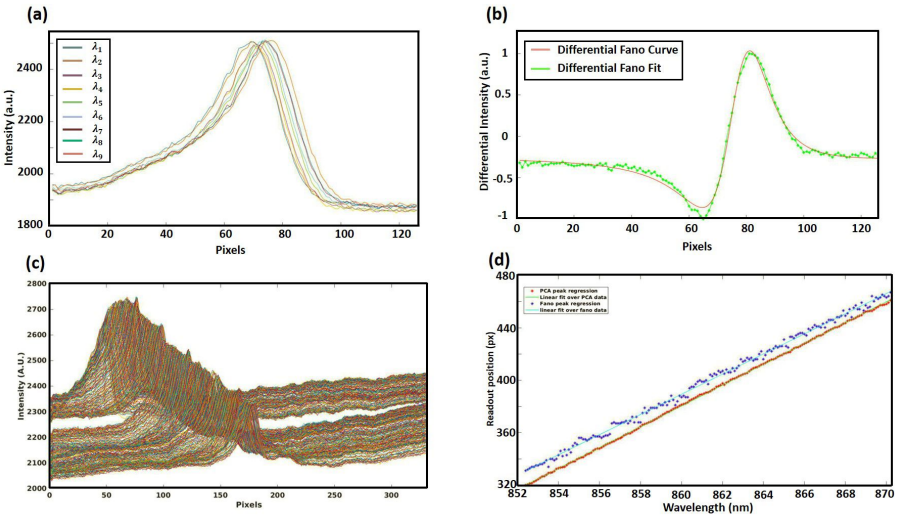

Figure 5.2: **Application of PCA to improve linear fit**. (a) Resonance response from a chirped GMR for 9 wavelengths modulated by 0.1 nm. (b) First principal component calculated using the data accumulated in (a). (c) Resonance response from the chirped GMR for the wavelengths from 852 nm to 870 nm in the step of 0.1 nm. (d) Resonance peak positions calculated using direct Fano fitting and PCA based differential fitting, here red curve denotes PCA based peak position estimation whereas blue curve denotes the Fano fitting based peak estimation.

the minimum deviation from the average peak position is present by considering 9 images.

For the linear modulation of incident light wavelength, direct Fano fitting method resulted in an estimated mean error of 3 pixels where as the PCA based differential fitting method resulted in a mean error of 1.26 pixels. This showed an improvement of PCA based mthod over direct Fano fitting method by a factor of two.

Number of images	**PCA peak position (pixels)**	**Average peak position (pixels)**	**Deviation from mean position (pixels)**
3	179.8	178.3	1.5
5	180.1	179.2	0.9
7	180.4	180.1	0.3
9	**181.09**	**181.1**	**0.01**
11	181.7	181.9	0.2
13	182.6	182.7	0.1
15	183.4	183.6	0.3

Table 5.1: Summary of characteristic variation of peak position to consider different number of images

3 σ stability estimation

The noise of a biosensor can be estimated by calculating the 3 times the standard deviation (σ) which is the smallest detectable shifts in the position for water placed over the grating. This value was calculated by measuring the standard deviation in the shift of peak position (P) and mean fitting error (F) (Eq. 5.3). The 3σ sensitivity of the chirped grating was estimated by keeping the incident laser wavelength at 830 nm with a modulation step of 0.1 nm. The smallest detectable shifts in peak position for water placed over the grating were observed for ∼20 minutes.

$$3\ \sigma = \sqrt{3std(P)^2 + F^2} \tag{5.3}$$

In the above equation, $std(P)$ symbolizes standard deviation in P. In order to make the calculations comparable, nine resonance images were fit using direct Fano fitting and the mean was calculated of the peak positions, in concurrence to one peak position calculated using PCA based method. Table 5.2 represents the variation of 3 σ value estimated for considering different number of images. Incidentally, the estimated 3 σ value for considering 9 images was found to be close to the minimum value.

Number of Images	**3 Sigma (pixels)**
3	3.6716
5	2.4789
7	2.2829
9	**2.2795**
11	2.2717
13	2.2813
15	2.2918

Table 5.2: Summary of 3 σ values calculated for different number of images.

Study to compare the performance of PCA based method with direct Fano fitting using two gratings

In order to analyse the improvement of PCA based method over conventional Fano fitting method, two types of gratings were considered. One smaller chirped grating with size of 400 μm by 500 μm, the pattern was within a write-field without stitching error, and another longer chirped grating of size 400 μm by 4 mm which

presented a stitching error. The chirped gratings were designed with two different channels, a signal channel and a reference channel. The reference channel was proposed for the purpose of accounting towards the temperature drift during the biological measurements.

5.2.2 Results

Improvement in resonance curve fitting

The identification of improvement in resonance curve fitting by using PCA based differential fitting was implemented by considering two different gratings with specific properties as discussed above.

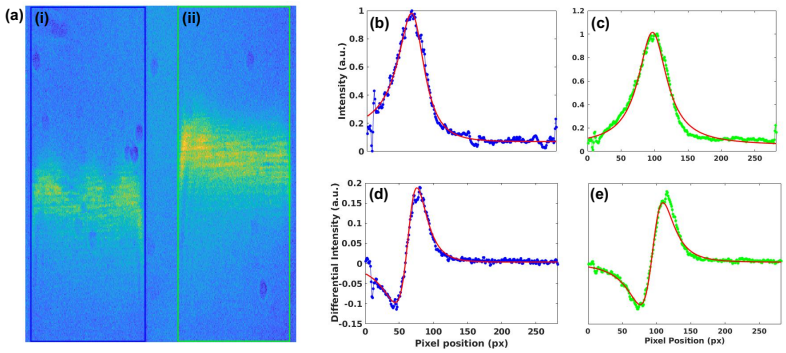

Figure 5.3: **Calculation of wavelength modulated resonance response for 500 μm long grating.** (a) Snapshot of the 500 μm long grating (i) Signal Channel highlighted in blue box (ii) Reference Channel highlighted in green box. Resonance response from the (b) signal channel and and (c) reference channel. Calculated wavelength modulated resonance response from the (d) Signal channel and (e) Reference channel. The plots colored in blue and green are the resonance response from the instrument and the red line is the fitting using Fano fitting for (b) and (c) whereas differential Fano fitting for (d) and (e).

The first grating was considered with the dimensions of 400 μm $\times$ 500 μm. A plane polarised light was incident on the grating from the substrate layer which induced a resonance response and high reflectance from the grating areas where the resonance condition is met. This resonance response can be identified from Fig. 5.3 (a) which shows two different microfluidic channels. Fig. 5.3 (a) (i) shows the signal channel whereas the Fig. 5.3 (a) (ii) shows the reference channel. Fig. 5.3 (b) and (c) show the mean evaluated Fano resonance response (dotted curve) and Fano curve fitting (red overlay curve) from the signal and reference channels. The wavelength modulated resonance response is depicted in Fig. 5.3 (d) and (e) representing the

differential Fano response (dotted curve) calculated using PCA and fitted differential Fano curve (red overlay curve) from the signal and reference channels respectively. The zero crossing of the differential curve signifies the peak position of the resonance response from the grating.

Characteristically, the second grating (a 400 μm $\times$ 4 mm grating) shows the presence of stitching error. The resonance response from this grating can be identified from Fig. 5.4 (a) which shows two different channels. Fig. 5.4 (a) (i) shows the signal channel whereas the Fig. 5.4 (a) (ii) shows the reference channel. Stitching error can be apprehended from the presence of horizontal lines at the illuminated region of the grating. Fig. 5.4 (b) and (c) show the mean evaluated Fano resonance response (dotted curve) and Fano curve fitting (red overlay curve) from the signal and reference channels. These curves show the presence of stitching error near the peak position. The wavelength modulated resonance response is depicted in Fig. 5.4 (d) and (e) representing the differential Fano response (dotted curve) calculated using PCA and fitted differential Fano curve (red overlay curve) from the signal and reference channels respectively.

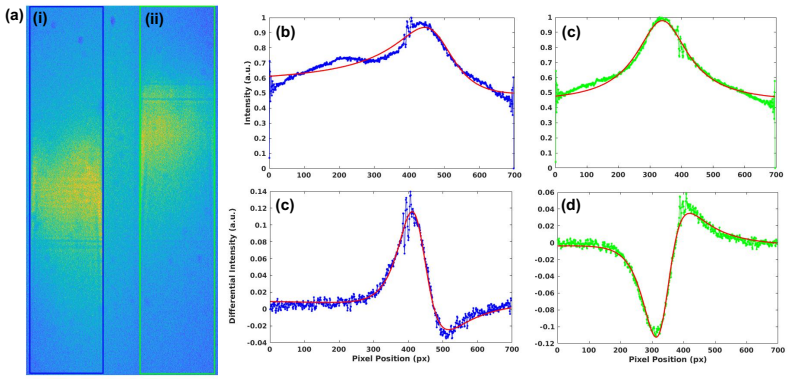

Figure 5.4: **Calculation of wavelength modulated resonance response for 4 mm long grating.** (a) Snapshot of the 4 mm long grating resembling the stitching error (i) Signal Channel highlighted in blue box (ii) Reference Channel highlighted in green box. Resonance response from the (b) signal channel and and (c) reference channel. Calculated wavelength modulated resonance response from the (d) Signal channel and (e) Reference channel. The plots colored in blue and green are the resonance response from the instrument and the red line is the fitting using Fano fitting for (b) and (c) whereas differential Fano fitting for (d) and (e).

The results suggest that PCA based method helps immensely to remove the presence of any other peaks which are not associated with the resonance response of the grating. When compared on the metric of R Squared values, the differential

fitting method shows a better fitting compared to direct Fano fitting method. For the 500 μm long grating, the R squared value for the signal and reference channel were 0.99 and 0.97 whereas the PCA based method showed the values as 0.98 and 0.98 respectively as can be observed in Fig. 5.3. For the 4 mm grating with stitching error, the R squared values for the signal and reference channel with respect to the Fano fitting came out to be 0.93 and 0.89 whereas PCA enhance the fitting and the values of 0.97 and 0.98 were calculated respectively which can be observed from Fig. 5.4.

Improvement in peak position determination

The peak position of the resonance response was calculated by implementing the PCA based differential Fano fitting method and direct Fano fitting method. The data was accumulated by linearly varying the incident laser wavelength from 850 nm to 856 nm with a constant step size of 0.1 nm. As explained before, the Fano response from the two gratings at each wavelength and differential Fano response by using 9 wavelength position were fit using the specified methods, for both 500 μm and 4mm gratings.

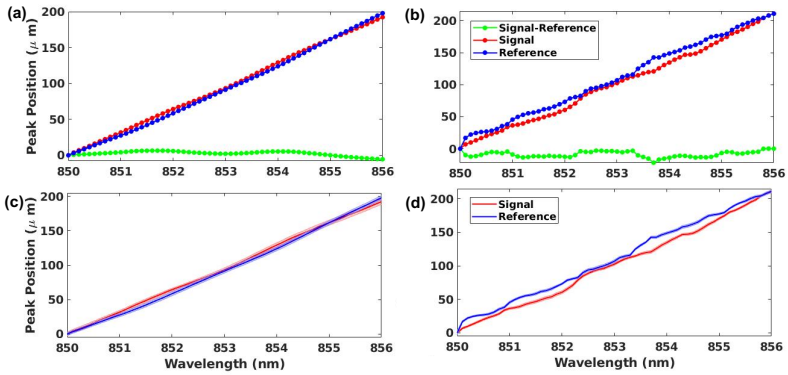

Figure 5.5: **Improvement in peak position determination for 500 μ m long grating.** (a) Calculated resonance peak position using Fano fitting, (c) Error calculated while determining the peak position. (b) Calculated resonance peak position using PCA based differential Fano fitting, (d) Error calculated while determining the peak position.

For 500 μm long grating, Fig 5.5 (a) and (b) represent the peak position variation with respect to the variation in the incident laser wavelength using direct Fano fitting and differential Fano fitting methods respectively, whereas Fig. 5.5 (c) and (d) represent the error evaluated at each data point for the above mentioned methods.

For PCA based method, the signal mean fitting error was calculated as 1.83 μm and the reference mean error fitting was calculated as 1.77 μm. This resulted in a total error of 3.6 μm. For the direct Fano fitting based method, the mean signal fitting error was calculated as 2.93 μm and the mean reference error was calculated as 2.74 μm. The total error evaluated was 5.67 μm.

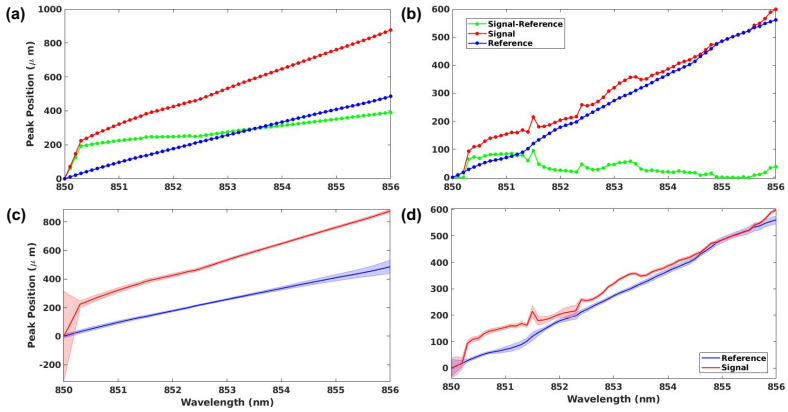

Figure 5.6: **Improvement in peak position determination for 4 mm long grating.** (a) Calculated resonance peak position using Fano fitting, (c) Error calculated while determining the peak position. (b) Calculated resonance peak position using PCA based differential Fano fitting, (d) Error calculated while determining the peak position.

For 4mm long grating, Fig 5.6 (a) and (b) represent the peak position variation with respect to the variation in the incident laser wavelength using direct Fano fitting and differential Fano fitting methods respectively, whereas Fig. 5.6 (c) and (d) represent the error evaluated at each data point for the above mentioned methods. For the direct Fano fitting based method, the mean signal fitting error was calculated as 17.11 μm and the mean reference error was calculated as 24.54 μm. The total error evaluated was 41.65 μm. For PCA based method, the signal mean fitting error was calculated as 6.12 μm and the reference mean error fitting was calculated as 0.03 μm. This resulted in a total error of 6.15 μm.

With respect to the results obtained for the two gratings it can be established that the two methods give very similar results for peak position estimation when the grating is free from any stitching error and external noises. However, the PCA based method proved to be highly efficient when implemented on a grating which presents an inherent stitching error. Evidently, this method showed an improvement by a factor of 6.77. Interestingly, it can be observed from the Fig. 5.5 and Fig. 5.6 that

PCA based analysis shows sharper variations in the peak position when compared to direct Fano fitting method.

Improvement in the 3 σ error

The 3 σ error of the two gratings was considered by measuring the variation in the resonance peak position while keeping the incident laser wavelength constant at 853 nm. To implement PCA based method, the light beam wavelength was modulated by 0.1 nm for 9 steps. This measurement was continued for $\sim$ 20 minutes, which is a reasonable time to observe a resonance shift due to biological binding.

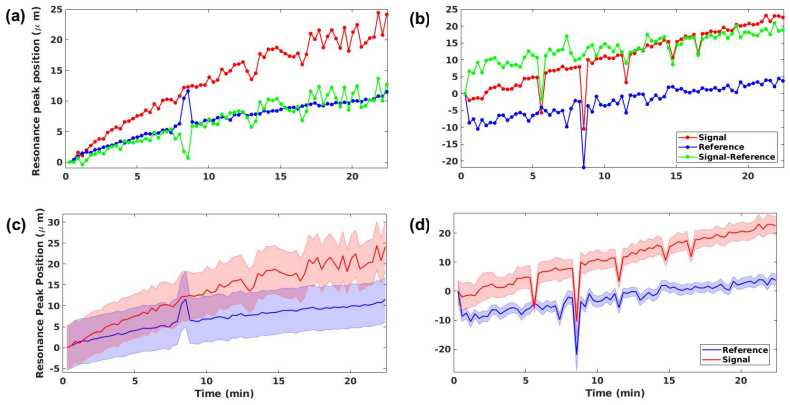

Figure 5.7: **3 σ error calculation for 500 μm long grating.** (a) Resonance peak position determination using Fano fitting method, (c) Error evaluated using the same method. (b) Resonance peak position determination using PCA based differential Fano fitting method, (d) Error evaluated using the same method.

Figure 5.7 (a) and (b) represent the variation in resonance peak position for 500 μm long grating using the direct Fano fitting and differential Fano fitting methods respectively. Figure 5.7 (c) and (d) represent the error evaluated while calculating the peak position using the two methods. Using Fano fitting method, the mean signal peak fitting error was calculated as 5.53 μm and the mean reference peak fitting error was calculated as 5.43 μm. The total fitting error was calculated as 10.96 μm. Total 3σ error was calculated as 15.68 μm. By implementing PCA based differential fitting method, the mean signal peak fitting error was calculated as 3.29 μm and the mean reference peak fitting error was evaluated as 1.97 μm. This resulted in the total fitting error of 5.26 μm. This method gave a 3σ error of 14.04 μm.

As shown in Figure 5.8, parts (a) and (b) represent the variation in resonance

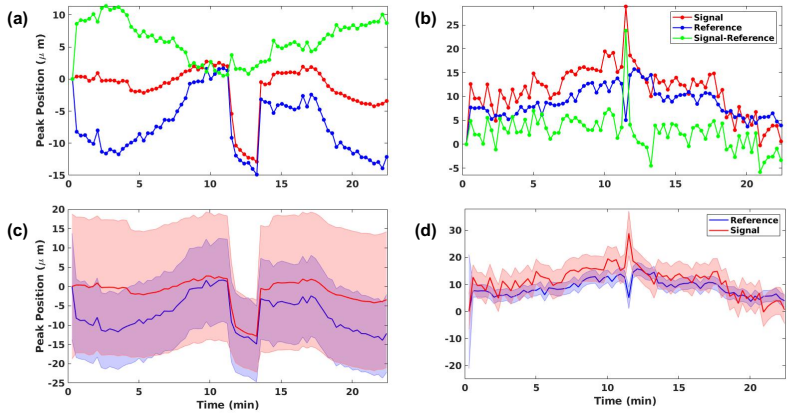

Figure 5.8: **3 σ error calculation for 4 mm long grating.** (a) Resonance peak position determination using Fano fitting method, (c) Error evaluated using the same method. (b) Resonance peak position determination using PCA based differential Fano fitting method, (d) Error evaluated using the same method.

peak position for 4 mm long grating using the direct Fano fitting and differential Fano fitting methods respectively. Figure 5.8 (c) and (d) represent the error evaluated while calculating the peak position using the two methods. Using Fano fitting method, the mean signal peak fitting error was calculated as 15.87 μm and the mean reference peak fitting error was calculated as 9.56 μm. The total fitting error was calculated as 25.43 μm. Total 3σ error was calculated as 26.95 μm. By implementing PCA based differential fitting method, the mean signal peak fitting error was calculated as 4.90 μm and the mean reference peak fitting error was evaluated as 2.42 μm. This resulted in the total fitting error of 7.32 μm. This method gave a 3σ error of 13.50 μm.

These results, in agreement with the linear wavelength variation results, suggest that PCA based method shows an improvement over direct Fano fitting method. As evaluated above, an improvement by the factor of 2 was observed when PCA based method was implemented to calculate the 3σ error for the longer 4 mm grating with stitching error.

Improvement in the detection of IgG

The detection of IgG binding to Anti-IgG antibody on the sensor was performed by measuring the variation in the resonance peak position by keeping the incident wavelength at 853 nm and applying a modulation of 0.1 nm for 9 steps. The measurements were done by varying the solutions in the microfluidic signal channel while

flowing only PBS in the reference channel.

Figure 5.9 represents the variation in resonance peak position with time, (a) by implementing direct Fano fitting and (b) by implementing PCA based differential fitting method whereas (c) and (d) represent the respective error calculated from the signal and reference channels using the mentioned methods. The variation in the resonance peak position with time represents precisely the variation in the presence of different biomolecules. After establishing the baseline with PBS, the Anti-IgG antibody binding curve saturated at 40 min because all the binding sites were occupied by the antibodies. This was followed by a PBS wash and then 1% casein was introduced as the blocking buffer. The chip was then washed again with PBS for 10 minutes between various concentrations of the IgG antigen being 10 pg/ml, 100 pg/ml and 10 μg/ml. Using the Fano fitting method, a mean signal peak fitting error came out as 10.37 μm and a mean reference peak fitting error came out as 6.67 μm. The total error came out to be 17.04 μm. Using PCA based method, a mean signal error came out to be 7.77 μm whereas the mean reference peak fitting error came out to be 0.03 μm. This resulted in a total error of 7.80 μm.

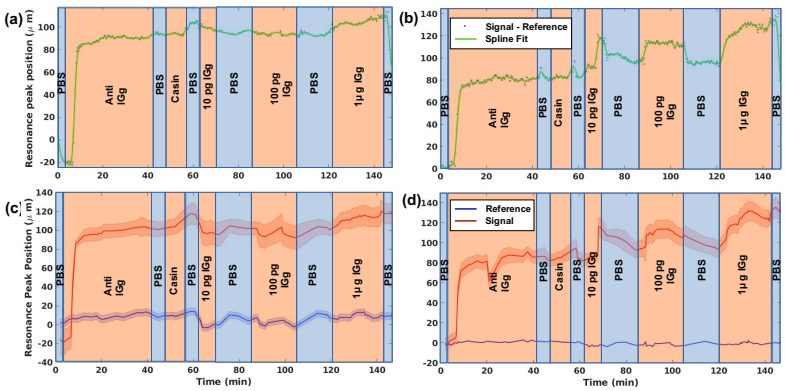

Figure 5.9: **IgG detection using 4 mm long grating.** (a) Resonance peak position determination using Fano fitting based method, (c) Error evaluated using the same method. (b) Resonance peak position determination using PCA based differential Fano fitting method, (d) Error evaluated using the same method.

While comparing the resonance peak positions evaluated using PCA based differential Fano fitting method and direct Fano fitting method, it can be confirmed that PCA based method shows an improvement over the other method. It is evident to distinguish each washing step by flushing PBS buffer and injecting IgG antigen in the time region between 60 to 150 minutes. When compared to Fig 5.9 (a), Fano fit

method does not show a clear behaviour since all the bonding and washing curve is buried in the noise. Additionally, this method shows a crest for PBS wash between 55 and 65 minutes where in principle a trough is expected, this can be explained due to the presence of stitching error. By using the PCA based method, the variations in peak position shifts are much more steep and detectable which shows that this method helps in improving the detection sensitivity of the grating by avoiding any inherent noise and improving on the overall fitting of the resonance responses. Based on the above comparisons, it can be summarized that using PCA based approach proves to be an improvement over direct Fano fitting method.

5.3 Classifying the resonance response of a chirped GMR using MLP

5.3.1 Methods

When a chirped GMR is shown with a light beam of a characteristic wavelength, it responds with a characteristic resonance. Hence, a characteristic resonance response from the GMR can be considered for classification problem. Inherently, in an experimental setting, the instrumental response may present noisy components resulting from the optics, vibrations or other experimental settings. The presence of these noisy components provides the variability in the dataset and prove to be fitting for the training of artificial neural networks (detailed in sections 2.3.6 and 2.4).

Data acquisition

To accumulate a dataset for the training of an MLP, a GMR chip was designed with two gratings with a similar resonance response (Fig. 5.10 (a)). First GMR chip was considered for training and the second chip was considered for the testing purposes. This dataset was accumulated by varying the incident light wavelength between 855.5 nm to 859.5 nm with a step of 0.1 nm. A total of 100 images were considered for each wavelength contributing to a total of 4100 images.

The two GMR's considered for this study (Fig. 5.10 (a)) show a linear response

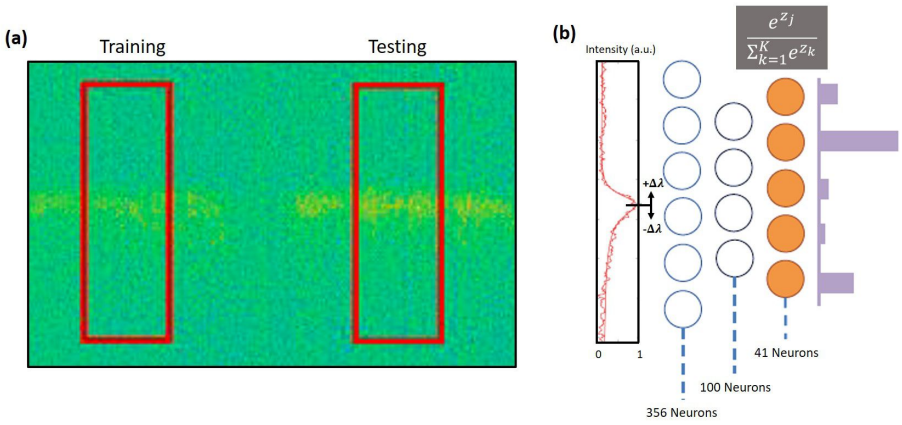

Figure 5.10: **Application of MLP for resonance peak detection**. (a) Wide field image of the resonance response from the training and testing GMR. The red boxes highlight the GMR's in consideration. (b) Schematic of the implemented MLP model with 356 input neurons, 100 hidden neurons and 41 output neurons. The highlighted equation in the figure is the softmax function as the output of the MLP.

to the incident wavelength. As shown in Fig. 5.11 (a), by implementing the direct Fano fitting method, the two GMR's show a linear variation in the readout peak position with respect to the incident light wavelength. The mean error in peak position detection for the training GMR was calculated as 3.09 μm and a mean error of 2.57 μm was calculated for the testing GMR.

To further emphasize the experimental and analytical fitting error, the resonance response from both the GMR's was considered at an incident wavelength of 855.5 nm. These curves were fit using the direct Fano fitting and are shown in Fig. 5.11. The direct Fano analysis resulted in the peak positions at 211 μm $\pm$ 2.38 μm (Fig. 5.11 (b)) and 217.38 μm $\pm$ 2.99 μm (Fig. 5.11 (c)).

Principle of application

To classify the resonance response from the chriped GMR, an MLP based model was considered. The dataset accumulated for training GMR, was considered for training and the was divided into training (70%), validation (20%) and test set (10%). The MLP model was optimized by considering an optimization routine [145] to draw maximum classification accuracy over the validation set. After the training, the data accumulated from the test GMR was considered for blind testing.

A 3 layered MLP with 356 neurons in the input layer, 100 neurons in the hidden

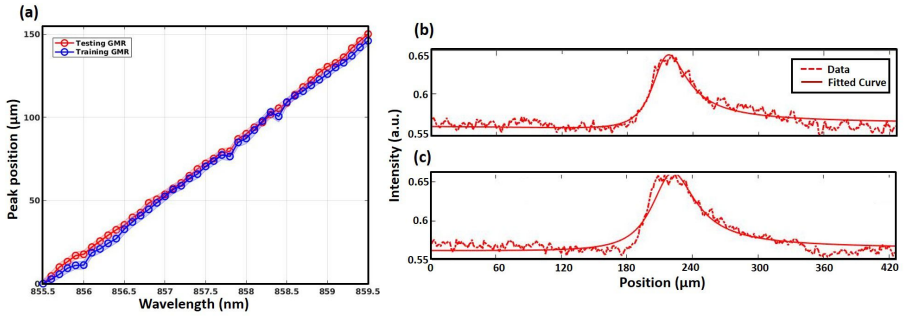

Figure 5.11: **Error evaluation using direct Fano fitting peak detection from the designed chip**. (a) Linear response from the GMRs in consideration, here shaded error bar represents the fitting error for each incident wavelength. Analytical fitting of the resonance response from the (b) training GMR and (c) testing GMR at the incident wavelength of 855.5 nm

layer and 41 neurons in the output layer was considered (Fig. 5.10 (b)) to classify the resonance peaks with respect to the incident laser wavelength. The training was applied using the resonance peaks obtained from training GMR. The MLP model was trained by considering the cross entropy cost function (also detailed in section 2.3.6) at the output layer as shown in equation 5.4.

$$\text{Cost} = -\frac{1}{k}\Sigma_x[y \times \log a + (1-y) \times \log(1-a)], \tag{5.4}$$

Here Σ_x represents training over all the input images x, k is the total number of training data points, y is the target output and a is the network output. Here y and a are the one hot vectors representing the category of the input resonance curves.

During the training, the MLP based classification model yielded a 100% classification accuracy over the test set from the training GMR. The best validation performance was identified at 7.56×10^{-4}. After the training process, the network's performance was blind tested over testing GMR.

Wavelength calibration and error analysis

With respect to the experimental fabrication process, a wavelength calibration for each pixel was evaluated as 1 px = 0.031 nm. Using this estimation, a comparitive error analysis was performed for the readout wavelength using MLP based classification model and direct Fano fitting mode.

To estimate the error of peak position detection using the MLP, the softmax

function (Eq. 5.5) was considered at the output of the MLP. This function inherently evaluates a probabilistic estimate on how correctly the neural network predicts a given class. To translate the softmax output of the network into wavelength, the MLP output for the correct class was subtracted from 1 and multiplied with the least count of wavelength step size, i.e., 0.1 nm.

$$\text{Softmax} = \frac{e^{z_j}}{\Sigma_k e^{z_k}} \tag{5.5}$$

Here, z_j is the j^{th} neuron of which the output is being evaluated and z_k are all the neurons of the layer.

In order to evaluate the grating stability for the same wavelength, a 3 σ value was evaluated by considering 100 images for each wavelength. The three sigma value was evaluated by considering the predicted peak position (P) and network error (E). The error was calculated as shown in equation 5.6.

$$3\sigma_{MLP} = \sqrt{3 \times std(P)^2 + E^2} \tag{5.6}$$

$std(P)$ is the standard deviation in the peak position detected using the MLP model.

5.3.2 Results

MLP performance comparison with direct Fano fitting

The network performance was compared with direct Fano fitting for the best and worst network wavelength predictions.

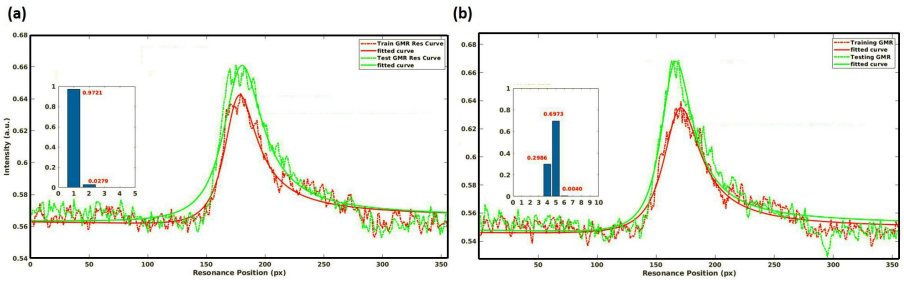

Figure 5.12: **Demonstration of classification abilities of MLP for peak position detection**. (a) Best case prediction from the MLP where the inset figure represents the probabilistic estimation of the class of the resonance response. (b) Worst case prediction of the resonance response using the MLP, here the inset figure shows the probabilistic prediction of the resonance classes.

Fig. 5.12, represents the best (part (a)) and worst (part (b)) performance of the MLP model and direct Fano fitting for the same resonance curves. The curve in red is the resonance curve obtained from training GMR and the green curve is from testing GMR. The inset figure shows the network's probabilistic (softmax) output.

For the best case scenario (Fig. 5.12 (a)), the network predicted the curve to be correct with a softmax classification error of 0.027. With respect to the error analysis explained before, MLP prediction error was calculated as 0.0027 nm whereas the direct Fano fitting yielded an error of 0.0765 nm. Comparing the two, an improvement by a factor of 28 was observed.

For the worst case scenario (Fig. 5.12 (b)), the MLP based error was estimated to be 0.17 nm whereas direct Fano fitting method resulted in an error of 0.03 nm. In this case, direct Fano fitting based method proved to be better by a factor of 1.7. Interestingly, in the worst case scenario, the ANN predicted between the adjacent classes of the wavelength in question showing a close probabilistic answer. Hence, considering the results, it can be understood that the MLP based model may perform well in case the training data shows ample variability.

Attributing to the presence of instrumental noise, once trained, the MLP model based results do not change with respect to the intensity variations between different datasets. Hence looking at these results, it may be concluded that the MLP performs very efficiently for 86.9% of the total dataset.

Comparison of 3 σ evaluation using MLP and direct Fano fitting

A 3 σ error was calculated for the best case scenario (Fig. 5.13 (a)). A total of 100 images were considered for the incident wavelength of 855.5 nm. Total 3 σ error for the MLP was calculated to be 4.91×10^{-6} nm whereas for the direct Fano fitting this error was evaluated as 6.68×10^{-2} nm. For this case, the neural network showed an improvement by a factor of 10000 times.

A histogram plot was also calculated to present the improvement of MLP based analysis when compared to the direct Fano fitting method for the complete dataset (testing GMR). As shown in Fig. 5.13 (b), the histogram plot shows the distribution of 3 sigma error for the complete test dataset comprising 4100 images. With respect to the 3 sigma histogram plot, MLP may predict the correct resonance response

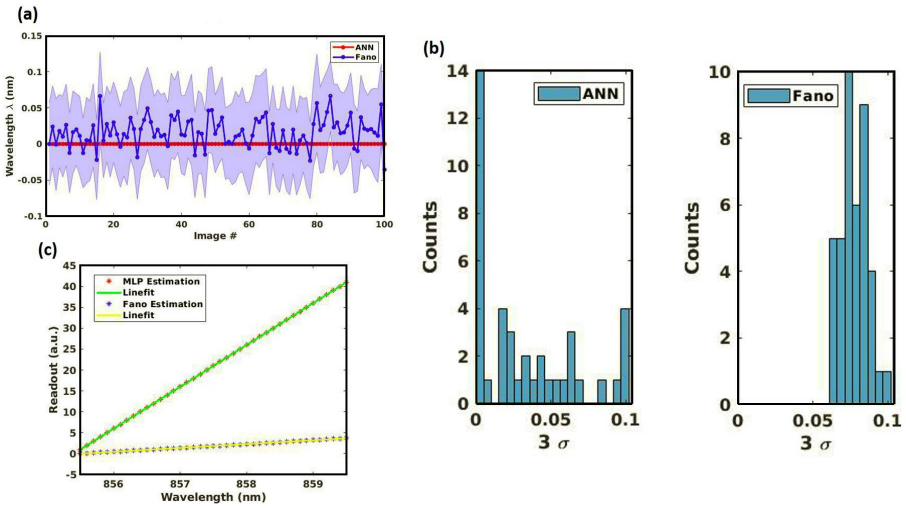

Figure 5.13: **Improvement in resonance peak position detection using MLP over direct Fano fitting**. (a) Representation of 3 σ value for a given dataset of 100 images at the incident wavelength of 855.5 nm. (b) Histogram plot distribution for the 3 σ over the complete dataset. (c) Readout curve showing the comparison between Fano fit and MLP based estimation.

with a zero deviation for 34% of instances. It was found that there was 14% chance that the MLP may give a 3 sigma error greater than 0.075 nm whereas for the direct Fano fitting method there is 95% chance that an average 3 σ value of 0.075 nm would be evaluated. A mean 3 σ error for the complete dataset of 4100 images was evaluated for both methods. MLP based method resulted in the mean 3 σ value of 0.031 nm whereas direct Fano fitting method resulted in a mean 3 sigma value of 0.077 nm. This resulted in a total improvement by a factor of 2.5 times.

To draw a comparison between Fano fit estimation and MLP estimation (best cases) for the readout value, a graph between readout value and incident light wavelength was plotted (Fig. 5.13 (c)). For the MLP based estimation, there was no deviation from the correct wavelength (class), whereas for the direct Fano fitting method a deviation of 0.03 px/nm was evaluated.

5.4 Conclusion and future work

This chapter examines the application of machine learning methods of principal component analysis and multi layered perceptron to improve the limit of detection of a chirped GMR. The results point in the direction that machine learning methods do indeed prove to be efficient in analysing the data by (1) removing the unnecessary

noisy variation (when PCA based method is applied) and (2) by harnessing the noisy variations (when MLP based method is applied). Both the methods show an improvement by at least a factor of two.

The application of principal component analysis based differential fitting method shows an improvement of at least 2 fold over the generic direct Fano fitting method. In particular, this approach indicates that chirped guided-mode resonance biosensor grating when combined with PCA, proves be highly stable and sensitive device to detect the bio-molecules, specifically it has been shown that IgG concentration of as low as 10 pg/ml can be detected. Moreover, it can be inferred that PCA based method may show a huge advantage when applied to track resonator resonances or surface plasmon resonances, essentially owing to its property for excluding any variations in the system which resemble random noise.

The application of MLP based classification method shows an improvement by a factor of 2.5 times for the complete dataset whereas an improvement by the factor of 10,000 is observed for the best case scenario. This study shows that MLP provides a massive advantage over direct Fano fitting or PCA based differential fitting provided the variations in the training dataset contain information for the interpolative estimation. Another aspect which can be recognized from this study is that the classification based approach is more accurate then regression approach for the estimation of resonance position.

As a future study, it would be interesting to test the trained MLP model over the GMR's which would be present on a completely different chip. Additonally, owing to the current results, in future, it would be interesting to study the MLP response over the smaller wavelength shifts which can prove to be very beneficial for the detection of much lower concentrations of IgG.

Contributions

The experiments were designed by the author, Dr. Kezheng Li, Prof. Kishan Dholakia and Prof. Thomas Krauss. The author developed and designed all machine learning methods and performed the numerical analysis. The instrument design and data acquisition was conducted by Dr. Kezheng Li.

Chapter 6

Conclusion and future outlook

6.1 Summary of the thesis

The work presented in this thesis focuses on the applications of machine learning in various aspects of optics and photonics. These aspects include the enhancement of label-free techniques of Raman spectroscopy and digital holographic microscopy, enhancement of laser speckle wavemeter and spectrometers and for the improvement in the sensitivity of the wavelength scale devices.

Chapter 2 introduced the foundations of machine learning by providing succinct introduction to various methods. A discussion of different approaches including, supervised learning, unsupervised and reinforcement learning is included. These discussions circumscribe the fundamentals of regression, classification, density estimation, clustering, dimensionality reduction, finite markov processes, dynamic programming, Monte Carlo methods and temporal difference learning. Following these discussions, this chapter provides a mathematical insight into the techniques of principal component analysis, linear discriminant analysis, k-means algorithm, support vector machines, t-distributed stochastic neighbour embedding and artificial neural networks. Subsequently, the fundamentals of deep learning and various applications in the context of optics and photonics are discussed with a mathematical description to feed forward networks, convolutional neural networks, recurrent neural networks and generative adversarial networks.

Chapter 3 explored the applications of label-free techniques of wavelength modulated Raman spectroscopy (WMRS) and digital holographic microscopy (DHM) to

classify the cells of immune system with a particular importance given to the analytic methods by employing deep learning based algorithms. The first study focuses on the application of artificial neural networks to improve the classification abilities of WMRS and DHM for the neutrophils and eosinophils. The WMRS spectral and DHM phase image data was acquired for both the cell types and then a comparison between the analysis methods of principal component analysis with leave one out cross validation and artificial neural networks was considered. As a final result it was identified that the application of deep learning based CNNs with DHM provides a route towards a robust, stand-alone and high throughput hemogram device.

The second study focuses on improving the throughput rate of DHM for the classification of T-cell subsets namely $CD4^+$ and $CD8^+$ cells. In this study the optical system was tweaked by changing the objectives with the magnifications of 20X, 60X and 100X which resulted in the phase images with varied sizes. To ensure an optimal CNN geometry for each class of phase images, a particle swarm optimization methods was considered to optimize the CNN geometries. Further investigation was considered to improve the throughput rate of the DHM system by employing cycle GAN type training module to train the CNN geometries for the transformation of phase images acquired from microscopic objective with 20X magnification to the phase images acquired from the microscopic objective with 100X magnification. This provided a way to improve the throughput rate by 100 folds with an additional improvement in the classification efficiency for DHM based hemogram system.

Chapter 4 explored the regime of the application of deep learning for the classification purposes. This chapter examined the possibility of developing the laser speckle based wavemeters and spectrometers. Initial study to develop a wavemeter was based on the wavelength modulation. The failure of this method provided an insight towards the working of a CNN which in-turn helped to develop a wavemeter with high dynamic range.

Subsequent study explored the application of CNNs to classify the speckle patterns with respect to incident laser wavelength without any explicit modulations. As a result of successful classification of femtometre resolved speckle images, the automated rejection of environmental and instrumental noise by the CNN was also demonstrated. Following the understanding of the working of the CNN, three ap-

plications namely, for attometre resolution, high dynamic range and generalization capabilities were explored. By invoking t-SNE based analysis method in combination with CNN, it was demonstrated that the CNN based speckle wavemeter can resolve speckle images with a wavelength deviation of as low as 2 am and as high as 486 nm. The generalization capabilities in terms of being applicable to a different diffusive medium (ground glass) was also demonstrated.

The next study explored the application of multi layered perceptrons (MLP) for the development of laser speckle spectrometer. This study investigated the application of stacking of these networks and training them in two steps. First step was considered as classification training module for an MLP which resulted in the development of a wavemeter with 2 femtometre. The second step was considered as a regression training step where a second MLP was trained over the probabilistic distribution output of the first MLP for the binary spectral speckle patterns. This resulted in the development of a spectrometer for binary spectral datasets.

Chapter 5 explored the application of machine learning to improve the sensitivity of wavelength scale devices, specifically chirped guided mode resonance (GMR) biosensors. The strong resonant reflections from the GMR results into a fano shaped intensity output whose peak position is directly proportional to the variation in incident light wavelength and refractive index over the surface of the sensor. To identify the sensitivity of this sensor, the intensity profile is fitted analytically with fano curve. First study in this chapter explored a comparison between the application of principal component analysis (PCA) based different fano fitting and direct fano fitting. As an outcome, PCA based method demonstrated an improvement by a factor of two over direct fano fitting method.

The second study was conducted as an exploration of MLP for the classification of output intensity reflection from the sensor with respect to the incident light wavelength. As an outcome, the MLP based method presented with different types of outcomes. For the best case scenario MLP based method resulted in an improvement by a factor of 10,000 over the conventional fano fitting method. However, for the worst case scenario the direct fano fitting method beat MLP by a factor of more than two. These results provide an insight that if the amount of data is adequate, MLP based method may prove to be highly beneficial in terms of improving the

sensitivity of the chirped biosensor device.

6.2 Future outlook

This section will discuss about the possible prospects which can be considered for future work.

6.2.1 High throughput label-free hemogram for all the cells of immune system

Chapter 3 of this thesis demonstrated the application WMRS and DHM for the classification of morphologically similar granulocyte and T immune cells namely, neutrophils, eosinophils, $CD4^+$ and $CD8^+$ T cells. The two studies paved a way in the direction that DHM based imaging systems when combined with deep learning can be implemented for the classification of these cells with high accuracy and high throughput. This makes the combination of DHM-CNN based hemogram an attractive option when compared to the generic flow cytometry based cell classification. The future work can be directed to train a CNN over the phase images of multiple cell types such as B cells, natural killer cells, mast cells, dentritic cells or macrophages. This would in-turn help in the development of a single unit DHM device which may finally replace the current labelling techniques.

Another application of the high-throughput nature of DHM based system can be thought of to study the activation of various cells of the immune system. The DHM based phase images for various stages of activation can be analysed and studies using the artificial neural networks. These networks can then be further implemented to detect the next state of the cells. This may find several applications in disease progression and provide an idea to the physicians on which medicine to be used at which stage.

From a commercial perspective, a future work can be thought of as development of a miniaturised handheld version of DHM device which can directly be provided to the consumers as a replacement of long term pathological tests.

6.2.2 Optical precision metrology

Chapter 4 explored the application of deep learning for the development of speckle based wavemeter. The development of a high dynamic range wavemeter with the capabilities of automated noise rejection opens the possibility of a plethora of applications in telecommunications. A future work can be incorporated to stabilize the laser wavelength using a feedback mechanism.

Another study can be directed to develop a universal speckle meter. This meter may detect a range of dependencies of speckle pattern. These dependencies can be variation in temperature, mode of incident beam, polarization, etc. This can further be commercialised to develop a handheld device which may help in the detection of minute changes in the properties of environment and light mentioned above.

As this chapter also explores the development of a speckle spectrometer, another future study can be directed towards the development of a spectrometer with broader range and ability to identify more than two wavelengths. With broad operation range, the spectrometer can be applied to detect flourescence, or Raman lines. This may provide a solution which would be highly accurate, cheap and small in size.

6.2.3 Wavelength scale device

Chapter 5 explores the application of MLP for the classification of intensity output of a chirped GMR with respect to the incident light wavelength. An interesting future study would be to test a trained MLP model over the GMR's which would be present on completely different chip. Additionally, it would be interesting to study the working of an MLP over smaller wavelength shifts which can prove to be very beneficial for the detection of much lower concentration of IgG.

From a fundamental perspective, the deep learning based algorithms can be used as predictive tools to design new types of wavelength scale devices with even higher efficiency.

6.3 Conclusion

The applications of machine learning presented in this thesis highlight its potential in the field of biophotonics and laser metrology. The improvement in classification

abilities of the label-free modalities using artificial neural networks have paved a way towards the development of DHM based hemograms. The further improvement in throughput rate of the DHM using cycle GAN module proved that the combination of DHM with CNNs is highly competitive with the current labelling methods for cellular classification. The broader applicability of deep learning for the development of a speckle wavemeter with high dynamic range may aid in telecommunications. The demonstration for the development of speckle based binary spectrometer paved a way towards a highly precise, cheap and robust Raman / Fluorescence spectrometers. Finally, the application of machine learning for the improvement in the sensitivity of the wavelength scale devices provided another vision where the improvements can be made beyond the expected levels of current methods.

Appendix A

Matlab Codes

A.1 Matlab Code for training CNN

Matlab code to train a CNN geometry with a given data (train, validation and test) location, layers and training options. "**XX**" in training options represent the hyperparameters which may be optimized with respect to a given dataset.

```
clear all;close all; clc; reset(gpuDevice(1)); rng('default');
trainLocation = 'TrainLocation';
ValLocation = 'ValLocation';
testLocation = 'TestLocation';

trainValImds = imageDatastore(trainLocation,'IncludeSubfolders',true,'LabelSource','foldernames','FileExtensions','.png');
valImds = imageDatastore(ValLocation,'IncludeSubfolders',true,'LabelSource','foldernames','FileExtensions','.png');
testImds = imageDatastore(testLocation,'IncludeSubfolders',true,'LabelSource','foldernames','FileExtensions','.png');

refImgSize = size(imread(trainImds.Files{1}));
```

```
11 fprintf('Defining layers and Gradient descent parameters...\
       n');
12
13 %Defining Layers
14 layers = ['Layers'];
15 % Defining training options
16 options = trainingOptions('optimizer',...
17     'MaxEpochs',XX,...
18     'InitialLearnRate',XX,...
19     'LearnRateDropFactor',XX,...
20     'LearnRateDropPeriod',XX,...
21     'L2Regularization',XX,...
22     'ValidationData',valImds,...
23     'ValidationFrequency',XX,...
24     'ValidationPatience',XX,...
25     'MiniBatchSize',XX,...
26     'Shuffle','every-epoch',...
27     'Plots','none',...
28     'Verbose',0,...
29     'ExecutionEnvironment','gpu');
30
31 [net,tr] = trainNetwork(trainImds,layers,options);
```

A.2 Matlab Code for optimizing CNN geometry using PSO

This section provides the source code for the optimization routine developed to identify the optimal CNN geometry with respect to a given dataset.

A.2.1 Main

The following script is the key code which contains “GetLayers” and “trainCNN” functions. Here the particle swarm optimization algorithm is employed with 40 particles. The particles’ position and velocity are updated using Eq. 3.9 and the cost is evaluated using Eq. 3.8.

```
clear all; close all; rng('default'); reset(gpuDevice(1));
    clc;

numInd = 40; %number of particles in the swarm
numVar = 33; %number of variables
maxIter = 500;
tolerance = 1e-2;

% Range for initial swarm's elements
range_max = 10;

iter = 1; % Number of iteration
k = 1.5; % weight of stocastic element

c_cost = 0;
gamma = 2.5;

flag = 1;
ind1 = randi(range_max,numInd,numVar-2);
ind2 = rand(numInd,2,1);
ind = abs([ind1 ind2]);
v = (zeros(numInd,numVar));

iter = 1;
radius = 1000;

while iter < maxIter && radius > tolerance
```

```
27      tic
28      ll=1;
29      while ll<=numInd
30          try
31              [a,~,~,~] = trainCNN(ind(ll,1:20),ind(ll,21),ind
                    (ll,22:26),ind(ll,27:31),ind(ll,32),ind(ll
                    ,33));
32              valF(ll,1)=a; % Fitness function for the swarm
33              ll = ll+1;
34          catch
35              valF(ll,1) = 1;
36              ll = ll + 1;
37          end
38      end
39      [valF_ord,index] = sort(valF); % Sort the objective
           function's values for the swarm and identify the
           leader
40      leader = ind(index(1),:)
41      [cst,net,~,~] = trainCNN(leader(1:15),leader(16),leader
           (17:21),leader(22:26),leader(27),leader(28));
42      cost(iter) = cst;
43      for l=1:size(ind,1)
44      % Calculates the new velocity and positions for all
           swarm's elements
45          fi = rand();
46          if (valF(l)==1)||isnan(valF(l))
47              ind1 = randi(range_max,1,numVar-2);
48              ind2 = rand(1,2,1);
49              ind(l,:) = abs([ind1 ind2]);
50          else
51              % Velocity
```

```
52                v(l,:) = (1-(sqrt(k*fi))/2)*v(l,:) + k*fi*(
                      leader-ind(l,:));
53                % Position
54                ff = ind(l,:) + gamma*(1-(sqrt(k*fi))/2)*v(l,:)
                      + (1-k*fi)*(leader-ind(l,:));
55                % Setting the values to be positive
56                ind(l,1:26) = abs(floor(ff(1:26)));
57                ind(l,27:28) = abs(ff(27:28));
58            end
59        end
60        radius = norm(leader-ind(index(end),:)); % Calculates
              the new radius
61        fprintf('Iteration Number:%d\t Radius:%f Cost:%f\n',iter
              ,radius,cst);
62        lead_db(:,iter) = leader;
63        iter = iter + 1; % Increases the number of iteration
64        toc
65
66    end
67
68    % Output variables
69    p_min = ind(1:numVar,:);
70    f_min = valF_ord(1:numVar,:);
```

A.2.2 GetLayers Function

```
1   function layers = GetLayers(imgSize,ind,fcn,cnn,pd,flg,dp)
2   % Setting thresholds
3   if dp<0.1
4       dp = 0.1;
5   elseif dp>1
6       dp=1;
```

```
7  end
8
9  for i = 1:length(ind)
10     if ind(i)<2
11         ind(i) = 0;
12     elseif ind(i)>10
13         ind(i) = 0;
14     end
15 end
16
17 for i = 1:length(cnn)
18     if cnn(i)<=0
19         cnn(i) = 1;
20     elseif cnn(i)>75
21         cnn(i) = 75;
22     end
23 end
24
25 for i = 1:length(pd)
26     if pd(i)<0
27         pd(i) = 0;
28     end
29 end
30
31 if flg<=0.5
32     flg=0;
33 elseif flg>0.5
34     flg=1;
35 end
36 % Defining Layers
37 layer{1} = [imageInputLayer(imgSize)];
38 layer{2} = [convolution2dLayer(1,cnn(1),'Padding',pd(1),'
```

```
    Stride',1)];
layer{3} = [convolution2dLayer(2,cnn(2),'Padding',pd(2),'
    Stride',1)];
layer{4} = [convolution2dLayer(3,cnn(3),'Padding',pd(3),'
    Stride',1)];
layer{5} = [convolution2dLayer(4,cnn(4),'Padding',pd(4),'
    Stride',1)];
layer{6} = [convolution2dLayer(5,cnn(5),'Padding',pd(5),'
    Stride',1)];
layer{7} = [batchNormalizationLayer];
layer{8} = [dropoutLayer(dp)];
layer{9} = [maxPooling2dLayer(2,'Padding',1)];
layer{10} = [reluLayer];
if flg
    layer{11} = [fullyConnectedLayer(fcn) tanhLayer
        fullyConnectedLayer(2) softmaxLayer
        classificationLayer];
else
    layer{11} = [fullyConnectedLayer(2) softmaxLayer
        classificationLayer];
end

% Loop to combine all the selected layers
layers = [layer{1}];
for i = 1:length(ind)
    if ind(i)
        layers = [layers layer{ind(i)}];
    end
end
layers = [layers layer{end}];
end
```

A.2.3 trainCNN Function

```
function [cst,net,tr,layers] = trainCNN(ind,fcn,cnn,pd,flg,
    dp)
%fixing the train validation data for consistent performance
load('trainValData_60X.mat');

refImgSize = size(imread(trainImds.Files{1}));
layers = GetLayers(refImgSize,ind,fcn,cnn,pd,flg,dp);

% Setting training options
options = trainingOptions('adam',...
      'MaxEpochs',100,...
      'InitialLearnRate',1e-3,...
      'LearnRateDropFactor',0.1,...
      'LearnRateDropPeriod',1,...
      'L2Regularization',5e-6,...
      'ValidationData',valImds,...
      'ValidationFrequency',40,...
      'ValidationPatience',5,...
      'MiniBatchSize',128,...
      'Shuffle','every-epoch',...
      'Verbose',0,...
      'Plots','none',...
      'ExecutionEnvironment','gpu');

[net,tr] = trainNetwork(trainImds,layers,options);

% Evaluation of cost based upon the sensitivity and
    specificity acquired using the performance over
    validation data
valPredictedLabels = classify(net,valImds);
valLabels = valImds.Labels;
```

```
accuracy_v = sum(valPredictedLabels == valLabels)/numel(
    valLabels);
cc = confusionmat(valPredictedLabels,valLabels)
sens = cc(1)/(cc(1)+cc(3)); spec = cc(4)/(cc(2)+cc(4));

fprintf('Validation Accuracy: %f \t MCC: %f \n',accuracy_v
    *100,((sens+spec)/2));
cst = 1-((sens+spec)/2);
clear trainImds valImds;
end
```

A.3 Matlab Code for cycle GAN training

This section provides the scripts written for the training of generative models of CNN for the transformation of images across domains as explained in section 3.3.1. The script "modelGradients" uses the GAN loss explained in Eq. 3.13 for the evaluation of gradients with respect to each model.

A.3.1 Main

```
clear all; close all; clc; rng('default'); reset(gpuDevice
    (1));
load('imgData_tr_vl.mat');

Ga =  generator1; %20X to 100X
Gb =  generator2; %100X to 20X
Da = discriminator1;
Db = discriminator2;

numEpochs = 2500;
miniBatchSize = 43;
lr = 0.0002;
gdf = 0.5;
```

```
13 sgdf = 0.999;
14 flag = 1; ctr = 0;
15
16 executionEnvironment = "auto";
17 flipFactor = 0.3;
18 validationFrequency = 25;
19 numValidationImages = 25;
20
21 trAvDa = []; trAvSqDa = [];
22 trAvDb = []; trAvSqDb = [];
23 trAvGa = []; trAvSqGa = [];
24 trAvGb = []; trAvSqGb = [];
25
26 f = figure;
27 f.Position(3) = 2*f.Position(3);
28 imageAxes = subplot(1,2,1);
29 imgAxes = subplot(1,2,2);
30
31
32 itr = 0;
33 start = tic;
34 %%
35 for epoch = 1:numEpochs
36     ctr = 0; flag = 1;
37     %get minibatch of images
38     while flag
39         mbatch = randsample(225,miniBatchSize);
40         IMG_mbatch_100_4 = IMG_tr_100_4(:,:,:,mbatch);
41         IMG_mbatch_100_8 = IMG_tr_100_8(:,:,:,mbatch);
42         IMG_mbatch_20_8 = IMG_tr_20_8(:,:,:,mbatch);
43         IMG_mbatch_20_4 = IMG_tr_20_4(:,:,:,mbatch);
44
```

```
45          mbatch_val = randsample(75,numValidationImages);
46          IMG_mbatch_val_100_4 = IMG_val_100_4(:,:,:,
                mbatch_val);
47          IMG_mbatch_val_100_8 = IMG_val_100_8(:,:,:,
                mbatch_val);
48          IMG_mbatch_val_20_8 = IMG_val_20_8(:,:,:,mbatch_val)
                ;
49          IMG_mbatch_val_20_4 = IMG_val_20_4(:,:,:,mbatch_val)
                ;
50
51          itr = itr + 1;
52          dl_IMG_mbatch_100_4  = dlarray(IMG_mbatch_100_4 , '
                SSCB');
53          dl_IMG_mbatch_20_4 = dlarray(IMG_mbatch_20_4,'SSCB')
                ;
54          dl_IMG_mbatch_100_8  = dlarray(IMG_mbatch_100_8 , '
                SSCB');
55          dl_IMG_mbatch_20_8 = dlarray(IMG_mbatch_20_8,'SSCB')
                ;
56
57          dl_IMG_mbatch_val_100_4  = dlarray(
                IMG_mbatch_val_100_4 , 'SSCB');
58          dl_IMG_mbatch_val_20_4 = dlarray(IMG_mbatch_val_20_4
                ,'SSCB');
59          dl_IMG_mbatch_val_100_8  = dlarray(
                IMG_mbatch_val_100_8 , 'SSCB');
60          dl_IMG_mbatch_val_20_8 = dlarray(IMG_mbatch_val_20_8
                ,'SSCB');
61
62          a_real = cat(4,dl_IMG_mbatch_100_4,
                dl_IMG_mbatch_100_8); a_real = gpuArray(a_real);
```

```
63          b_real = cat(4,dl_IMG_mbatch_20_4,dl_IMG_mbatch_20_8
                );b_real = gpuArray(b_real);
64
65          a_real_val = cat(4,dl_IMG_mbatch_val_100_4,
                dl_IMG_mbatch_val_100_8); a_real_val = gpuArray(
                a_real_val);
66          b_real_val = cat(4,dl_IMG_mbatch_val_20_4,
                dl_IMG_mbatch_val_20_8); b_real_val = gpuArray(
                b_real_val);
67
68          % evaluate model gradients for all generators and
                discriminators
69          [gradGa,gradGb,gradDa,gradDb,stateGa,stateGb,G_loss]
                = ...
70              dlfeval(@modelGradients,Ga,Gb,Da,Db,a_real,
                    b_real);
71          Ga.State = stateGa; Gb.State = stateGb;
72
73          % Update the discriminator network parameters.
74          [Da,trAvDa,trAvSqDa] = adamupdate(Da,gradDa,trAvDa,
                trAvSqDa,itr,lr,gdf,sgdf);
75          [Db,trAvDb,trAvSqDb] = adamupdate(Db,gradDb,trAvDb,
                trAvSqDb,itr,lr,gdf,sgdf);
76          % Update the generator network parameters.
77          [Ga,trAvGa,trAvSqGa] = adamupdate(Ga,gradGa,trAvGa,
                trAvSqGa,itr,lr,gdf,sgdf);
78          [Gb,trAvGb,trAvSqGb] = adamupdate(Gb,gradGb,trAvGb,
                trAvSqGb,itr,lr,gdf,sgdf);
79
80          % Every validationFrequency iterations, display
                batch of generated images using the
81          % held-out generator input
```

```
82          if mod(itr,validationFrequency) == 0 || itr == 1
83              % Generate images using the held-out generator
                    input.
84
85              a_val_gen = predict(Ga,b_real_val);
86
87              b_val_gen = predict(Gb,a_real_val);
88
89              % Tile the images
90              I = imtile(gather((extractdata(a_val_gen))));
91              I1 = imtile(gather(extractdata(b_val_gen)));
92
93              % Display the images.
94              subplot(1,2,1);
95              imagesc(imageAxes,I); axis image;
96              xticklabels([]);
97              yticklabels([]);
98              title("G_a 20X -> 100X images");
99
100             subplot(1,2,2);
101             imagesc(imgAxes,I1); axis image;
102             xticklabels([]);
103             yticklabels([]);
104             title(['G_b 20X->100X images Iterations: '
                    num2str(itr)])
105             drawnow;
106             saveas(gcf,['F:\WideFieldDHM\
                    CD4_CD8_imgSizeRecalibration\Results\CGAN\
                    incImprovement_04-09-20\Iteration_' num2str(
                    itr) '.png']);
107         end
108         % Update the scores plot
```

```
109             CycleGan_loss(itr) = gather(extractdata(G_loss));
110             D = duration(0,0,toc(start),'Format','hh:mm:ss');
111             fprintf('Epoch: %d \t Iteration: %d \t C-Gan Loss:%f
                    \t Duration: %s\n',epoch,itr,CycleGan_loss(itr),
                string(D));
112
113             ctr = ctr+miniBatchSize;
114             if ctr >= 450
115                 flag = 0;
116                 ctr = 0;
117             else
118                 flag = 1;
119             end
120         end
121 end
```

A.3.2 Model Gradients

```
1 function [gradGa,gradGb,gradDa,gradDb,stateGa,stateGb,g_loss
    ] = ...
2     modelGradients(Ga,Gb,Da,Db,a_real,b_real)
3
4 % Calculate the predictions for generated data with the
    discriminator network.
5 [a_fake] = forward(Ga,b_real);
6 [b_fake] = forward(Gb,a_real);
7 [a_rec,stateGa] = forward(Ga,b_fake);
8 [b_rec,stateGb] = forward(Gb,a_fake);
9
10 b_r_dis = sigmoid(forward(Da,b_real));
11 a_r_dis = sigmoid(forward(Db,a_real));
12 b_f_dis = sigmoid(forward(Da,b_fake));
13 a_f_dis = sigmoid(forward(Db,a_fake));
```

```
14
15 r_label = ones(size(b_f_dis));
16 f_label = zeros(size(a_f_dis));
17 a_gen_loss = mean((a_f_dis - r_label).^2);
18 b_gen_loss = mean((b_f_dis - r_label).^2);
19
20 a_rec_loss = mean(mean(mean(abs(a_rec - a_real))));
21 b_rec_loss = mean(mean(mean(abs(b_rec - b_real))));
22 g_loss = a_gen_loss + b_gen_loss + 10*(a_rec_loss +
      b_rec_loss);
23
24 a_d_r_loss = mean((a_r_dis - r_label).^2);
25 a_d_f_loss = mean((a_f_dis - f_label).^2);
26 b_d_r_loss = mean((b_r_dis - r_label).^2);
27 b_d_f_loss = mean((b_f_dis - f_label).^2);
28
29 a_d_loss = a_d_r_loss + a_d_f_loss;
30 b_d_loss = b_d_r_loss + b_d_f_loss;
31
32 gradGa = dlgradient(g_loss,Ga.Learnables,'RetainData',true);
33 gradGb = dlgradient(g_loss,Gb.Learnables,'RetainData',true);
34 gradDa = dlgradient(a_d_loss,Da.Learnables);
35 gradDb = dlgradient(b_d_loss,Db.Learnables);
36
37 end
```

A.3.3 Discriminator1

```
1 function [disc] = discriminator1
2 filterSize = 5;numFilters = 64;scale = 0.8;
3 layers = [
4     imageInputLayer([52 52 1],'Normalization','none','Name',
        'in')
```

```
5       dropoutLayer(0.5,'Name','dropout')
6       convolution2dLayer(filterSize,numFilters,'Stride',2,'
            Padding','same','Name','conv1')
7       leakyReluLayer(scale,'Name','lrelu1')
8       convolution2dLayer(4,2*numFilters,'Stride',2,'Padding','
            same','Name','conv2')
9       batchNormalizationLayer('Name','bn2')
10      leakyReluLayer(scale,'Name','lrelu2')
11      convolution2dLayer(3,4*numFilters,'Stride',2,'Padding','
            same','Name','conv3')
12      batchNormalizationLayer('Name','bn3')
13      leakyReluLayer(scale,'Name','lrelu3')
14      convolution2dLayer(3,8*numFilters,'Stride',2,'Padding','
            same','Name','conv4')
15      batchNormalizationLayer('Name','bn4')
16      leakyReluLayer(scale,'Name','lrelu4')
17      convolution2dLayer(3,16,'Name','conv5')
18       convolution2dLayer(2,1,'Name','conv6')];
19
20  lgraphDiscriminator = layerGraph(layers);
21  disc = dlnetwork(lgraphDiscriminator);
22  end
```

A.3.4 Discriminator2

```
1   function [disc] = discriminator2
2   filterSize = 5;numFilters = 64;scale = 0.8;
3   layers = [
4       imageInputLayer([200 200 1],'Normalization','none','Name
            ','in')
5       dropoutLayer(0.5,'Name','dropout')
6       convolution2dLayer(filterSize,numFilters,'Stride',2,'
            Padding','same','Name','conv1')
```

```
7       leakyReluLayer(scale,'Name','lrelu1')
8       convolution2dLayer(5,2*numFilters,'Stride',2,'Padding','
            same','Name','conv2')
9       batchNormalizationLayer('Name','bn2')
10      leakyReluLayer(scale,'Name','lrelu2')
11      convolution2dLayer(5,4*numFilters,'Stride',2,'Padding','
            same','Name','conv3')
12      batchNormalizationLayer('Name','bn3')
13      leakyReluLayer(scale,'Name','lrelu3')
14      convolution2dLayer(5,8*numFilters,'Stride',2,'Padding','
            same','Name','conv4')
15      batchNormalizationLayer('Name','bn4')
16      leakyReluLayer(scale,'Name','lrelu4')
17      convolution2dLayer(5,8*numFilters,'Stride',2,'Padding','
            same','Name','conv5')
18      batchNormalizationLayer('Name','bn5')
19      leakyReluLayer(scale,'Name','lrelu5')
20      convolution2dLayer(5,4*numFilters,'Stride',2,'Padding','
            same','Name','conv6')
21      batchNormalizationLayer('Name','bn6')
22      leakyReluLayer(scale,'Name','lrelu6')
23      convolution2dLayer(5,1,'Stride',2,'Padding','same','Name
            ','conv7')
24      batchNormalizationLayer('Name','bn7')
25      leakyReluLayer(scale,'Name','lrelu7')
26      convolution2dLayer(5,1,'Stride',2,'Padding','same','Name
            ','conv8')];
27
28  lgraphDiscriminator = layerGraph(layers);
29  disc = dlnetwork(lgraphDiscriminator);
30  end
```

A.3.5 Generator 1

```
function [gen] = generator1
layers = [
    imageInputLayer([52 52 1],'Normalization','none','Name',
        'in')
    convolution2dLayer(5,8,'Stride',2,'Padding',1,'Name','c1
        ')
    reluLayer('Name','rl')
    convolution2dLayer(5,16,'Stride',2,'Padding',1,'Name','
        c2')
    reluLayer('Name','rl1')
    batchNormalizationLayer('Name','bn1')
    convolution2dLayer(4,16,'Stride',2,'Padding',1,'Name','
        c3')
    reluLayer('Name','rl2')
    convolution2dLayer(4,32,'Stride',2,'Padding',1,'Name','
        c4')
    reluLayer('Name','rl3')
    batchNormalizationLayer('Name','bn2')
    convolution2dLayer(3,32,'Stride',1,'Padding',0,'Name','
        c5')
    transposedConv2dLayer(10,64,'Name','tconv1')
    batchNormalizationLayer('Name','bnorm1')
    reluLayer('Name','relu1')
    transposedConv2dLayer(5,64,'Stride',2,'Cropping','same',
        'Name','tconv2')
    batchNormalizationLayer('Name','bnorm2')
    reluLayer('Name','relu2')
    transposedConv2dLayer(5,64,'Stride',2,'Cropping','same',
        'Name','tconv3')
    batchNormalizationLayer('Name','bnorm3')
    reluLayer('Name','relu3')
```

```
24      transposedConv2dLayer(5,64,'Stride',2,'Cropping','same',
            'Name','tconv4')
25      batchNormalizationLayer('Name','bnorm4')
26      reluLayer('Name','relu4')
27      transposedConv2dLayer(4,32,'Stride',2,'Cropping','same',
            'Name','tconv5')
28      batchNormalizationLayer('Name','bnorm5')
29      reluLayer('Name','relu5')
30      transposedConv2dLayer(5,32,'Name','tconv6')
31      batchNormalizationLayer('Name','bnorm6')
32      reluLayer('Name','relu6')
33      transposedConv2dLayer(5,16,'Name','tconv7')
34      reluLayer('Name','relu7')
35      transposedConv2dLayer(5,16,'Name','tconv8')
36      batchNormalizationLayer('Name','bnorm8')
37      reluLayer('Name','relu8')
38      transposedConv2dLayer(5,8,'Name','tconv9')
39      batchNormalizationLayer('Name','bnorm9')
40      reluLayer('Name','relu9')
41      transposedConv2dLayer(5,8,'Name','tconv10')
42      batchNormalizationLayer('Name','bnorm10')
43      reluLayer('Name','relu10')
44      transposedConv2dLayer(5,4,'Name','tconv11')
45      batchNormalizationLayer('Name','bnorm11')
46      reluLayer('Name','relu11')
47      transposedConv2dLayer(5,4,'Name','tconv12')
48      batchNormalizationLayer('Name','bnorm12')
49      reluLayer('Name','relu12')
50      transposedConv2dLayer(5,2,'Name','tconv13')
51      batchNormalizationLayer('Name','bnorm13')
52      reluLayer('Name','relu13')
53      transposedConv2dLayer(5,2,'Name','tconv14')
```

```
54      batchNormalizationLayer('Name','bnorm14')
55      reluLayer('Name','relu14')
56      transposedConv2dLayer(5,1,'Name','tconv15')
57      ];
58  lgraphGen = layerGraph(layers);
59  gen = dlnetwork(lgraphGen);
60  end
```

A.3.6 Generator 2

```
1   function [gen] = generator2
2   layers = [
3       imageInputLayer([200 200 1],'Normalization','none','Name
            ','in')
4       convolution2dLayer(5,16,'Stride',2,'Padding',1,'Name','
            c1')
5       reluLayer('Name','rl')
6       convolution2dLayer(5,16,'Stride',2,'Padding',1,'Name','
            c2')
7       reluLayer('Name','rl1')
8       batchNormalizationLayer('Name','bn1')
9       convolution2dLayer(4,32,'Stride',2,'Padding',1,'Name','
            c3')
10      reluLayer('Name','rl2')
11      convolution2dLayer(4,32,'Stride',2,'Padding',1,'Name','
            c4')
12      reluLayer('Name','rl3')
13      batchNormalizationLayer('Name','bn2')
14      convolution2dLayer(3,64,'Stride',1,'Padding',0,'Name','
            c5')
15      reluLayer('Name','rl4')
16      batchNormalizationLayer('Name','bn3')
17      convolution2dLayer(3,64,'Stride',2,'Padding',0,'Name','
```

```
        c6')
18      reluLayer('Name','rl5')
19      batchNormalizationLayer('Name','bn4')
20      convolution2dLayer(3,128,'Stride',2,'Padding',0,'Name','
            c7')
21      transposedConv2dLayer(10,128,'Name','tconv1')
22      batchNormalizationLayer('Name','bnorm1')
23      reluLayer('Name','relu1')
24      transposedConv2dLayer(5,64,'Stride',2,'Cropping','same',
            'Name','tconv2')
25      batchNormalizationLayer('Name','bnorm2')
26      reluLayer('Name','relu2')
27      transposedConv2dLayer(5,64,'Stride',2,'Cropping','same',
            'Name','tconv3')
28      batchNormalizationLayer('Name','bnorm3')
29      reluLayer('Name','relu3')
30      transposedConv2dLayer(5,32,'Name','tconv4')
31      batchNormalizationLayer('Name','bnorm4')
32      reluLayer('Name','relu4')
33      transposedConv2dLayer(5,16,'Name','tconv5')
34      batchNormalizationLayer('Name','bnorm5')
35      reluLayer('Name','relu5')
36      transposedConv2dLayer(5,1,'Name','tconv6')
37      ];
38  lgraphGen = layerGraph(layers);
39  gen = dlnetwork(lgraphGen);
40  end
```

A.4 Matlab code: PCA

A general script where the variable "data" can be loaded into the work-space and then principal components can be calculated.

```
1  clear all; close all; clc;
```

```
2
3  load('data.mat');
4
5  % Normalizing data to remove intensity fluctuations
6  data = data - repmat(mean(data,2),1,size(data,2));
7  % Evaluating the principal components
8  cMat = cov(double(data));
9  [eVec eVal] = eigs(cMat);
10
11 % pc_i = eVec(:,i);
```

Bibliography

[1] JH Hubbell. Review of photon interaction cross section data in the medical and biological context. *Physics in Medicine & Biology*, 44(1):R1, 1999.

[2] Karel Svoboda and Steven M Block. Biological applications of optical forces. *Annual review of biophysics and biomolecular structure*, 23(1):247–285, 1994.

[3] P. Parham. *The Immune System*. Garland Science, 4th edition, 2015.

[4] Taro Ichimura, Liang-da Chiu, Katsumasa Fujita, Hiroaki Machiyama, Tomoyuki Yamaguchi, Tomonobu M. Watanabe, and Hideaki Fujita. Non-label immune cell state prediction using raman spectroscopy. *Scientific Reports*, 6(1):37562, Nov 2016.

[5] Tom C Bakker Schut, Peter J Caspers, Gerwin J Puppels, Annieke Nijssen, Freerk Heule, Martino HA Neumann, and Donal P Hayes. Discriminating basal cell carcinoma from its surrounding tissue by raman spectroscopy. *Journal of investigative dermatology*, 119(1):64–69, 2002.

[6] James W Chan, Deborah K Lieu, Thomas Huser, and Ronald A Li. Label-free separation of human embryonic stem cells and their cardiac derivatives using raman spectroscopy. *Analytical chemistry*, 81(4):1324–1331, 2009.

[7] H Georg Schulze, Stanislav O Konorov, Nicolas J Caron, James M Piret, Michael W Blades, and Robin FB Turner. Assessing differentiation status of human embryonic stem cells noninvasively using raman microspectroscopy. *Analytical chemistry*, 82(12):5020–5027, 2010.

[8] Faliu Yi, Inkyu Moon, Bahram Javidi, Daniel Boss, and Pierre P Marquet. Automated segmentation of multiple red blood cells with digital holographic microscopy. *Journal of Biomedical Optics*, 18(2):026006, 2013.

[9] Elliott Daniel SoRelle, Derek William Yecies, Orly Liba, Frederick Christian Bennett, Claus Moritz Graef, Rebecca Dutta, Siddhartha Mitra, Lydia-Marie Joubert, Samuel Cheshier, Gerald A Grant, et al. Spatiotemporal tracking of brain-tumor-associated myeloid cells in vivo through optical coherence tomography with plasmonic labeling and speckle modulation. *ACS nano*, 13(7):7985–7995, 2019.

[10] Samira M Azarin, Ji Yi, Robert M Gower, Brian A Aguado, Megan E Sullivan, Ashley G Goodman, Eric J Jiang, Shreyas S Rao, Yinying Ren, Susan L Tucker, et al. In vivo capture and label-free detection of early metastatic cells. *Nature communications*, 6(1):1–9, 2015.

[11] Jonghee Yoon, Kyoohyun Kim, HyunJoo Park, Chulhee Choi, Seongsoo Jang, and YongKeun Park. Label-free characterization of white blood cells by measuring 3d refractive index maps. *Biomedical optics express*, 6(10):3865–3875, 2015.

[12] Arthur L. Samuel. Some studies in machine learning using the game of checkers. *IBM journal of research and development*, pages 71–105, 1959.

[13] Cheolsoo Park, Clive Cheong Took, and Joon-Kyung Seong. Machine learning in biomedical engineering. *Biomedical Engineering Letters*, 8(1):1–3, Feb 2018.

[14] Christos Davatzikos. Machine learning in neuroimaging: Progress and challenges. *NeuroImage*, 197:652–656, Aug 2019. 30296563[pmid].

[15] Lenka Zdeborová. New tool in the box. *Nature Physics*, 13(5):420–421, May 2017.

[16] Haibin Chang and Dongxiao Zhang. Machine learning subsurface flow equations from data. *Computational Geosciences*, 23(5):895–910, Oct 2019.

[17] Pavlo O. Dral, Mario Barbatti, and Walter Thiel. Nonadiabatic excited-state dynamics with machine learning. *The Journal of Physical Chemistry Letters*, 9(19):5660–5663, Oct 2018.

[18] John R. Kitchin. Machine learning in catalysis. *Nature Catalysis*, 1(4):230–232, Apr 2018.

[19] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[20] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[21] Yu Zhu and Peng Zeng. Fourier methods for estimating the central subspace and the central mean subspace in regression. *Journal of the American Statistical Association*, 101(476):1638–1651, 2006.

[22] M F F Mardianto, E Tjahjono, and M Rifada. Semiparametric regression based on three forms of trigonometric function in fourier series estimator. *Journal of Physics: Conference Series*, 1277:012052, jul 2019.

[23] Sarah J Ratcliffe, Gillian Z Heller, and Leo R Leader. Functional data analysis with application to periodically stimulated foetal heart rate data. ii: Functional logistic regression. *Statistics in medicine*, 21(8):1115–1127, 2002.

[24] Todd Ogden. *Essential Wavelets for Statistical Applications and Data Analysis*. Springer, 1997.

[25] Stéphane Mallat. *A Wavelet Tour of Signal Processing (Third Ed.)*. Academic Press, 2009.

[26] Brani Vidakovic. *Statistical Modeling by Wavelets*. Wiley, 1999.

[27] Peter McCullagh. *Generalized linear models*. Routledge, 2018.

[28] T Toss. Bayes' theorem. *nature methods*, 12(4):277, 2015.

[29] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[30] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[31] Xin Jin and Jiawei Han. *K-Means Clustering*, pages 695–697. Springer US, Boston, MA, 2017.

[32] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[33] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[34] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[35] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[36] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *J. Mach. Learn. Res.*, 9:2579–2605, 2008.

[37] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*, volume 1. MIT Press, 1998.

[38] Laia Domingo Colomer, Michalis Skotiniotis, and Ramon Muñoz-Tapia. Reinforcement learning for optimal error correction of toric codes. *Physics Letters A*, 384(17):126353, 2020.

[39] Han Xu, Junning Li, Liqiang Liu, Yu Wang, Haidong Yuan, and Xin Wang. Generalizable control for quantum parameter estimation through reinforcement learning. *npj Quantum Information*, 5(1):82, Oct 2019.

[40] Gongning Luo, Suyu Dong, Kuanquan Wang, Dong Zhang, Yue Gao, Xin Chen, Henggui Zhang, and Shuo Li. A deep reinforcement learning framework for frame-by-frame plaque tracking on intravascular optical coherence tomography image. In Dinggang Shen, Tianming Liu, Terry M. Peters, Lawrence H. Staib, Caroline Essert, Sean Zhou, Pew-Thian Yap, and Ali Khan, editors,

Medical Image Computing and Computer Assisted Intervention – MICCAI 2019, pages 12–20, Cham, 2019. Springer International Publishing.

[41] Dailos Guerra-Ramos, Juan Trujillo-Sevilla, and Jose Manuel Rodríguez-Ramos. Towards piston fine tuning of segmented mirrors through reinforcement learning. *Applied Sciences*, 10(9):3207, May 2020.

[42] F Albarrán-Arriagada, JC Retamal, E Solano, and L Lamata. Reinforcement learning for semi-autonomous approximate quantum eigensolver. *Machine Learning: Science and Technology*, 1(1):015002, 2020.

[43] Seyed Sajad Mousavi, Michael Schukat, and Enda Howley. Deep reinforcement learning: an overview. In *Proceedings of SAI Intelligent Systems Conference*, pages 426–440. Springer, 2016.

[44] Rui Song and Xiyuan Chen. Analysis of fiber optic gyroscope vibration error based on improved local mean decomposition and kernel principal component analysis. *Applied Optics*, 56(8):2265–2272, 2017.

[45] Liangyun Liu, Bowen Song, Su Zhang, and Xinjie Liu. A novel principal component analysis method for the reconstruction of leaf reflectance spectra and retrieval of leaf biochemical contents. *Remote Sensing*, 9(11):1113, 2017.

[46] Ningliang Liu, Yaxiong Guo, Houmin Jiang, and Weisong Yi. Gastric cancer diagnosis using hyperspectral imaging with principal component analysis and spectral angle mapper. *Journal of Biomedical Optics*, 25(6):066005, 2020.

[47] Hengying Xu, Lishan Yang, Xinkuo Yu, Zibo Zheng, Chenglin Bai, Weibin Sun, and Xiaoguang Zhang. Blind and low-complexity modulation format identification scheme using principal component analysis of stokes parameters for elastic optical networks. *Optics Express*, 28(14):20249–20263, 2020.

[48] B Williams, YF Zhu, and Z He. Applications of principal component analysis for energy reconstruction in position-sensitive semiconductor detectors. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 954:162047, 2020.

[49] Juan D Osorio, Sergio Vilches, and Hans Zappe. Diffuse reflectance spectroscopy as a monitoring tool for gastric mucosal devitalization treatments with argon plasma coagulation. *Journal of Biophotonics*, 13(3):e201960125, 2020.

[50] Camilo Morais, Panagiotis Giamougiannis, Rita Oliwia Grabowska, Nicholas J Wood, Pierre L Martin-Hirsch, and Francis L Martin. A three-dimensional discriminant analysis approach for hyperspectral images. *Analyst*, 2020.

[51] Xiaoyu Cui, Tao Liu, Xiaosong Xu, Zeyin Zhao, Ye Tian, Yue Zhao, Shuo Chen, Zhe Wang, Yiding Wang, Dayu Hu, et al. Label-free detection of multiple genitourinary cancers from urine by surface-enhanced raman spectroscopy. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, page 118543, 2020.

[52] Xishuo Wang, Qi Zhang, Xiangjun Xin, Ran Gao, Qinghua Tian, Feng Tian, Chuxuan Wang, Xiaolong Pan, Yongjun Wang, and Leijing Yang. Robust weighted k-means clustering algorithm for a probabilistic-shaped 64qam coherent optical communication system. *Optics Express*, 27(26):37601–37613, 2019.

[53] G Hari Hara Sudhan, R Ganesh Aravind, K Gowri, and V Rajinikanth. Optic disc segmentation based on otsu's thresholding and level set. In *2017 International Conference on Computer Communication and Informatics (ICCCI)*, pages 1–5. IEEE, 2017.

[54] Jean-Martial Mari, Tin Aung, Ching-Yu Cheng, Nicholas G Strouthidis, and Michael JA Girard. A digital staining algorithm for optical coherence tomography images of the optic nerve head. *Translational vision science & technology*, 6(1):8–8, 2017.

[55] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

[56] Steve R Gunn et al. Support vector machines for classification and regression. *ISIS technical report*, 14(1):5–16, 1998.

[57] Haiqin Yang, Laiwan Chan, and Irwin King. Support vector machine regression for volatile stock market prediction. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 391–396. Springer, 2002.

[58] Mahesh Pal and PM Mather. Support vector machines for classification in remote sensing. *International journal of remote sensing*, 26(5):1007–1011, 2005.

[59] Sebastian Van der Linden, Andreas Janz, Björn Waske, Michael Eiden, and Patrick Hostert. Classifying segmented hyperspectral data from a heterogeneous urban environment using support vector machines. *Journal of Applied Remote Sensing*, 1(1):013543, 2007.

[60] J Anthony Gualtieri and Robert F Cromp. Support vector machines for hyperspectral remote sensing classification. In *27th AIPR Workshop: Advances in Computer-Assisted Recognition*, volume 3584, pages 221–232. International Society for Optics and Photonics, 1999.

[61] Zhicheng Wang, Jinwen Tian, Jian Liu, and Sheng Zheng. Small infrared target fusion detection based on support vector machines in the wavelet domain. *Optical Engineering*, 45(7):076401, 2006.

[62] Fabio Roli and Giorgio Fumera. Support vector machines for remote sensing image classification. In *Image and Signal Processing for Remote Sensing VI*, volume 4170, pages 160–166. International Society for Optics and Photonics, 2001.

[63] Ilir Gashi, Vladimir Stankovic, Corrado Leita, and Olivier Thonnard. An experimental study of diversity with off-the-shelf antivirus engines. In *2009 Eighth IEEE International Symposium on Network Computing and Applications*, pages 4–11. IEEE, 2009.

[64] Philippe Hamel and Douglas Eck. Learning features from music audio with deep belief networks. In *ISMIR*, volume 10, pages 339–344. Utrecht, The Netherlands, 2010.

[65] Andrew R Jamieson, Maryellen L Giger, Karen Drukker, Hui Li, Yading Yuan, and Neha Bhooshan. Exploring nonlinear feature space dimension reduction and data representation in breast cadx with laplacian eigenmaps and-sne. *Medical physics*, 37(1):339–351, 2010.

[66] Izhar Wallach and Ryan Lilien. The protein–small-molecule database, a non-redundant structural resource for the analysis of protein-ligand binding. *Bioinformatics*, 25(5):615–620, 2009.

[67] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.

[68] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[69] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[70] JC Hoskins and DM Himmelblau. Process control via artificial neural networks and reinforcement learning. *Computers & chemical engineering*, 16(4):241–251, 1992.

[71] Stephen Dominic, R Das, D Whitley, and C Anderson. Genetic reinforcement learning for neural networks. In *IJCNN-91-Seattle International Joint Conference on Neural Networks*, volume 2, pages 71–76. IEEE, 1991.

[72] Geng Deng and Michael C. Ferris. *Neuro-dynamic programming for fractionated radiotherapy planning*, pages 47–70. Springer New York, New York, NY, 2008.

[73] Model estimation of arma using genetic algorithms: A case study of forecasting natural gas consumption. *Procedia - Social and Behavioral Sciences*, 235:537 – 545, 2016. 12th International Strategic Management Conference, ISMC 2016, 28-30 October 2016, Antalya, Turkey.

[74] Roman M. Balabin and Ekaterina I. Lomakina. Neural network approach to quantum-chemistry data: Accurate prediction of density functional theory energies. *The Journal of Chemical Physics*, 131(7):074104, 2009.

[75] Lung sound classification using cepstral-based statistical features. *Computers in Biology and Medicine*, 75:118 – 129, 2016.

[76] N Ganesan, K Venkatesh, MA Rama, and A Malathi Palani. Application of neural networks in diagnosing cancer disease using demographic data. *International Journal of Computer Applications*, 1:76–85, 2010.

[77] Marc Buyse and Pascal Piedbois. Artificial neural networks. *The Lancet*, 350(9085):1175, 1997.

[78] Elaheh Alizadeh, Samanthe Merrick Lyons, Jordan Marie Castle, and Ashok Prasad. Measuring systematic changes in invasive cancer cell shape using zernike moments. *Integr. Biol.*, 8:1183–1193, 2016.

[79] Samanthe M. Lyons, Elaheh Alizadeh, Joshua Mannheimer, Katherine Schuamberg, Jordan Castle, Bryce Schroder, Philip Turk, Douglas Thamm, and Ashok Prasad. Changes in cell shape are correlated with metastatic potential in murine and human osteosarcomas. *Biology Open*, 5(3):289–299, 2016.

[80] Significant wave height record extension by neural networks and reanalysis wind data. *Ocean Modelling*, 94:128 – 140, 2015.

[81] Bilwaj Gaonkar, David Hovda, Neil Martin, and Luke Macyszyn. Deep learning in the small sample size setting: cascaded feed forward neural networks for medical image segmentation. In *Medical Imaging 2016: Computer-Aided Diagnosis*, volume 9785, page 97852I. International Society for Optics and Photonics, 2016.

[82] Yuan Jing, Bahar Youssefi, Mitra Mirhassani, and Roberto Muscedere. An efficient fpga implementation of optical character recognition for license plate recognition. In *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–4. IEEE, 2017.

[83] Yousef Fazea, Mohd Samsu Sajat, Amran Ahmad, and Mustafa Muwafak Alobaedy. Channel optimization in mode division multiplexing using neural networks. In *2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA)*, pages 173–175. IEEE, 2018.

[84] Hasan Sildir, Erdal Aydin, and Taskin Kavzoglu. Design of feedforward neural networks in the classification of hyperspectral imagery using superstructural optimization. *Remote Sensing*, 12(6):956, 2020.

[85] Shotaro Owaki and Moriya Nakamura. Equalization of optical nonlinear waveform distortion using neural-network based digital signal processing. In *2016 21st OptoElectronics and Communications Conference (OECC) held jointly with 2016 International Conference on Photonics in Switching (PS)*, pages 1–3. IEEE, 2016.

[86] Sunny Chugh, Aamir Gulistan, Souvik Ghosh, and BMA Rahman. Machine learning approach for computing optical properties of a photonic crystal fiber. *Optics Express*, 27(25):36414–36425, 2019.

[87] Petya Em Pavlova, Nikola G Shakev, and Ekaterina G Borisova. Comparative analysis of methods for ascertainment the similarity between reflected spectra obtained from skin lesions. *IFAC-PapersOnLine*, 52(25):365–369, 2019.

[88] Gang Wu, Thomas Nowotny, Yongliang Zhang, Hong-Qi Yu, and David Day-Uei Li. Artificial neural network approaches for fluorescence lifetime imaging techniques. *Optics letters*, 41(11):2561–2564, 2016.

[89] Bendix Schneider, Joni Dambre, and Peter Bienstman. Fast particle characterization using digital holography and neural networks. *Applied optics*, 55(1):133–139, 2016.

[90] S Pilati and P Pieri. Supervised machine learning of ultracold atoms with speckle disorder. *Scientific reports*, 9(1):1–12, 2019.

[91] Henry Pinkard, Zachary Phillips, Arman Babakhani, Daniel A Fletcher, and Laura Waller. Deep learning for single-shot autofocus microscopy. *Optica*, 6(6):794–797, 2019.

[92] Yann LeCun et al. Generalization and network design strategies. *Connectionism in perspective*, 19:143–155, 1989.

[93] Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 111–118, 2010.

[94] Sascha D Krauß, Raphael Roy, Hesham K Yosef, Tatjana Lechtonen, Samir F El-Mashtoly, Klaus Gerwert, and Axel Mosig. Hierarchical deep convolutional neural networks combine spectral and spatial information for highly accurate raman-microscopy-based cytopathology. *Journal of biophotonics*, 11(10):e201800022, 2018.

[95] Sayed Mohammad Ebrahim Sahraeian, Ruolin Liu, Bayo Lau, Karl Podesta, Marghoob Mohiyuddin, and Hugo YK Lam. Deep convolutional neural networks for accurate somatic mutation detection. *Nature communications*, 10(1):1–10, 2019.

[96] Sebastian Berisha, Mahsa Lotfollahi, Jahandar Jahanipour, Ilker Gurcan, Michael Walsh, Rohit Bhargava, Hien Van Nguyen, and David Mayerich. Deep learning for ftir histology: leveraging spatial and spectral features with convolutional neural networks. *Analyst*, 144(5):1642–1653, 2019.

[97] Zijie J Wang, Alex J Walsh, Melissa C Skala, and Anthony Gitter. Classifying t cell activity in autofluorescence intensity images with convolutional neural networks. *Journal of Biophotonics*, 13(3):e201960050, 2020.

[98] Talmo D. Pereira, Diego E. Aldarondo, Lindsay Willmore, Mikhail Kislin, Samuel S.-H. Wang, Mala Murthy, and Joshua W. Shaevitz. Fast animal pose estimation using deep neural networks. *Nature Methods*, 16(1):117–125, Jan 2019.

[99] Tristan Bepler, Andrew Morin, Micah Rapp, Julia Brasch, Lawrence Shapiro, Alex J. Noble, and Bonnie Berger. Positive-unlabeled convolutional neural networks for particle picking in cryo-electron micrographs. *Nature Methods*, 16(11):1153–1160, Nov 2019.

[100] Wonseok Jeon, Wooyoung Jeong, Kyungchan Son, and Hyunseok Yang. Speckle noise reduction for digital holographic images using multi-scale convolutional neural networks. *Optics letters*, 43(17):4240–4243, 2018.

[101] Min Xu, Chen Tang, Fugui Hao, Mingming Chen, and Zhenkun Lei. Texture preservation and speckle reduction in poor optical coherence tomography using the convolutional neural network. *Medical Image Analysis*, page 101727, 2020.

[102] Yair Rivenson, Zoltán Göröcs, Harun Günaydin, Yibo Zhang, Hongda Wang, and Aydogan Ozcan. Deep learning microscopy. *Optica*, 4(11):1437–1443, 2017.

[103] Yichen Wu, Yair Rivenson, Hongda Wang, Yilin Luo, Eyal Ben-David, Laurent A Bentolila, Christian Pritz, and Aydogan Ozcan. Three-dimensional virtual refocusing of fluorescence microscopy images using deep learning. *Nature methods*, 16(12):1323–1331, 2019.

[104] Saga Helgadottir, Aykut Argun, and Giovanni Volpe. Digital video microscopy enhanced by deep learning. *Optica*, 6(4):506–513, Apr 2019.

[105] Ayan Sinha, Justin Lee, Shuai Li, and George Barbastathis. Lensless computational imaging through deep learning. *Optica*, 4(9):1117–1125, 2017.

[106] Yunzhe Li, Yujia Xue, and Lei Tian. Deep speckle correlation: a deep learning approach toward scalable imaging through scattering media. *Optica*, 5(10):1181–1190, 2018.

[107] Yohei Nishizaki, Matias Valdivia, Ryoichi Horisaki, Katsuhisa Kitaguchi, Mamoru Saito, Jun Tanida, and Esteban Vera. Deep learning wavefront sensing. *Optics express*, 27(1):240–251, 2019.

[108] Gong Zhang, Tian Guan, Zhiyuan Shen, Xiangnan Wang, Tao Hu, Delai Wang, Yonghong He, and Ni Xie. Fast phase retrieval in off-axis digital holographic microscopy through deep learning. *Optics express*, 26(15):19388–19405, 2018.

[109] Ronald J Williams and Jing Peng. An efficient gradient-based algorithm for on-line training of recurrent network trajectories. *Neural computation*, 2(4):490–501, 1990.

[110] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[111] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2016.

[112] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[113] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[114] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[115] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1):1–38, 2019.

[116] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pages 7354–7363, 2019.

[117] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

[118] Roman Spilger, Andrea Imle, Ji-Young Lee, Barbara Müller, Oliver T Fackler, Ralf Bartenschlager, and Karl Rohr. A recurrent neural network for particle tracking in microscopy images using future information, track hypotheses, and multiple detections. *IEEE Transactions on Image Processing*, 29:3681–3694, 2020.

[119] Assaf Arbelle and Tammy Riklin Raviv. Microscopy cell segmentation via convolutional lstm networks. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 1008–1012. IEEE, 2019.

[120] Alican Bozkurt, Trevor Gale, Kivanc Kose, Christi Alessi-Fox, Dana H Brooks, Milind Rajadhyaksha, and Jennifer Dy. Delineation of skin strata in reflectance confocal microscopy images with recurrent convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 25–33, 2017.

[121] Jianxu Chen, Lin Yang, Yizhe Zhang, Mark Alber, and Danny Z Chen. Combining fully convolutional and recurrent neural networks for 3d biomedical image segmentation. In *Advances in neural information processing systems*, pages 3036–3044, 2016.

[122] Stefano Bo, Falko Schmidt, Ralf Eichhorn, and Giovanni Volpe. Measurement of anomalous diffusion using recurrent neural networks. *Physical Review E*, 100(1):010102, 2019.

[123] Adrian MeidellFiorito, Andreas Østvik, Erik Smistad, Sarah Leclerc, Olivier Bernard, and Lasse Lovstakken. Detection of cardiac events in echocardiography using 3d convolutional recurrent neural networks. In *2018 IEEE International Ultrasonics Symposium (IUS)*, pages 1–4. IEEE, 2018.

[124] Timothy O'Connor, Arun Anand, Biree Andemariam, and Bahram Javidi. Deep learning-based cell identification and disease diagnosis using spatio-temporal cellular dynamics in compact digital holographic microscopy. *Biomedical Optics Express*, 11(8):4491–4508, 2020.

[125] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative ad-

versarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[126] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[127] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[128] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.

[129] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915, 2017.

[130] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[131] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*, 2017.

[132] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[133] Kevin de Haan, Zachary S Ballard, Yair Rivenson, Yichen Wu, and Aydogan Ozcan. Resolution enhancement in scanning electron microscopy using deep learning. *Scientific Reports*, 9(1):1–7, 2019.

[134] Hongda Wang, Yair Rivenson, Yiyin Jin, Zhensong Wei, Ronald Gao, Harun Günaydın, Laurent A Bentolila, Comert Kural, and Aydogan Ozcan. Deep

learning enables cross-modality super-resolution in fluorescence microscopy. *Nat. Methods*, 16:103–110, 2019.

[135] Wei Ouyang, Andrey Aristov, Mickaël Lelek, Xian Hao, and Christophe Zimmer. Deep learning massively accelerates super-resolution localization microscopy. *Nature biotechnology*, 36(5):460–468, 2018.

[136] Stephan J Ihle, Andreas M Reichmuth, Sophie Girardin, Hana Han, Flurin Stauffer, Anne Bonnin, Marco Stampanoni, Karthik Pattisapu, János Vörös, and Csaba Forró. Unsupervised data to content transformation with histogram-matching cycle-consistent generative adversarial networks. *Nature Machine Intelligence*, 1(10):461–470, 2019.

[137] Xiu Li, Jiuyang Dong, Bowen Li, Yi Zhang, Yongbing Zhang, Ashok Veeraraghavan, and Xiangyang Ji. Fast confocal microscopy imaging based on deep learning. In *2020 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12. IEEE, 2020.

[138] Soonam Lee, Shuo Han, Paul Salama, Kenneth W Dunn, and Edward J Delp. Three dimensional blind image deconvolution for fluorescence microscopy using generative adversarial networks. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 538–542. IEEE, 2019.

[139] Junxi Feng, Xiaohai He, Qizhi Teng, Chao Ren, Honggang Chen, and Yang Li. Reconstruction of porous media from extremely limited information using conditional generative adversarial networks. *Physical Review E*, 100(3):033308, 2019.

[140] Yichen Wu, Yilin Luo, Gunvant Chaudhari, Yair Rivenson, Ayfer Calis, Kevin De Haan, and Aydogan Ozcan. Bright-field holography: cross-modality deep learning enables snapshot 3d imaging with bright-field contrast using a single hologram. *Light: Science & Applications*, 8(1):1–7, 2019.

[141] Gunho Choi, DongHun Ryu, YoungJu Jo, Young Seo Kim, Weisun Park, Hyun-seok Min, and YongKeun Park. Cycle-consistent deep learning approach to coherent noise reduction in optical diffraction tomography. *Optics express*, 27(4):4927–4943, 2019.

[142] Lennart Bargsten and Alexander Schlaefer. Specklegan: a generative adversarial network with an adaptive speckle layer to augment limited training data for ultrasound image processing. *International Journal of Computer Assisted Radiology and Surgery*, pages 1–10, 2020.

[143] Zhao Dong, Guoyan Liu, Guangming Ni, Jason Jerwick, Lian Duan, and Chao Zhou. Optical coherence tomography image denoising using a generative adversarial network with speckle modulation. *Journal of Biophotonics*, 13(4):e201960135, 2020.

[144] Jiaqi Jiang, David Sell, Stephan Hoyer, Jason Hickey, Jianji Yang, and Jonathan A Fan. Free-form diffractive metagrating design based on generative adversarial networks. *ACS nano*, 13(8):8872–8878, 2019.

[145] Roopam K. Gupta, Mingzhou Chen, Graeme P. A. Malcolm, Nils Hempler, Kishan Dholakia, and Simon J. Powis. Label-free optical hemogram of granulocytes enhanced by artificial neural networks. *Opt. Express*, 27(10):13706–13720, May 2019.

[146] Abul K Abbas, Andrew H Lichtman, and Shiv Pillai. *Basic Immunology E-Book: Functions and Disorders of the Immune System*. Elsevier Health Sciences, 2019.

[147] Anuradha Ramoji, Ute Neugebauer, Thomas Bocklitz, Martin Foerster, Michael Kiehntopf, Michael Bauer, and Jürgen Popp. Toward a spectroscopic hemogram: Raman spectroscopic differentiation of the two most abundant leukocytes from peripheral blood. *Analytical Chemistry*, 84(12):5335–5342, 2012. PMID: 22721427.

[148] M. H. Julius L. A. Herzenberg and T. Masada. Demonstration that antigen-binding cells are precursors of antibody-producing cells after purification with fluorescence activated cell sorter. *PNAS*, 69(7):1934–1938, 1972.

[149] Immunolgy/immunofluorescence protocol.

[150] J. et al. Sulé-Suso. Vibrational spectroscopy in stem cell characterisation: is there a niche? *Trends in Biotechnology*, 32:254–262.

[151] R.J. Swain and M.M. Stevens. Raman microspectroscopy for non-invasive biochemical analysis of single cells. *Biochemical Society Transactions*, 35(3):544–549, 2007.

[152] Alina Bogumila Zoladek, Ramneek Kaur Johal, Samuel Garcia-Nieto, Flavius Pascut, Kevin M. Shakesheff, Amir M. Ghaemmaghami, and Ioan Notingher. Label-free molecular imaging of immunological synapses between dendritic and t cells by raman micro-spectroscopy. *Analyst*, 135:3205–3212, 2010.

[153] Alison J. Hobro, Yutaro Kumagai, Shizuo Akira, and Nicholas I. Smith. Raman spectroscopy as a tool for label-free lymphocyte cell line discrimination. *Analyst*, 141:3756–3764, 2016.

[154] Mengqiu Li, Jian Xu, Maria Romero-Gonzalez, Steve A Banwart, and Wei E Huang. Single cell raman spectroscopy for cell sorting and imaging. *Current Opinion in Biotechnology*, 23(1):56 – 63, 2012. Analytical biotechnology.

[155] Derek J. Gardiner and Pierre R. Graves, editors. *Practical Raman Spectroscopy*. Springer-Verlag Berlin Heidelberg.

[156] Chandrasekhara Venkata Raman and Kariamanikkam Srinivasa Krishnan. A new type of secondary radiation. *Nature*, 121(3048):501–502, 1928.

[157] Debabrata Mandal, Misao Mizuno, and Tahei Tahara. Temporal fluorescence rejection in raman spectroscopy using femtosecond up-conversion with single- and multi-channel detection. *Journal of Molecular Structure*, 735:189 – 195, 2005. Raman, Infrared and Theoretical Investigations of Molecular Structures.A collection of invited papers in honor of the 70th Birthday of Professor Hiroaki Takahashi.

[158] Takeshi Hasegawa, Jujiro Nishijo, and Junzo Umemura. Separation of raman spectra from fluorescence emission background by principal component analysis. *Chemical Physics Letters*, 317(6):642 – 646, 2000.

[159] E. Campani, G. Gorini, and G. J. Masetti. *J. Appl. Phys*, 14:2189–2197, 1981.

[160] S. M. Angel, M. K. DeArmond, K. W. Hanck, and D. W. Wertz. Computer-controlled instrument for the recovery of a resonance raman spectrum in the

presence of strong luminescence. *Analytical Chemistry*, 56(14):3000–3001, 1984.

[161] I. G. Cormack, M. Mazilu, K. Dholakia, and C. S. Herrington. *Appl. Phys. Lett*, 91, 2007.

[162] G. Rusciano, A. C. De Luca, A. Sasso, and G. Pesce. Enhancing raman tweezers by phase-sensitive detection. *Analytical Chemistry*, 79(10):3708–3715, 2007. PMID: 17444615.

[163] Anna Chiara De Luca, Michael Mazilu, Andrew Riches, C. Simon Herrington, and Kishan Dholakia. Online fluorescence suppression in modulated raman spectroscopy. *Analytical Chemistry*, 82(2):738–745, 2010. PMID: 20017474.

[164] Mingzhou Chen, Naomi McReynolds, Elaine C. Campbell, Michael Mazilu, João Barbosa, Kishan Dholakia, and Simon J. Powis. The use of wavelength modulated raman spectroscopy in label-free identification of t lymphocyte subsets, natural killer cells and dendritic cells. *PLOS ONE*, 10(5):1–14, 05 2015.

[165] Derek Craig, Michael Mazilu, and Kishan Dholakia. Quantitative detection of pharmaceuticals using a combination of paper microfluidics and wavelength modulated raman spectroscopy. *PLOS ONE*, 10(5):1–10, 05 2015.

[166] Bavishna B. Praveen, Michael Mazilu, Robert F. Marchington, C. Simon Herrington, Andrew Riches, and Kishan Dholakia. Optimisation of wavelength modulated raman spectroscopy: Towards high throughput cell screening. *PLOS ONE*, 8(6):1–5, 06 2013.

[167] Etienne Cuche, Frédéric Bevilacqua, and Christian Depeursinge. Digital holography for quantitative phase-contrast imaging. *Opt. Lett.*, 24(5):291–293, Mar 1999.

[168] Zahra Monemhaghdoust, Frédéric Montfort, Yves Emery, Christian Depeursinge, and Christophe Moser. Off-axis digital holographic camera for quantitative phase microscopy. *Biomed. Opt. Express*, 5(6):1721–1730, Jun 2014.

[169] Shigeru Murata and Norifumi Yasuda. Potential of digital holography in particle measurement. *Optics and Laser Technology*, 32(7):567 – 574, 2000. Optical methods in heat and fluid flow.

[170] Chetan A. Patil, Nienke Bosschaart, Matthew D. Keller, Ton G. van Leeuwen, and Anita Mahadevan-Jansen. Combined raman spectroscopy and optical coherence tomography device for tissue characterization. *Opt. Lett.*, 33(10):1135–1137, May 2008.

[171] Naomi McReynolds, Fiona G. M. Cooke, Mingzhou Chen, Simon J. Powis, and Kishan Dholakia. Multimodal discrimination of immune cells using a combination of raman spectroscopy and digital holographic microscopy. *Scientific Reports*, 7(43631), 2017.

[172] Kazuyoshi Itoh. Analysis of the phase unwrapping algorithm. *Appl. Opt.*, 21(14):2470–2470, Jul 1982.

[173] Miguel Arevallilo Herráez, David R. Burton, Michael J. Lalor, and Munther A. Gdeisat. Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a noncontinuous path. *Appl. Opt.*, 41(35):7437–7444, Dec 2002.

[174] Vincent O Baron, Mingzhou Chen, Simon O Clark, Ann Williams, Robert J. H. Hammond, Kishan Dholakia, and Stephen H. Gillespie. Label-free optical vibrational spectroscopy to detect the metabolic state of M. tuberculosis cells at the site of disease. *Sci. Rep.*, 7(1):9844, dec 2017.

[175] Vincent O. Baron, Mingzhou Chen, Simon O. Clark, Ann Williams, Kishan Dholakia, and Stephen H. Gillespie. *Detecting Phenotypically Resistant Mycobacterium tuberculosis Using Wavelength Modulated Raman Spectroscopy*, pages 41–50. Springer New York, New York, NY, 2018.

[176] Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.

[177] R. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160, 1981.

[178] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proceedings of the 32nd International Conference on Machine Learning, Lille, France*, 37:448–456, 2015.

[179] Gavin C. Cawley and Nicola L.C. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *J. Mach. Learn. Res.*, 11:2079–2107, August 2010.

[180] Hongmei Yan, Yingtao Jiang, Jun Zheng, Chenglin Peng, and Qinghui Li. A multilayer perceptron-based medical decision support system for heart disease diagnosis. *Expert Systems with Applications*, 30(2):272 – 281, 2006.

[181] Umut Orhan, Mahmut Hekim, and Mahmut Ozer. Eeg signals classification using the k-means clustering and a multilayer perceptron neural network model. *Expert Systems with Applications*, 38(10):13475 – 13481, 2011.

[182] Le Zhang and P.N. Suganthan. A survey of randomized algorithms for training neural networks. *Information Sciences*, 364-365:146 – 155, 2016.

[183] Tao Peng. Detect circles with various radii in grayscale image via hough transform, 2020.

[184] W. Yang, X. Zhang, Y. Tian, W. Wang, J. Xue, and Q. Liao. Deep learning for single image super-resolution: A brief review. *IEEE Transactions on Multimedia*, 21(12):3106–3121, 2019.

[185] Larissa Heinrich, John A Bogovic, and Stephan Saalfeld. Deep learning for isotropic super-resolution from non-isotropic 3d electron microscopy. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 135–143. Springer, 2017.

[186] Elias Nehme, Lucien E Weiss, Tomer Michaeli, and Yoav Shechtman. Deep-storm: super-resolution single-molecule microscopy by deep learning. *Optica*, 5(4):458–464, 2018.

[187] Ruud JG van Sloun, Oren Solomon, Matthew Bruce, Zin Z Khaing, Hessel Wijkstra, Yonina C Eldar, and Massimo Mischi. Super-resolution ultrasound localization microscopy through deep learning. *arXiv preprint arXiv:1804.07661*, 2018.

[188] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics.

[189] Roopam K. Gupta, Graham D. Bruce, Simon J. Powis, and Kishan Dholakia. Deep learning enabled laser speckle wavemeter with a high dynamic range. *Laser & Photonics Reviews*, 14(9):2000120, 2020.

[190] P Hariharan. Speckle patterns: a historical retrospect. *Optica Acta: International Journal of Optics*, 19(9):791–793, 1972.

[191] Robert V Langmuir. Scattering of laser light. *Applied Physics Letters*, 2(2):29–30, 1963.

[192] Joseph W Goodman. Statistical properties of laser sparkle patterns. Technical report, STANFORD UNIV CA STANFORD ELECTRONICS LABS, 1963.

[193] Zong-Guo Xia and Yongwei Sheng. Radar speckle: noise or information? In *IGARSS'96. 1996 International Geoscience and Remote Sensing Symposium*, volume 1, pages 48–50. IEEE, 1996.

[194] Brandon Redding and Hui Cao. Using a multimode fiber as a high-resolution, low-loss spectrometer. *Opt. Lett.*, 37:3384, 2012.

[195] Brandon Redding, Sebastien M. Popoff, and Hui Cao. All-fiber spectrometer based on speckle pattern reconstruction. *Opt. Express*, 21:6584, 2013.

[196] Brandon Redding, Mansoor Alam, Martin Seifert, and Hui Cao. High-resolution and broadband all-fiber spectrometers. *Optica*, 1:175, 2014.

[197] Noel H. Wan, Fan Meng, Tim Schröder, Ren-Jye Shiue, Edward H. Chen, and Dirk Englund. High-resolution optical spectroscopy using multimode interference in a compact tapered fibre. *Nat. Commun.*, 6:7762, 2015.

[198] Graham D. Bruce, Laura O'Donnell, Mingzhou Chen, and Kishan Dholakia. Overcoming the speckle correlation limit to achieve a fiber wavemeter with attometer resolution. *Opt. Lett.*, 44(6):1367–1370, 2019.

[199] Graham D Bruce, Laura O'Donnell, Mingzhou Chen, Morgan Facchin, and Kishan Dholakia. Femtometer-resolved simultaneous measurement of multiple laser wavelengths in a speckle wavemeter. *Opt. Lett.*, 45(7):1926–1929, 2020.

[200] Maumita Chakrabarti, Michael Linde Jakobsen, and Steen G. Hanson. Speckle-based spectrometer. *Opt. Lett.*, 40:3264, 2015.

[201] Seng Fatt Liew, Brandon Redding, Michael A. Choma, Hemant D. Tagare, and Hui Cao. Broadband multimode fiber spectrometer. *Opt. Lett.*, 41:2029, 2016.

[202] Hui Cao. Perspective on speckle spectrometers. *J. Opt.*, 19:060402, 2017.

[203] S. M. Popoff, G. Lerosey, R. Carminati, M. Fink, A. C. Boccara, and S. Gigan. Measuring the transmission matrix in optics: An approach to the study and control of light propagation in disordered media. *Phys. Rev. Lett.*, 104:100601, 2010.

[204] Michael Mazilu, Tom Vettenburg, Andrea Di Falco, and Kishan Dholakia. Random super-prism wavelength meter. *Opt. Lett.*, 39:96, 2014.

[205] Nikolaus Klaus Metzger, Roman Spesyvtsev, Graham D. Bruce, Bill Miller, Gareth T. Maker, Graeme Malcolm, Michael Mazilu, and Kishan Dholakia. Harnessing speckle for a sub-femtometre resolved broadband wavemeter and laser stabilization. *Nat. Commun.*, 8:15610, 2017.

[206] Graham D Bruce, Laura O'Donnell, Mingzhou Chen, Morgan Facchin, and Kishan Dholakia. Femtometer-resolved simultaneous measurement of multiple laser wavelengths in a speckle wavemeter. *Optics Letters*, 45(7):1926–1929, 2020.

[207] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation Learning: A Review and New Perspectives. *arXiv:1206.5538*, 2012.

[208] M. Facchin, G. D. Bruce, and K. Dholakia. Speckle-based determination of the polarisation state of single and multiple laser beams. *OSA Continuum*, 3:1302, 2020.

[209] A Mourka, M Mazilu, Ewan M Wright, and K Dholakia. Modal characterization using principal component analysis: application to bessel, higher-order gaussian beams and their superposition. *Sci. Rep.*, 3:1422, 2013.

[210] Salla Gangi Reddy, Shashi Prabhakar, Ashok Kumar, J Banerji, and RP Singh. Higher order optical vortices and formation of speckles. *Opt. Lett.*, 39(15):4364–4367, 2014.

[211] Xiao-Bo Hu, Meng-Xuan Dong, Zhi-Han Zhu, Wei Gao, and Carmelo Rosales-Guzmán. Does the structure of light influence the speckle size? *Sci. Rep.*, 10:199, 2020.

[212] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In *Proc. ICML*, volume 30, page 3, 2013.

[213] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.*, 15:1929–1958, 2014.

[214] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[215] Nicholas George and Atul Jain. Space and wavelength dependence of speckle intensity. *Appl. Phys.*, 4:201–212, 1974.

[216] Martin Fodslette Møller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks*, 6(4):525–533, 1993.

[217] Einar B. Magnusson, J. P. Balthasar Mueller, Michael Juhl, Carlos Mendoza, and Kristjan Leosson. Neural Polarimeter and Wavemeter. *ACS Photonics*, 5:2682–2687, 2018.

[218] A. Pantelopoulos and N. Bourbakis. A survey on wearable biosensor systems for health monitoring. In *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4887–4890, Aug 2008.

[219] Laura B Sagle, Laura K Ruvuna, Julia A Ruemmele, and Richard P Van Duyne. Advances in localized surface plasmon resonance spectroscopy biosensing. *Nanomedicine*, 6(8):1447–1462, 2011. PMID: 22026381.

[220] Adam K. Wanekaya, Wilfred Chen, and Ashok Mulchandani. Recent biosensing developments in environmental security. *J. Environ. Monit.*, 10:703–712, 2008.

[221] Kirill Zinoviev, Laura G. Carrascosa, José Sánchez del Río, Borja Sepúlveda, Carlos Domínguez, and Laura M. Lechuga. Silicon photonic biosensors for lab-on-a-chip applications. *Advances in Optical Technologies*, 2008, 2008. Article ID: 383927.

[222] Diming Zhang and Qingjun Liu. Biosensors and bioelectronics on smartphone for portable biochemical detection. *Biosensors and Bioelectronics*, 75:273 – 284, 2016.

[223] G. L. Cote, R. M. Lec, and M. V. Pishko. Emerging biomedical sensing technologies and their applications. *IEEE Sensors Journal*, 3(3):251–266, June 2003.

[224] J. Dübendorfer and R. E. Kunz. Compact integrated optical immunosensor using replicated chirped grating coupler sensor chips. *Appl. Opt.*, 37(10):1890–1894, Apr 1998.

[225] Graham J. Triggs, Yue Wang, Christopher P. Reardon, Matthias Fischer, Gareth J. O. Evans, and Thomas F. Krauss. Chirped guided-mode resonance biosensor. *Optica*, 4(2):229–234, Feb 2017.

[226] Bo Liedberg, Claes Nylander, and Ingemar Lunström. Surface plasmon resonance for gas detection and biosensing. *Sensors and Actuators*, 4:299 – 304, 1983.

[227] Steve Blair and Yan Chen. Resonant-enhanced evanescent-wave fluorescence biosensing with cylindrical optical cavities. *Appl. Opt.*, 40(4):570–582, Feb 2001.

[228] Victor S.-Y. Lin, Kianoush Motesharei, Keiki-Pua S. Dancil, Michael J. Sailor, and M. Reza Ghadiri. A porous silicon-based optical interferometric biosensor. *Science*, 278(5339):840–843, 1997.

[229] Robert W. Boyd and John E. Heebner. Sensitive disk resonator photonic biosensor. *Appl. Opt.*, 40(31):5742–5747, Nov 2001.

[230] Chanda Ranjit Yonzon, Christy L. Haynes, Xiaoyu Zhang, Joseph T. Walsh, and Richard P. Van Duyne. A glucose biosensor based on surface-enhanced raman scattering: improved partition layer, temporal stability, reversibility, and resistance to serum protein interference. *Analytical Chemistry*, 76(1):78–85, 2004. PMID: 14697035.

[231] Marta Bally, Martin Halter, Janos Vörös, and H. Michelle Grandin. Optical microarray biosensing techniques. *Surface and Interface Analysis*, 38(11):1442–1458, 2006.

[232] Giampaolo Pitruzzello and Thomas F Krauss. Photonic crystal resonances for sensing and imaging. *Journal of Optics*, 20(7):073004, jun 2018.

[233] G. J. Triggs, M. Fischer, D. Stellinga, M. G. Scullion, G. J. O. Evans, and T. F. Krauss. Spatial resolution and refractive index contrast of resonant

photonic crystal surfaces for biosensing. *IEEE Photonics Journal*, 7(3):1–10, June 2015.

[234] U. Fano. Effects of configuration interaction on intensities and phase shifts. *Phys. Rev.*, 124:1866–1878, Dec 1961.

[235] Shanhui Fan and J. D. Joannopoulos. Analysis of guided resonances in photonic crystal slabs. *Phys. Rev. B*, 65:235112, Jun 2002.

[236] J. Albert, S. Theriault, F. Bilodeau, D. C. Johnson, K. O. Hill, P. Sixt, and M. J. Rooks. Minimization of phase errors in long fiber bragg grating phase masks made using electron beam lithography. *IEEE Photonics Technology Letters*, 8(10):1334–1336, Oct 1996.